H. Ishibuchi, K. Narukawa, N. Tsukamoto, and Y. Nojima, “An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization,” *European Journal of Operational Research*, vol. 188, no. 1, pp. 57-75, July 2008.

An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization

Hisao Ishibuchi*, Kaname Narukawa, Noritaka Tsukamoto, and Yusuke Nojima
Department of Computer Science and Intelligent Systems, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan

* Corresponding Author
Prof. Hisao Ishibuchi
Department of Computer Science and Intelligent Systems,
Osaka Prefecture University,
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
Fax: +81-72-254-9915. E-mail address: hisaoi@cs.osakafu-u.ac.jp

Abstract

We have already proposed a similarity-based mating scheme to recombine extreme and similar parents for evolutionary multiobjective optimization. In this paper, we examine the effect of the similarity-based mating scheme on the performance of evolutionary multiobjective optimization (EMO) algorithms. First we examine which is better between recombining similar or dissimilar parents. Next we examine the effect of biasing selection probabilities toward extreme solutions that are dissimilar from other solutions in each population. Then we examine the effect of dynamically changing the strength of this bias during the execution of EMO algorithms. Computational experiments are performed on a wide variety of test problems for multiobjective combinatorial optimization. Experimental results show that the performance of EMO algorithms can be improved by the similarity-based mating scheme for many test problems.

Keywords: Multiple objective programming, combinatorial optimization, evolutionary computation, genetic algorithms.

1. Introduction

Since Schaffer’s pioneering study (Schaffer (1985)), evolutionary algorithms have been applied to various multiobjective optimization problems for finding their Pareto-optimal solutions (e.g., see Deb (2001), Coello et al. (2002), and Coello and Lamont (2004)). Those algorithms are often referred to as evolutionary multiobjective optimization (EMO) algorithms. Recent EMO algorithms usually share some common ideas such as Pareto ranking, diversity preserving and elitism. While mating restriction has often been discussed in the literature, it has not been used in many EMO algorithms as pointed out in some reviews on EMO algorithms (e.g., see Fonseca and Fleming (1995), Zitzler and Thiele (1999), and Van Veldhuizen and Lamont (2000)).

Mating restriction was suggested by Goldberg (1989) for single-objective genetic algorithms. Hajela and Lin (1992) and Fonseca and Fleming (1993) used it in their EMO algorithms. The basic idea of mating restriction is to ban the recombination of dissimilar parents from which good offspring are not likely to be generated. In the implementation of mating restriction, a user-definable parameter σ_{mating} called the mating radius is usually used for banning the recombination of two parents whose distance is larger than σ_{mating}. The distance between two parents is measured in the decision space or the objective space. The necessity of mating restriction in EMO algorithms was also stressed by Jaszkiewicz (2002a), Watanabe et al. (2002), Kim et al. (2004), and Sato et al. (2004). On the other hand, Zitzler and Thiele (1998) reported that no improvement was achieved by mating restriction in their computational experiments. Van Veldhuizen and Lamont (2000) mentioned that the empirical evidence presented in the literature could be interpreted as an argument either for or against the use of mating restriction. Moreover, there was also an argument for the recombination of dissimilar parents. Horn et al. (1994) argued that information from very different types of tradeoffs could be combined to yield other kinds of good tradeoffs. Schaffer (1985) examined the recombination of dissimilar parents but observed no improvement.

One difficulty in the use of mating restriction is the specification of the mating radius. It is very difficult to pre-specify an appropriate value of the mating radius when we do not know the distribution of Pareto-optimal solutions. As a simple mating restriction method with no mating radius, we proposed a similarity-based mating scheme to recombine similar parents (Ishibuchi and Shibata (2003a)). This mating scheme was extended to recombine extreme and similar parents in Ishibuchi and Shibata (2003b). An idea of dynamically controlling the strength of the bias toward extreme parents was suggested in Ishibuchi and Shibata (2004). Ishibuchi and Narukawa (2005) examined the effect of crossover and mutation on the diversity of solutions and their convergence to the Pareto front using the similarity-based mating scheme.

In this paper, we examine the effect of mating restriction on the search ability of EMO algorithms using the similarity-based mating scheme through computational experiments on a variety of test problems for multiobjective combinatorial optimization. As a representative EMO algorithm, we use NSGA-II of Deb et al. (2002) because of its simplicity, popularity and high search ability. First we examine the effect of recombining similar or dissimilar parents on the performance of NSGA-II. Then we examine the effect of biasing selection probabilities toward extreme solutions. The similarity-based mating scheme is used to recombine extreme and similar parents. Finally we examine the effect of dynamically controlling the strength of the bias toward extreme solutions. The effect of the similarity-based mating scheme is also examined using SPEA of Zitzler and Thiele (1999).

2. Similarity-based mating scheme

In general, a k-objective maximization problem can be written as

$$\text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ ...,\ f_k(\mathbf{x})), \tag{1}$$

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

where $\mathbf{f}(\mathbf{x})$ is the k-dimensional objective vector, $f_i(\mathbf{x})$ is the i-th objective function to be maximized, $\mathbf{x}$ is the decision vector, and $\mathbf{X}$ is the feasible region in the decision space.

Let us denote the distance between two solutions $\mathbf{x}$ and $\mathbf{y}$ as $\| \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}) \|$ in the objective space. In this paper, we measure the distance $\| \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}) \|$ by the Euclidean distance as

$$\| \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}) \| = \sqrt{\{f_1(\mathbf{x}) - f_1(\mathbf{y})\}^2 + \cdots + \{f_k(\mathbf{x}) - f_k(\mathbf{y})\}^2}\ . \tag{3}$$

On the other hand, the definition of the distance $\| \mathbf{x} - \mathbf{y} \|$ in the decision space totally depends on the representation of solutions in a particular problem. Whereas the similarity-based mating scheme can be implemented using the distance in the decision space as well as in the objective space, we mainly use the distance in the objective space due to the simplicity of its definition as the Euclidean distance.

The similarity-based mating scheme of Ishibuchi and Shibata (2003b) is illustrated in Fig. 1. First, a pre-specified number of candidates (say α candidates) are selected from the current population by iterating the standard binary tournament selection procedure α times. Open circles at the bottom of Fig. 1 denote randomly drawn solutions that join the binary tournament selection procedure. Next the average objective vector over the α candidates is calculated in the objective space. The most dissimilar one among the α candidates from the average objective vector is chosen as Parent A in Fig. 1. That is, the most distant candidate from the average objective vector in the objective space is chosen as Parent A. In order to choose a mate for Parent A (i.e., to choose Parent B in Fig. 1), β candidates are selected from the current population by iterating the standard binary tournament selection procedure β times. The most similar one among the β candidates to Parent A is chosen as Parent B in Fig. 1. That is, the closest candidate to Parent A in the objective space is chosen as its mate.

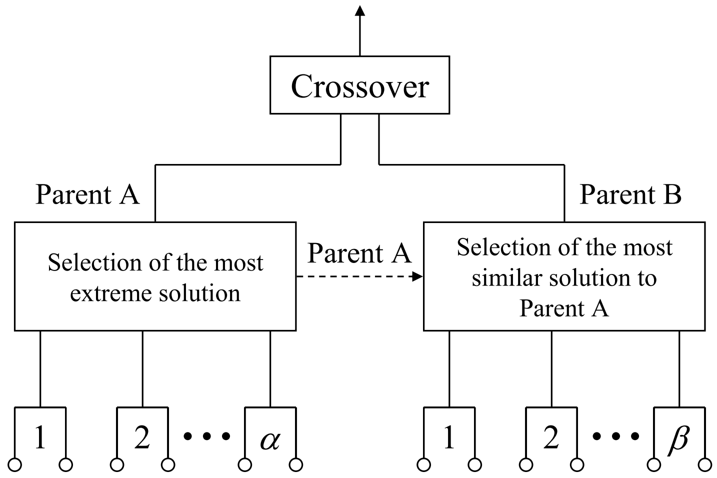

Fig. 1. Similarity-based mating scheme of Ishibuchi and Shibata (2003b).

The original version of the similarity-based mating scheme in Ishibuchi and Shibata (2003a) corresponds to the case of $\alpha = 1$ in Fig. 1. The left-hand side of Fig. 1 was added to the original version in order to choose extreme parents by Ishibuchi and Shibata (2003b). The similarity-based mating scheme can be easily incorporated into almost all EMO algorithms because it needs only the distance between solutions. As we have already mentioned, the distance between solutions can be measured in the decision space as well as in the objective space.

The similarity-based mating scheme in Fig. 1 has the following features:

(a) The strength of the bias toward extreme parents can be easily adjusted by the value of α. Larger values of α mean stronger bias toward extreme parents. When $\alpha = 1$, no bias toward extreme parents is added to EMO algorithms.

(b) The strength of the bias toward similar parents can be easily adjusted by the value of β in the same manner as the adjustment of the bias toward extreme parents using the value of α.

(c) The specification of β is more intuitive than that of the mating radius in mating restriction.

(d) The similarity-based mating scheme can be also used to recombine dissimilar parents by choosing the most dissimilar one as Parent B in Fig. 1.

3. Evolutionary multiobjective optimization

3.1. Performance indices

A number of performance indices have been proposed to evaluate non-dominated solution sets (e.g., see Deb (2001) and Coello et al. (2002)). Those performance indices have been used to compare different non-dominated solution sets (i.e., to compare different EMO algorithms). There is, however, no performance index that can simultaneously measure various aspects of non-dominated solution sets (e.g., the diversity of solutions and the convergence of solutions to the Pareto front) as pointed out in some studies (e.g., Knowles and Corne (2002), Okabe et al. (2003), and Zitzler et al. (2003)). Moreover the use of only a single performance index is sometimes misleading. Thus we use several performance indices in order to evaluate various aspects of non-dominated solution sets.

Let S and S^* be a non-dominated solution set and the Pareto-optimal solution set, respectively. The convergence of the non-dominated solution set S to the Pareto-optimal solution set S^* is measured by the following performance index called the generational distance (Van Veldhuizen (1999)):

$$\mathrm{GD}(S) = \frac{1}{|S|} \sum_{\mathbf{x} \in S} \min\{\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| : \mathbf{y} \in S^*\}, \tag{4}$$

where $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|$ is the Euclidean distance between the two solutions $\mathbf{x}$ and $\mathbf{y}$ in the objective space, and $|S|$ is the number of solutions in S (i.e., $|S|$ is the cardinality of S). The generational distance is the average distance from each solution in S to its nearest Pareto-optimal solution in S^*.

Whereas the generational distance measures the proximity of the non-dominated solution set S to the Pareto-optimal solution set S^*, it can not measure the diversity of solutions. In order to measure not only the convergence but also the diversity, we use the following performance index called the $\mathrm{D1_R}$ measure (Knowles and Corne (2002)):

$$\mathrm{D1_R}(S) = \frac{1}{|S^*|} \sum_{\mathbf{y} \in S^*} \min\{\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| : \mathbf{x} \in S\}. \tag{5}$$

The $\mathrm{D1_R}$ measure is the average distance from each Pareto-optimal solution $\mathbf{y}$ in S^* to its nearest solution in S. Conceptually the same measure was used in a slightly different form by Czyzak and Jaszkiewicz (1998).

Whereas both GD and $\mathrm{D1_R}$ are calculated using the true Pareto-optimal solution set S^*, we usually do not know S^* of each test problem. In our computational experiments, we use as S^* a set of non-dominated solutions among all solutions examined in our computational experiments in this paper. Of course, we use the true Pareto-optimal solution set S^* when it is available.

The diversity of the non-dominated solution set S can be more directly measured by the sum of the range of objective values for each objective function:

$$\mathrm{Range}(S) = \sum_{i=1}^{k} [\max_{\mathbf{x} \in S}\{f_i(\mathbf{x})\} - \min_{\mathbf{x} \in S}\{f_i(\mathbf{x})\}]. \tag{6}$$

This measure is similar to the maximum spread of Zitzler (1999).

In order to measure both the diversity and the convergence, we can also use the hypervolume measure (Zitzler and Thiele (1998)) that calculates the volume of the dominated region by the non-dominated solution set S in the objective space. The boundary of the dominated region in the objective space is called the attainment surface (Fonseca and Fleming (1996)). From multiple attainment surfaces obtained by multiple runs of an EMO algorithm for a multiobjective optimization problem, we can calculate the 50% attainment surface as a kind of their average result. For details of the calculation of the 50% attainment surface, see Fonseca and Fleming (1996) and Deb (2001).

3.2. Evolutionary multiobjective optimization algorithms

We use NSGA-II due to its simplicity, popularity and high search ability. The basic framework of NSGA-II can be written as follows:

[NSGA-II]

Step 1: P := Initialize (P)

Step 2: While the stopping condition is not satisfied, do

Step 3: P' = Parent Selection (P)

Step 4: P'' = Genetic Operations (P')

Step 5: P = Replace ($P \cup P''$)

Step 6: End while

Step 7: Return (P)

In Step 1, the population P is initialized. The initialization is usually performed randomly. In Step 3, a set of pairs of parent solutions (i.e., P') is selected from the current population P. The standard binary tournament selection procedure is usually used to choose a pair of parent solutions. In Step 4, an offspring population P'' is generated from the parent population P' by genetic operations (i.e., crossover and mutation). The size of the offspring population P'' is usually the same as that of the current population P. In Step 5, the best solutions are chosen from the merged population ($P \cup P''$) to construct the next population P. Elitism is realized in this step.

Multiobjective evolution in NSGA-II is mainly driven by choosing better solutions in terms of Pareto dominance relation in the parent selection phase and the generation update phase. A Pareto sorting procedure is used as the primary fitness evaluation criterion. The diversity of solutions is maintained by choosing solutions in less crowded regions in the objective space among solutions with the same rank in terms of Pareto dominance relation. A crowding measure is used as the secondary criterion for diversity maintenance. For details of NSGA-II, see Deb (2001) and Deb et al. (2002).

3.3. Test problems

Zitzler and Thiele (1999) 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 is written as follows:

$$\text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ ...,\ f_k(\mathbf{x}))\,, \tag{7}$$

$$\text{subject to } \sum_{j=1}^{n} w_{ij} x_j \le c_i \,, \quad i = 1, 2, ..., k \,, \tag{8}$$

$$\text{where } f_i(\mathbf{x}) = \sum_{j=1}^{n} p_{ij} x_j \,, \quad i = 1, 2, ..., k \,. \tag{9}$$

In this formulation, **x** is an n-dimensional binary vector (i.e., $x_j = 0$ or $x_j = 1$ for $j = 1, 2, ..., n$), 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 **x** is handled as a binary string of length n.

We denote the k-objective n-item test problem as the k-n problem. Multiobjective 0/1 knapsack problems have been frequently used to examine the performance of EMO algorithms in the literature (e.g., Jaszkiewicz (2001, 2002b), Knowles and Corne (2000), Mumford (2003), and Zydallis and Lamont (2003)). In this paper, we mainly use the 2-500 problem since its Pareto-optimal solution set S^* is known. In some computational experiments, we also use the 3-500 and 4-500 problems to examine the performance of EMO algorithms for many-objective optimization problems.

When an EMO algorithm is applied to the multiobjective 0/1 knapsack problem in (7)-(9), genetic operations often generate infeasible solutions that do not satisfy the constraint conditions in (8). We use a greedy repair method based on a maximum profit/weight ratio as suggested by Zitzler and Thiele (1999). 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 **x** from 1 to 0) in the ascending order of the maximum profit/weight ratio. We implemented the greedy repair in the Lamarckian framework. See Ishibuchi et al. (2005) for Lamarckian repair and Baldwinian repair.

Kumar and Banerjee (2005) used a different two-objective 0/1 knapsack problem, which has only a single knapsack and no constraint condition as follows:

$$\text{Maximize } f_1(\mathbf{x}) = \sum_{j=1}^{n} p_j x_j \,, \tag{10}$$

$$\text{Minimize } f_2(\mathbf{x}) = \sum_{j=1}^{n} w_j x_j \,, \tag{11}$$

where **x** is an n-dimensional binary vector, p_j is the profit of item j, w_j is the weight of item j. The first objective in (10) is the maximization of the total profit whereas the second objective in (11) is the minimization of the total weight.

The two-objective 0/1 knapsack problem in (10)-(11) has two extreme Pareto-optimal solutions. One extreme is the binary vector of all 1's. In this case, the first objective assumes its maximum value. The other extreme is the binary vector of all 0's. In this case, the second objective assumes its minimum value. Thus the two-objective 0/1 knapsack problem of Kumar and Banerjee (2005) has a large Pareto front. We denote the two-objective n-item problem as the 2-n problem. In this paper, we use the 2-500 problem with the same profit and weight of each item as those for the first knapsack of the 2-500 problem of Zitzler and Thiele (1999). The Pareto-optimal solution set of the 2-500 problem of Kumar and Banerjee (2005) is not known. So we use a set of non-dominated solutions among all solutions examined in our computational experiments in this paper as S^*.

Pelikan et al. (2005) used the following two-objective optimization problem called the onemax-zeromax problem to demonstrate the high search ability of their EMO algorithm:

$$\text{Maximize } f_1(\mathbf{x}) = \sum_{j=1}^{n} x_j , \tag{12}$$

$$\text{Maximize } f_2(\mathbf{x}) = \sum_{j=1}^{n} (1 - x_j) . \tag{13}$$

The first objective in (12) is the maximization of the number of 1's in the binary string $\mathbf{x}$ while the second objective in (13) is the maximization of the number of 0's. The onemax-zeromax problem can be viewed as a special case of the two-objective 0/1 knapsack problem of Kumar and Banerjee (2005) with the following specifications of the profit and the weight of each item:

$$p_j = 1 \text{ and } w_j = 1 \text{ for } j = 1, 2, ..., n . \tag{14}$$

It is interesting to note that all feasible solutions of the onemax-zeromax problem are Pareto-optimal solutions. Thus the total number of the Pareto-optimal solutions of the onemax-zeromax problem with n items is 2^n whereas the total number of different Pareto-optimal objective vectors is $(n+1)$. In this paper, we use the 500-item onemax-zeromax problem. This onemax-zeromax problem has 501 different Pareto-optimal objective vectors: (0, 500), (1, 499), (2, 498), ..., (500, 0). The set of these Pareto-optimal objective vectors is used as S^*.

In Ishibuchi and Murata (1998) and Ishibuchi et al. (2003), multiobjective flowshop scheduling problems were used to examine the performance of their memetic EMO algorithms. Each solution of a flowshop scheduling problem with n jobs is represented by a permutation of the given n jobs $\{$J1, J2, ..., J$n\}$. A two-objective test problem is written as

$$\text{Minimize } f_1(\mathbf{x}) = \max\{C_j : j = 1, 2, ..., n\} , \tag{15}$$

$$\text{Minimize } f_2(\mathbf{x}) = \max\{\max\{C_j - d_j, 0\} : j = 1, 2, ..., n\} , \tag{16}$$

where C_j and d_j are the completion time and the due-date of each job, respectively. The first objective is to minimize the makespan while the second objective is to minimize the maximum tardiness.

Each test problem of multiobjective flowshop scheduling has 20 machines and 20, 40, 60 or 80 jobs. We denote the k-objective problem with n jobs as the k-n flowshop scheduling problem. In this paper, we use the 2-80 flowshop scheduling problem. The total number of feasible solutions of the 2-80 flowshop scheduling problem is $80! \cong 7 \times 10^{118}$. The Pareto-optimal solution set of this test problem is not known. So we use a set of non-dominated solutions among all solutions examined in our computational experiments in this paper as S^*.

4. Recombination of similar or dissimilar parents

4.1. Settings of computational experiments

The similarity-based mating scheme is incorporated into NSGA-II. In order to concentrate on the examination of the effect of recombining similar or dissimilar parents, the value of α is always specified as $\alpha = 1$ in this section. The effect of choosing extreme parents (i.e., the effect of α) is examined in the next section. Various values of β are examined in computational experiments in this section (i.e., $\beta = 1, 2, ..., 10$) in combination with the fixed value of α (i.e., $\alpha = 1$). We examine not only the recombination of similar parents but also dissimilar parents. The most similar one among β candidates is chosen as Parent B in our similarity-based mating scheme in the former case whereas the most dissimilar one is chosen in the latter case.

The modified NSGA-II algorithm with the similarity-based mating scheme is applied to the two knapsack problems and the onemax-zeromax problem using the following specifications:

Population size: 200,
Crossover probability: 0.8 (One-point crossover),
Mutation probability: 1/500 per bit (Bit-flip mutation),
Stopping condition: 2000 generations.

On the other hand, the following specifications are used in the flowshop scheduling problem:

Population size: 200,
Crossover probability: 0.8 (Two-point order crossover),
Mutation probability: 0.5 per string (Shift mutation),
Stopping condition: 2000 generations.

For details of genetic operations for flowshop scheduling, see Murata et al. (1996) where a number of different crossover and mutation operations were compared with each other. The best results were obtained from the combination of the two-point order crossover and the shift mutation in Murata et al. (1996). We use this combination of the genetic operations in this paper.

4.2. Results on the 2-500 knapsack problem of Zitzler and Thiele

In order to visually demonstrate the effect of recombining similar or dissimilar parents, we show in Fig. 2 the 50% attainment surface after the 2000th generation over 50 runs of each algorithm: NSGA-II, NSGA-II with the recombination of similar parents using $\beta = 5$, and NSGA-II with the recombination of dissimilar parents using $\beta = 5$. For comparison, the Pareto front of the 2-500 knapsack problem is also shown in Fig. 2.

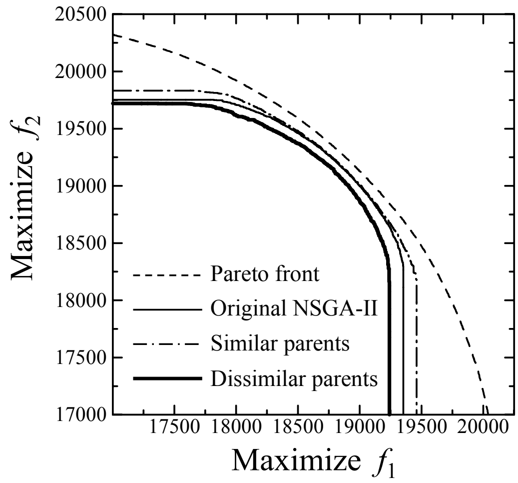

Fig. 2. Average results over 50 runs for the 2-500 knapsack problem of Zitzler and Thiele (1999).

In Fig. 2, the recombination of similar parents has improved the diversity of solutions without degrading the convergence of solutions to the Pareto front. On the other hand, we cannot observe any positive effects of recombining dissimilar parents. While the diversity of solutions has been improved by recombining similar parents in Fig. 2, it is still small if compared with the range of the Pareto front. This is the motivation behind the bias toward extreme solutions, which is examined in the next section.

The effect of recombining similar or dissimilar parents is further examined in Figs. 3-6. Average results over 50 runs for each specification of β are shown in Figs. 3-6 where two cases are examined with respect to the similarity between solutions: the Euclidean distance in the objective space and the Hamming distance in the decision space. In each figure, $\beta = 1$ means the original NSGA-II algorithm. From Figs. 3-6, we can see that the recombination of similar parents has a positive effect on both the diversity of solutions (e.g., Fig. 5) and the convergence of solutions to the Pareto front (e.g., Fig. 3).

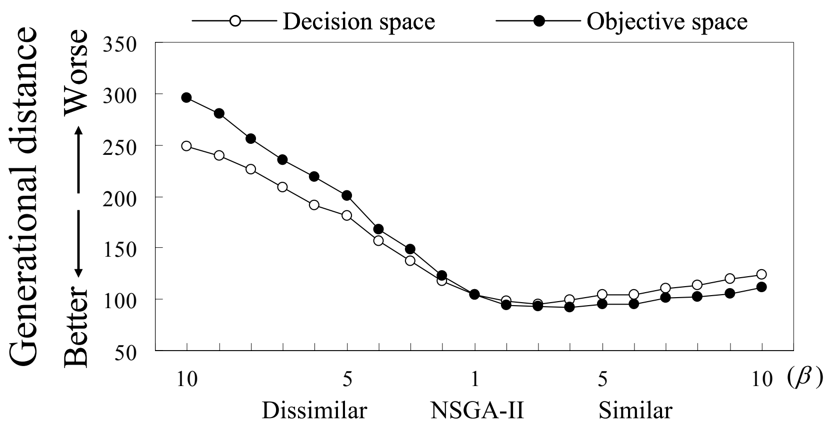

Fig. 3. Generational distance for the 2-500 knapsack problem of Zitzler and Thiele (1999).

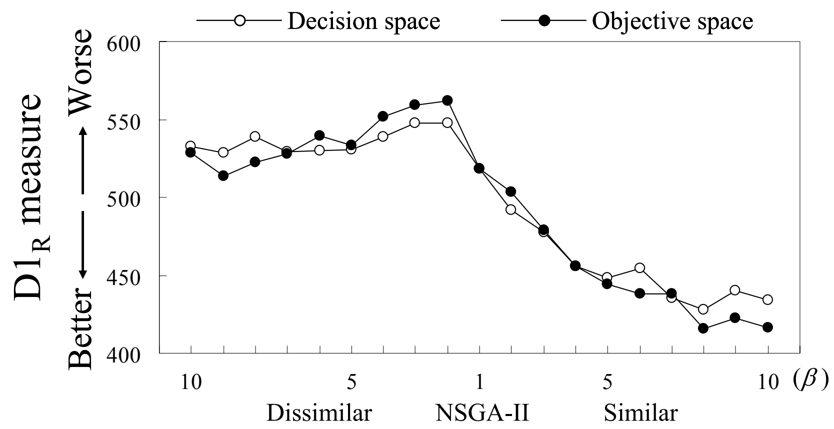

Fig. 4. $D1_R$ measure for the 2-500 knapsack problem of Zitzler and Thiele (1999).

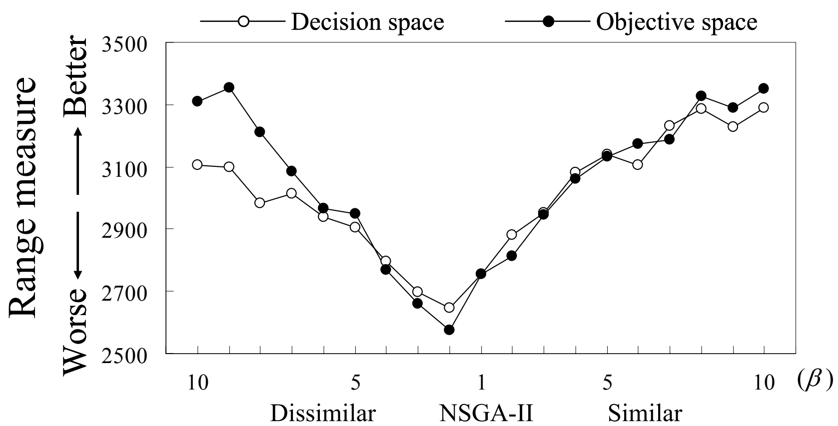

Fig. 5. Range measure for the 2-500 knapsack problem of Zitzler and Thiele (1999).

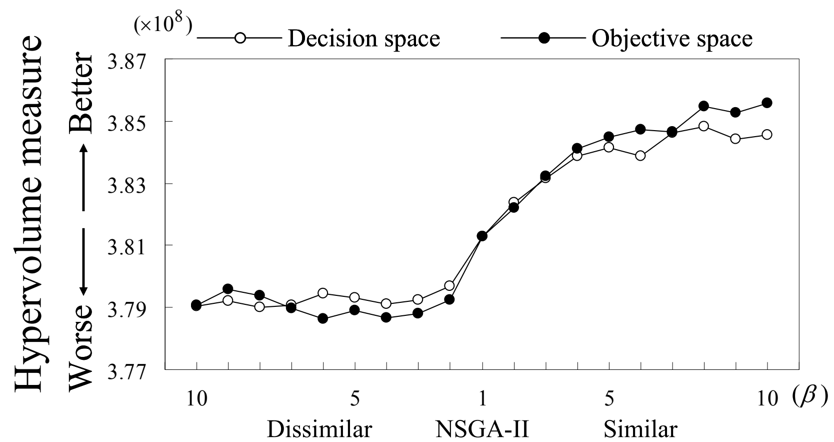

Fig. 6. Hypervolume measure for the 2-500 knapsack problem of Zitzler and Thiele (1999).

Let us examine the statistical significance of the improvement by the similarity-based mating scheme in Figs. 3-6. We compare the two cases: NSGA-II with $\beta = 1$ and NSGA-II with the recombination of similar parents using $\beta = 5$. The average value and the standard deviation (in the parentheses) of each performance index over 50 runs are shown in Table 1. Using Student's *t*-test, we examine the statistical significance of the difference in each performance index between the two cases in Table 1. Only for the hypervolume measure, we use Welch's *t*-test because the standard deviations of this measure are clearly different between the two cases. Based on the statistical tests, we can say with the confidence level 99.9% that the search ability of NSGA-II has been significantly improved by recombining similar parents using the similarity-based mating scheme with $\beta = 5$ in terms of all the four performance indices.

Table 1. The average value and the standard deviation of each measure over 50 runs for the 2-500 knapsack problem of Zitzler and Thiele (1999). The standard deviation is shown in parentheses.

Measure	NSGA-II	Similar Parents
Generational distance	104.67 (12.86)	94.67 (11.57)
$D1_R$	518.06 (39.67)	444.36 (46.04)
Range	2754.28 (217.50)	3133.46 (277.05)
Hypervolume ($\times 10^8$)	3.81 (0.0128)	3.84 (0.0176)

In our experimental results in Figs. 3-6, very similar results were obtained from the two versions of the similarity-based mating scheme: objective space mating and decision space mating. These results suggest that the distance between solutions in the objective space is closely related to the distance in the decision space. In Fig. 7, we show the relation between the Hamming distance in the decision space and the Euclidean distance in the objective space for 200 pairs of solutions. Those pairs were randomly drawn from a population after the 1000th generation when the original NSGA-II algorithm was applied to the 2-500 knapsack problem. From Fig. 7, we can see that the distances in the two spaces are closely related to each other.

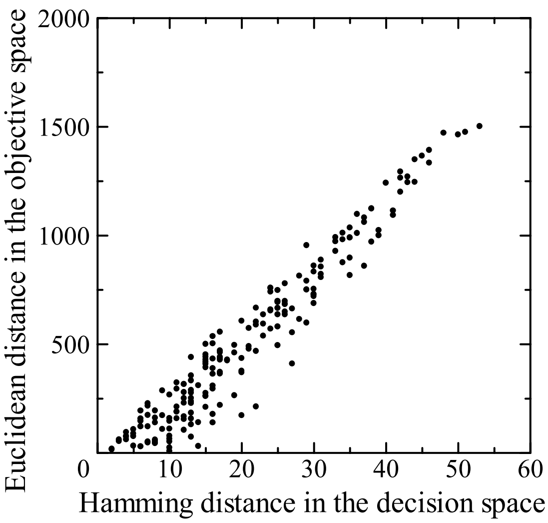

Fig. 7. Relation between the Hamming distance in the decision space and the Euclidean distance in the objective space for 200 pairs of solutions of the 2-500 knapsack problem of Zitzler and Thiele (1999).

4.3. Results on the 2-500 knapsack problem of Kumar and Banerjee

As in Fig. 2, we visually demonstrate the effect of recombining similar or dissimilar parents on the performance of NSGA-II for the 2-500 knapsack problem of Kumar and Banerjee (2005) in Fig. 8. We can see that the performance of NSGA-II has been improved by recombining similar parents in Fig. 8. We can not observe any positive effect of recombining dissimilar parents in Fig. 8.

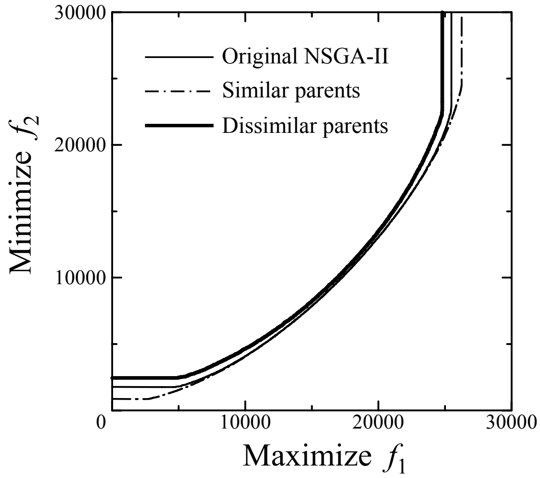

Fig. 8. Average results over 50 runs for the 2-500 knapsack problem of Kumar and Banerjee (2005).

4.4. Results on the onemax-zeromax problem

All feasible solutions of the onemax-zeromax problem are Pareto-optimal solutions. That is, any solution sets are always on the Pareto front. Due to this special feature of solution sets, we only report experimental results with respect to the range measure. Fig. 9 shows average results over 50 runs for each specification of β. We can see from Fig. 9 that the performance of NSGA-II on the onemax-zeromax problem has been improved by recombining similar parents. This improvement is statistically significant. For example, we can say with the confidence level 99.9% based on Student's *t*-test that the difference in the range measure between $\beta = 1$ and $\beta = 5$ (objective space mating) is statistically significant. The average values of the range measure are 829.56 ($\beta = 1$) and 873.46 ($\beta = 5$) while the standard deviations are 13.69 ($\beta = 1$) and 12.03 ($\beta = 5$).

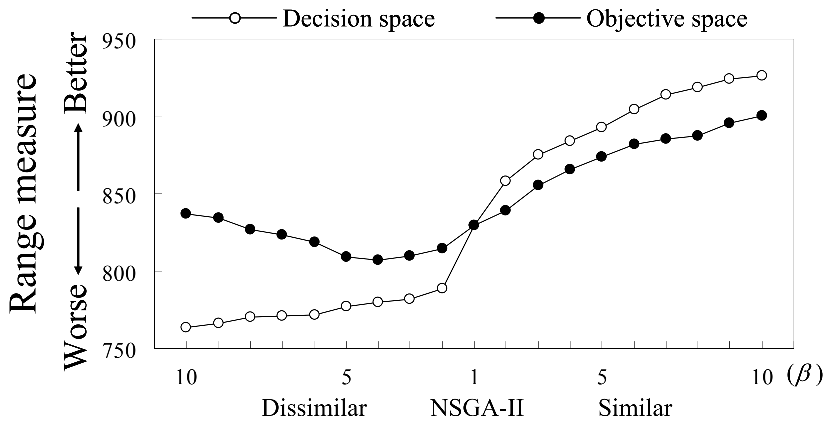

Fig. 9. Range measure for the onemax-zeromax problem.

In Fig. 9, different results were obtained by the two versions of the similarity-based mating scheme (i.e., objective space mating and decision space mating). In Fig. 10, we show the relation between the Hamming distance in the decision space and the Euclidean distance in the objective space for 200 pairs of solutions. Those pairs are randomly drawn from a population after the 1000th generation when the original NSGA-II algorithm is applied to the onemax-zeromax problem. Whereas the two distances were closely related to each other in the 2-500 knapsack problem of Zitzler and Thiele (1999) in Fig. 7, they are not related in Fig. 10. This may lead to experimental results in Fig. 9 where different results were obtained by the two versions of the similarity-based mating scheme.

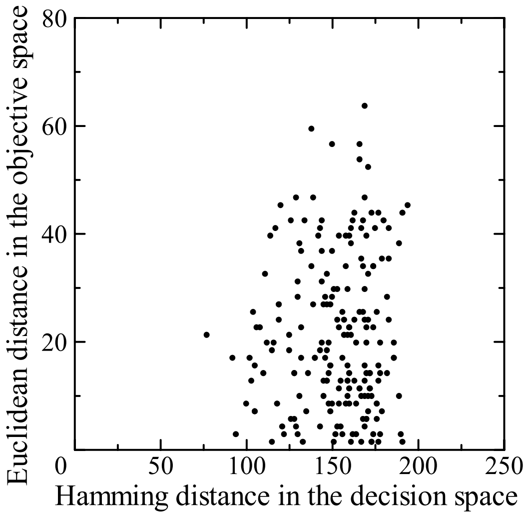

Fig. 10. Relation between the Hamming distance in the decision space and the Euclidean distance in the objective space for 200 pairs of solutions of the onemax-zeromax problem.

4.5. Results on the flowshop scheduling problem

We visually demonstrate the effect of recombining similar or dissimilar parents on the performance of NSGA-II for the 2-80 flowshop scheduling problem in Fig. 11 and Fig. 12. Fig. 11 shows an obtained non-dominated solution set after the 2000th generation by a single run of each algorithm. On the other hand, Fig. 12 shows the 50% attainment surface after the 2000th generation over 50 runs of each algorithm. The value of β is specified as $\beta = 5$ (objective space mating) in Fig. 11 and Fig. 12. We can see that the performance of NSGA-II has been improved by recombining similar parents in Fig. 11 and Fig. 12. An interesting observation in Fig. 11 is that a non-convex solution set is obtained in Fig. 11 from each of the three algorithms.

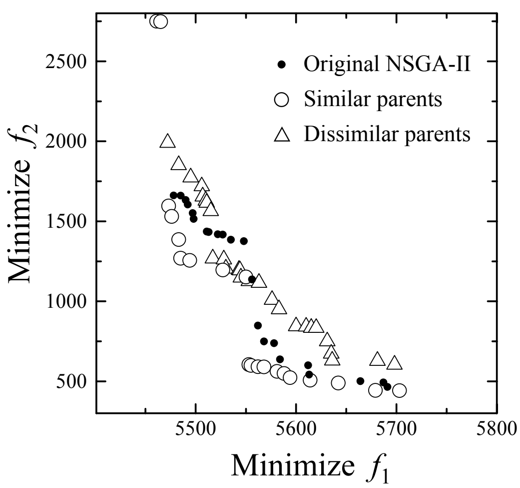

Fig. 11. Results of a single run for the 2-80 flowshop scheduling problem.

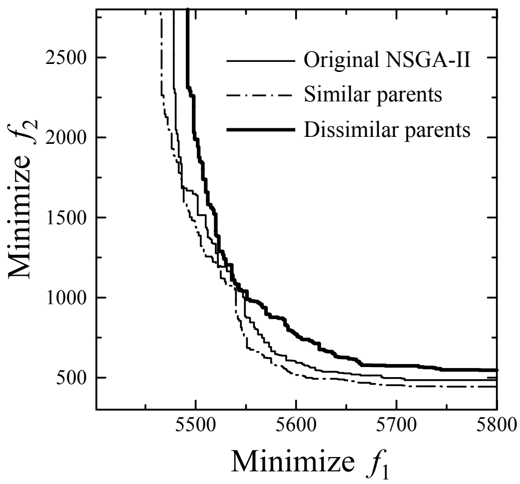

Fig. 12. Average results over 50 runs for the 2-80 flowshop scheduling problem.

4.6. Results on test problems with more than two objectives

In the above-mentioned computational experiments, we visually demonstrated the performance improvement of NSGA-II by the similarity-based mating scheme for the two-objective test problems. In this section, we apply NSGA-II and its modified version to three-objective and four-objective knapsack problems (i.e., 3-500 and 4-500 problems) of Zitzler and Thiele (1999).

Experimental results on each test problem are shown in Fig. 13 and Fig. 14 using the hypervolume measure. We can see from these figures that the similarity-based mating scheme has improved the search ability of NSGA-II for the three-objective and four-objective knapsack problems. The difference in the hypervolume measure between $\beta = 1$ and $\beta = 5$ (objective space mating) is statistically significant with the confidence level 99.9% based on Student's *t*-test in Fig. 13 and Fig. 14.

As pointed out by Hughes (2005), usually EMO algorithms do not work well on many-objective optimization problems with four or more objectives, especially in terms of the convergence of solutions to the Pareto front. Thus we may need other methods for improving the search ability of EMO algorithms on the four-objective knapsack problem. See Hughes (2005), Ishibuchi et al. (2006), Jaszkiewicz (2004), and Purshouse and Fleming (2003) for further discussions on the handling of many-objective optimization problems.

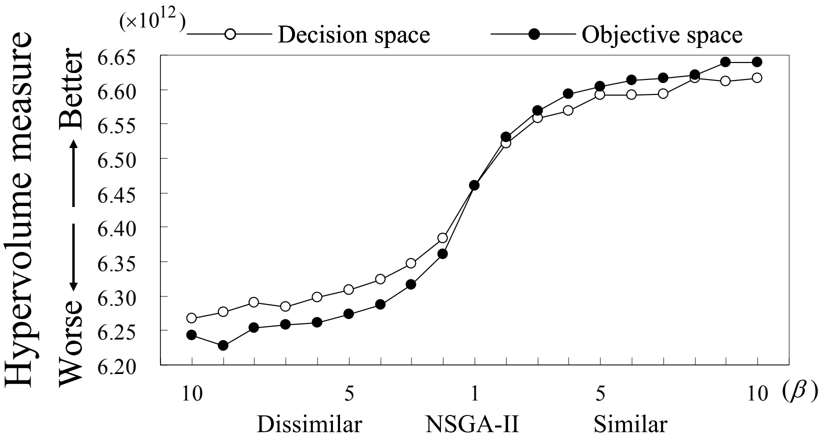

Fig. 13. Hypervolume measure for the 3-500 knapsack problem of Zitzler and Thiele (1999).

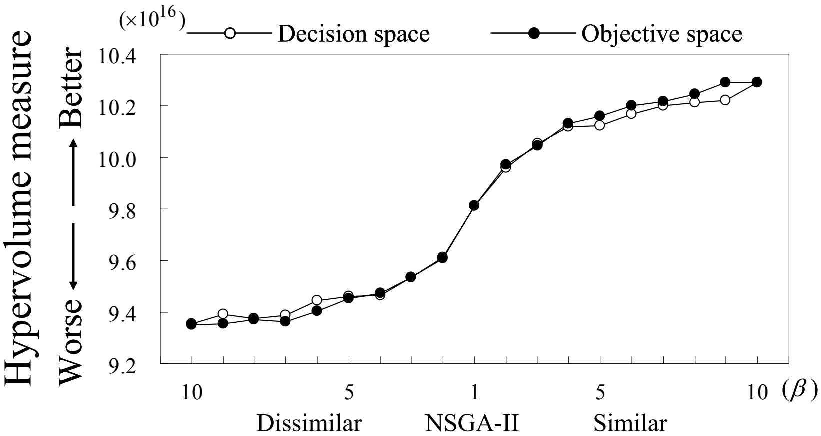

Fig. 14. Hypervolume measure for the 4-500 knapsack problem of Zitzler and Thiele (1999).

5. Recombining extreme and similar parents

5.1. Settings of computational experiments

We use the same parameter specifications as in Section 4 except for the value of α in the similarity-based mating scheme. The value of α can be viewed as the strength of the bias toward extreme parents. Various combinations of α and β are used in this section. The similarity between solutions is measured by the Euclidean distance in the objective space. We do not examine the case of recombining dissimilar parents because it is not useful for our test problems.

5.2. Results on the 2-500 knapsack problem of Zitzler and Thiele

In Fig. 15, we compare the three specifications of α and β (i.e., $(\alpha, \beta) = (1, 1), (1, 5), (5, 5)$) in the similarity-based mating scheme for the 2-500 knapsack problem of Zitzler and Thiele (1999). Fig. 15 shows the 50% attainment surface over 50 runs at the 2000th generation of NSGA-II with each specification of (α, β). In Fig. 15, $(\alpha, \beta) = (1, 5)$ means the recombination of similar parents whereas $(\alpha, \beta) = (5, 5)$ means the recombination of extreme and similar parents. It should be noted that the case of $(\alpha, \beta) = (1, 1)$ is exactly the same as the original NSGA-II algorithm. From Fig. 15, we can see that the recombination of extreme and similar parents (i.e., $(\alpha, \beta) = (5, 5)$) further improves the performance of NSGA-II with the recombination of similar parents (i.e., $(\alpha, \beta) = (1, 5)$).

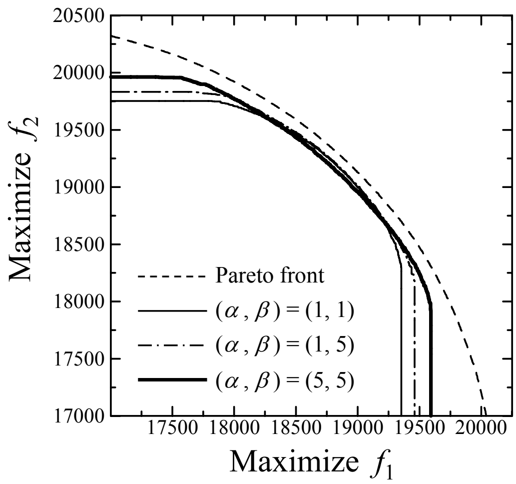

Fig. 15. Average results over 50 runs for the 2-500 knapsack problem of Zitzler and Thiele (1999).

In general, parameter specifications in EMO algorithms (e.g., crossover and mutation probabilities) have large effects on the diversity-convergence balance of multiobjective evolution. Thus we examine the effect of the crossover probability (say P_C) and the mutation probability (say P_M) on the performance of NSGA-II. Our computational experiments are performed using 25 combinations of five crossover probabilities and five mutation probabilities: $P_C = 0.1, 0.2, 0.4, 0.8, 1.0$ and $P_M = 0.001, 0.002, 0.004, 0.008, 0.01$. The original NSGA-II algorithm is used for each combination. For comparison, average results are also calculated for two specifications of the parameters α and β (i.e., $(\alpha, \beta) = (5, 5)$ and $(\alpha, \beta) = (10, 10)$) where the crossover and mutation probabilities are specified as $P_C = 0.8$ and $P_M = 1/500$ as in Fig. 15.

Experimental results are summarized in Figs. 16-19. From these figures, we can see that the effect of recombining extreme and similar parents is much larger than that of the parameter specifications in NSGA-II. For example, the average values of the range measure in Fig. 18 and the hypervolume measure in Fig. 19 are much larger in the case of $(\alpha, \beta) = (10, 10)$ than any other cases of the original NSGA-II algorithm with various specifications of the crossover and mutation probabilities. That is, the effect of recombining extreme and similar parents on the diversity of solutions is much larger than that of the parameter specifications for the crossover and mutation probabilities. Much better results are also obtained in the case of $(\alpha, \beta) = (10, 10)$ in Fig. 17 for the $D1_R$ measure than any other cases of the original NSGA-II algorithm with various specifications of the crossover and mutation probabilities. We can also see that the convergence of solutions to the Pareto front is slightly degraded by the recombination of extreme and similar parents in Fig. 16.

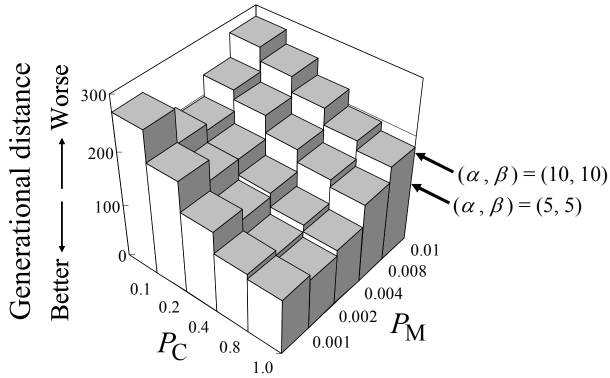

Fig. 16. Generational distance for the 2-500 knapsack problem of Zitzler and Thiele (1999).

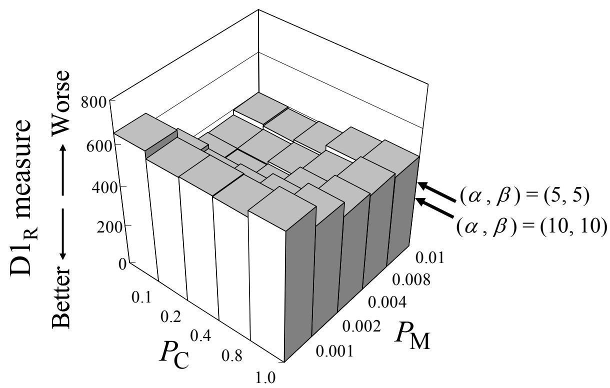

Fig. 17. $D1_R$ measure for the 2-500 knapsack problem of Zitzler and Thiele (1999).

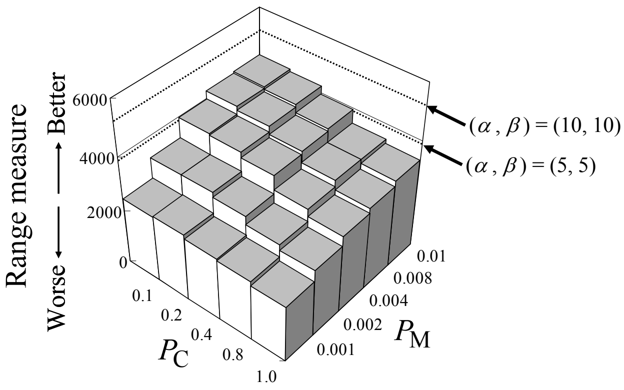

Fig. 18. Range measure for the 2-500 knapsack problem of Zitzler and Thiele (1999).

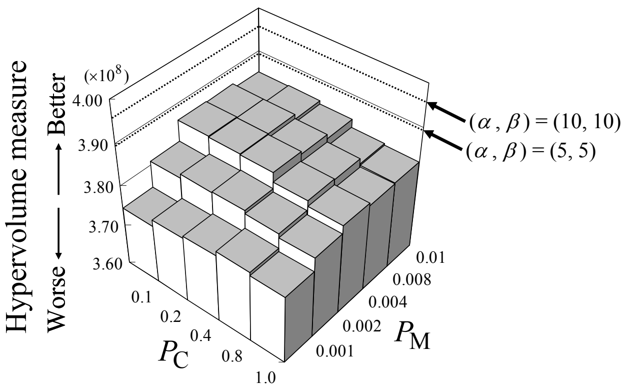

Fig. 19. Hypervolume measure for the 2-500 knapsack problem of Zitzler and Thiele (1999).

5.3. Results on the 2-500 knapsack problem of Kumar and Banerjee

In the same manner as Fig. 15, we show experimental results on the 2-500 knapsack problem of Kumar and Banerjee (2005) in Fig. 20. We can see from Fig. 20 that the diversity of solutions has been improved by recombining extreme and similar parents. We can not observe any negative effect of the recombination of extreme and similar parents on the convergence of solutions to the Pareto front in Fig. 20.

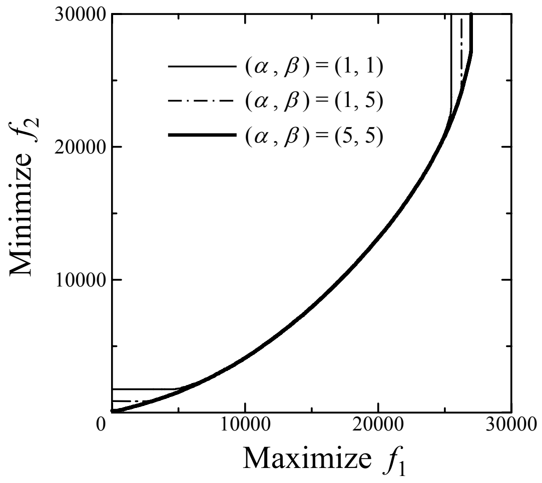

Fig. 20. Average results over 50 runs for the 2-500 knapsack problem of Kumar and Banerjee (2005).

5.4. Results on the onemax-zeromax problem

As we have already explained, any solution sets are always on the Pareto front. Thus we only report experimental results with respect to the range measure. Fig. 21 shows average results over 50 runs for each specification of α and β : $(\alpha, \beta) = (1, 1)$, (5, 1), (5, 5), (10, 10). From Fig. 21, we can see that the diversity of solutions has been improved by recombining extreme and similar solutions. We can also see that the stronger bias toward extreme and similar solutions (i.e., $(\alpha, \beta) = (10, 10)$) has a larger effect on the diversity of solutions.

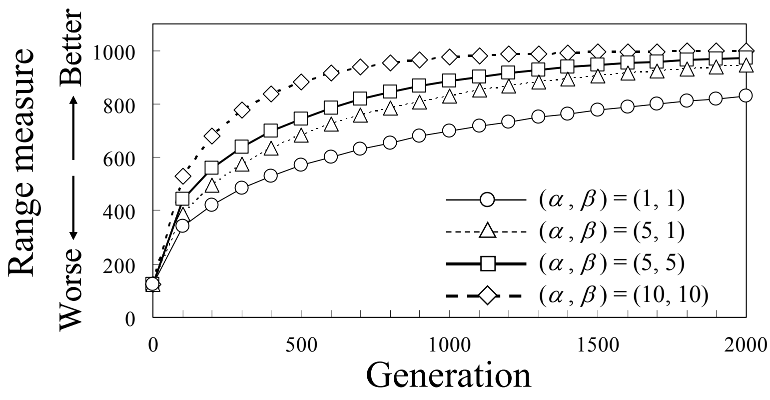

Fig. 21. Range measure for the onemax-zeromax problem.

5.5. Results on the flowshop scheduling problem

In the same manner as Fig. 15 and Fig. 20, we show experimental results on the 2-80 flowshop scheduling problem in Fig. 22. We can see from Fig. 22 that the recombination of extreme and similar parents (i.e., $(\alpha, \beta) = (5, 5)$) has improved the diversity of solutions. We can also see that good non-dominated solutions were not obtained around the center region of the Pareto front (e.g., (5500, 1250) in Fig. 22) when (α, β) was specified as (5, 5). This is due to the bias toward extreme solutions in the similarity-based mating scheme. This issue is further discussed in the next section.

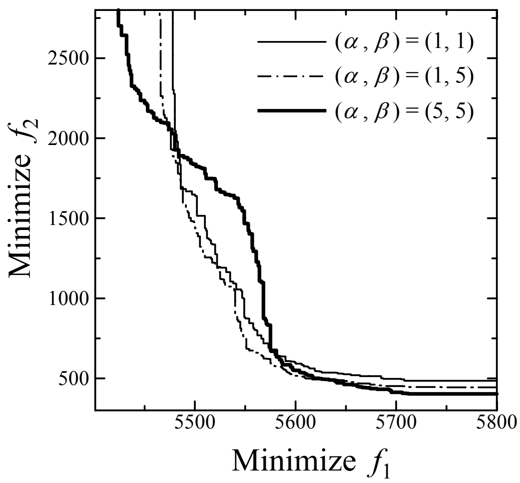

Fig. 22. Average results over 50 runs for the 2-80 flowshop scheduling problem.

6. Dynamic control of mating

As shown in the previous section, the diversity of solutions can be improved by recombining extreme and similar parents. Since there exists a tradeoff between the diversity of solutions and the convergence to the Pareto front in multiobjective evolution, the improvement in the diversity of solutions usually leads to the deterioration in the convergence to the Pareto front. Moreover a strong bias toward extreme solutions using a large value of β is likely to make it difficult for EMO algorithms to find good solutions around the center of the Pareto front as shown in Fig. 22.

Examples of multiobjective evolution are shown in Fig. 23 and Fig. 24 for the 2-500 knapsack problem of Zitzler and Thiele (1999). The two parameters α and β in the similarity-based mating scheme are specified as $(\alpha, \beta) = (1, 1)$ in Fig. 23 and $(\alpha, \beta) = (10, 10)$ in Fig. 24. Each figure shows non-dominated solution sets at the 200th, 400th, 1000th and 2000th generations of a single run of NSGA-II. From the comparison between Fig. 23 and Fig. 24, we can see that the use of large values for α and β significantly improves the diversity of solutions at the cost of the deterioration in the convergence to the Pareto front (especially around the center of the Pareto front).

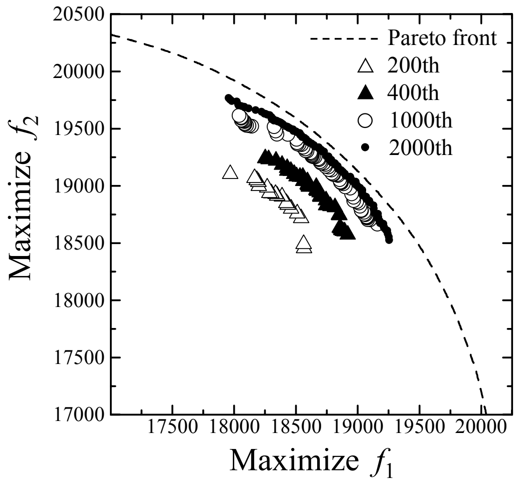

Fig. 23. Results of a single run of NSGA-II with $(\alpha, \beta) = (1, 1)$ for the 2-500 knapsack problem.

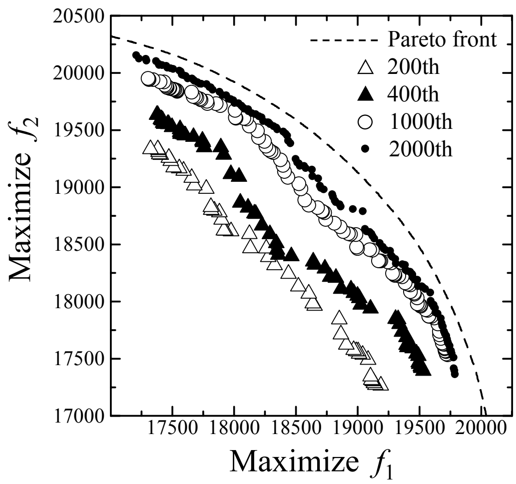

Fig. 24. Results of a single run of NSGA-II with $(\alpha, \beta) = (10, 10)$ for the 2-500 knapsack problem.

Ishibuchi and Shibata (2004) proposed an idea of changing the values of the two parameters α and β during the execution of EMO algorithms. For example, good results with respect to both the diversity and the convergence are obtained by changing the values of α and β at the 1000th generation from $(\alpha, \beta) = (10, 10)$ to $(\alpha, \beta) = (1, 1)$ as shown in Fig. 25. It should be noted that the non-dominated solution sets in Fig. 25 are exactly the same as those in Fig. 24 during the first 1000 generations. Better results are obtained by changing the values of α and β from (10, 10) to (1, 1) than the fixed use of either specification throughout the execution of NSGA-II.

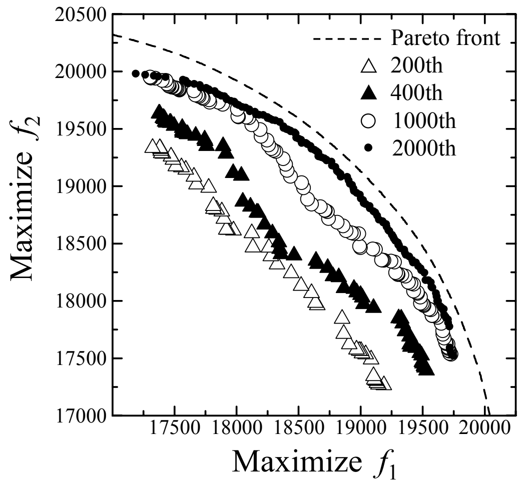

Fig. 25. Results of a single run of NSGA-II with $(\alpha, \beta) = (10, 10)$ in the first 1000 generations and $(\alpha, \beta) = (1, 1)$ in the last 1000 generations.

7. Extensions and future research directions

Let us examine the effect of the similarity-based mating scheme using SPEA of Zitzler and Thiele (1999). SPEA is also a very popular and frequently-used EMO algorithm. High performance of SPEA has been reported in many studies (e.g., Zitzler and Thiele (1999) and Zitzler et al. (2000)).

In the same manner as Fig. 23 and Fig. 25, we show experimental results of a single run of SPEA and its variant with the similarity-based mating scheme in Fig. 26 and Fig. 27, respectively. The same parameter specifications as in the case of NSGA-II are used in these figures for SPEA except for the population size. The size of both the internal and external populations is 100 in SPEA whereas the population size is 200 in NSGA-II (NSGA-II has no external population). From the comparison between Fig. 26 and Fig. 27, we can see that the diversity of solutions has been drastically improved by the similarity-based mating scheme. Such a drastic improvement, however, is not always observed. The distribution of the values of the range measure over 100 runs of SPEA and SPEA with the similarity-based mating scheme is shown in Fig. 28. The value of the range measure in Fig. 28 has been improved from about 2000 to about 5500 by the similarity-based mating scheme in many cases (e.g., more than 4000 in 61 runs among the 100 runs of SPEA with the similarity-based mating scheme). Such a drastic improvement, however, is not observed in the other runs in Fig. 28. We perform the same computational experiments using NSGA-II. The diversity of solutions has been drastically improved by the similarity-based mating scheme in all the 100 runs of NSGA-II as shown in Fig. 29.

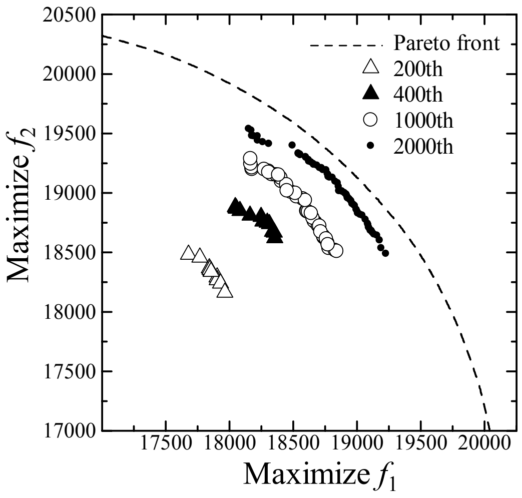

Fig. 26. Results of a single run of the original SPEA algorithm with $(\alpha, \beta) = (1, 1)$.

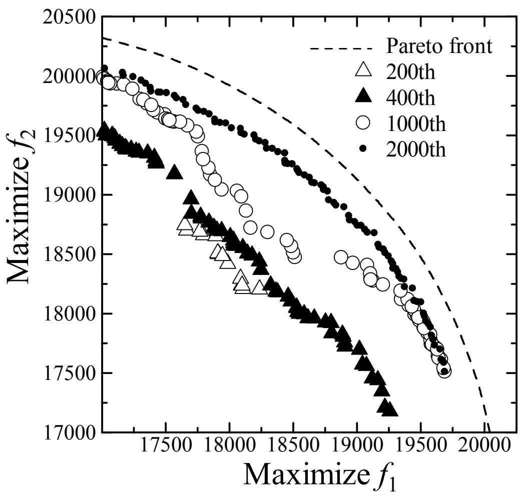

Fig. 27. Results of a single run of the modified SPEA algorithm with $(\alpha, \beta) = (10, 10)$ in the first 1000 generations and $(\alpha, \beta) = (1, 1)$ in the last 1000 generations.

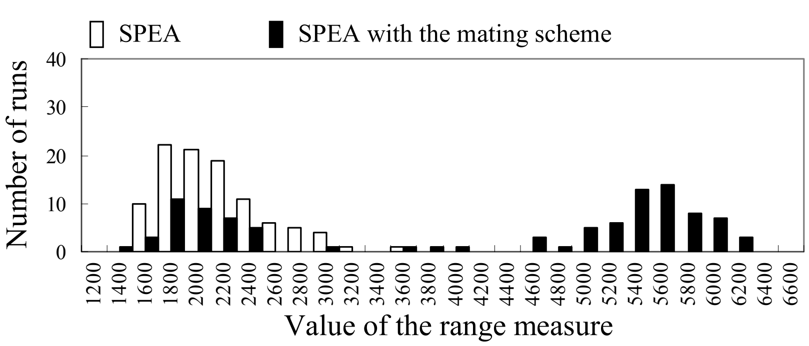

Fig. 28. Distribution of the values of the range measure over 100 runs of SPEA and its variants with the similarity-based mating scheme.

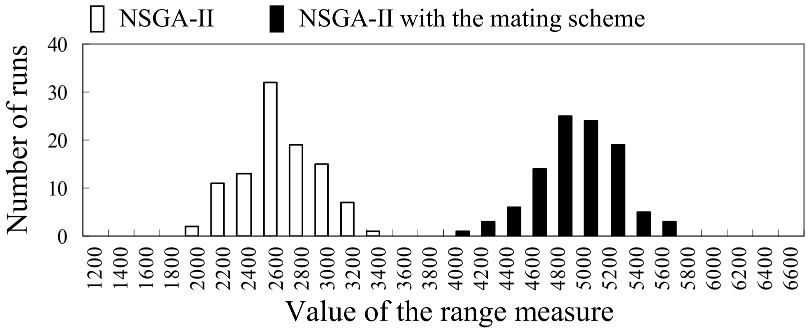

Fig. 29. Distribution of the values of the range measure over 100 runs of NSGA-II and its variants with the similarity-based mating scheme.

Let us discuss why the similarity-based mating scheme does not always work well for SPEA. The choice of extreme parents by the similarity-based mating scheme is to widen a population along the Pareto front as typically observed from the comparison between Fig. 26 and Fig. 27. When we have a population whose solutions are distributed along the Pareto front, the choice of extreme parents actually widens the population as intended. On the other hand, when we have a long population as in Fig. 30 rather than a wide population, poor solutions far from the Pareto front are more likely to be selected as parents by the similarity-based mating scheme. As a result, the choice of extreme parents does not work as intended (i.e., it does not widen the population but just slows down the convergence of solutions to the Pareto front). In this case, the incorporation of the similarity-based mating scheme into SPEA does not improve its performance. Such an undesired effect of the similarity-based mating scheme was dominant in about 40 runs out of the 100 runs of SPEA with the similarity-based mating scheme in Fig. 28 whereas a drastic diversity improvement was observed in the other runs.

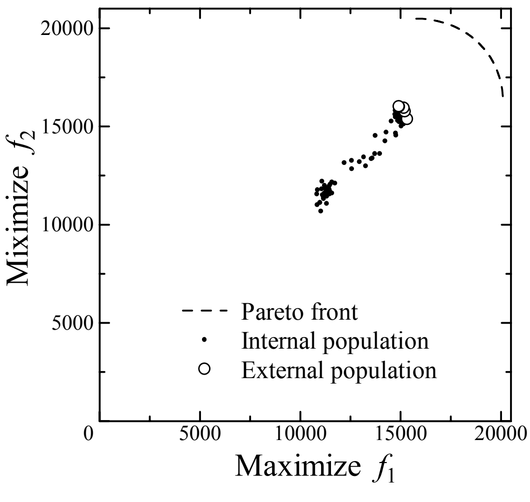

Fig. 30. Solutions in the 10th generation of a single run of SPEA with the similarity-based mating scheme with $(\alpha, \beta) = (10, 10)$.

In order to remove the above-mentioned undesired effect of the similarity-based mating scheme, we modify the procedure for the choice of the first parent (i.e., Parent A in Fig. 1) as follows: Candidates dominated by the average vector over α candidates are not chosen as Parent A. That is, we choose the most dissimilar candidate from the center vector as Parent A after we remove the dominated candidates by the average vector. This modification has improved the performance of SPEA with the similarity-based mating scheme as shown in Fig. 31 where we observe a drastic increase in the range measure in almost all runs of SPEA with the modified mating scheme. The modified mating scheme still works very well for NSGA-II. Almost the same results as Fig. 29 are obtained by NSGA-II with the modified mating scheme.

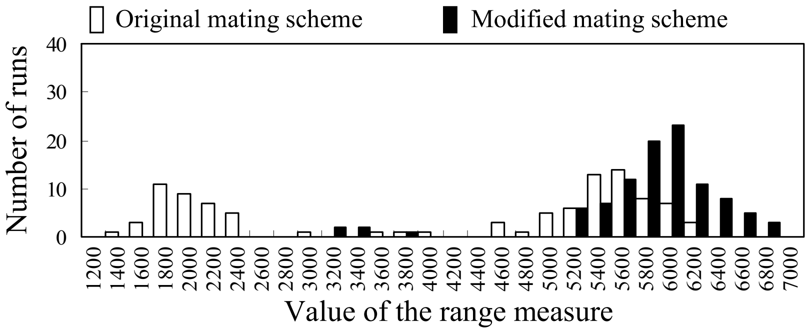

Fig. 31. Distribution of the values of the range measure over 100 runs of SPEA with the similarity-based mating scheme before and after its modification.

Even after the modification, the similarity-based mating scheme does not always work well on any multiobjective optimization problems. In some cases, the bias toward extreme solutions together with the recombination of similar parents may degrade the performance of EMO algorithms especially the convergence of solutions to the center region of the Pareto front as shown in Fig. 22 and Fig. 24. Fig. 32 is a typical example of such a multiobjective optimization problem where the bias toward extreme solutions prevents EMO algorithms from finding good non-dominated solutions along the entire Pareto front. When we have a strong bias toward extreme solutions together with a strong bias toward the recombination of similar parents in Fig. 32, it is very difficult for EMO algorithms to find good non-dominated solutions in the center region marked B. In this case, the population will be divided into two sub-populations in the two extreme regions marked A and C. More sophisticated extensions to the similarity-based mating scheme are needed for handling multiobjective optimization problems such as Fig. 32. Even in this case, the recombination of similar parents with no bias toward extreme solutions may work well as shown for the two-objective flowshop scheduling problems in Fig. 12.

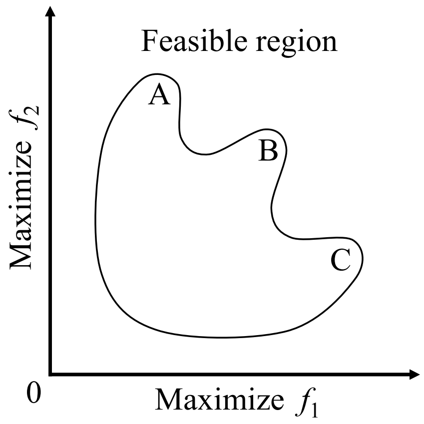

Fig. 32. Illustration of multiobjective optimization problems where the bias toward extreme solutions together with the recombination of similar parents has a bad effect on the performance of EMO algorithms.

8. Conclusion

In this paper, we first demonstrated that the performance of NSGA-II was improved by recombining similar parents. More specifically, the recombination of similar parents improved the diversity of solutions without degrading their convergence to the Pareto front. The strength of the bias toward similar (or dissimilar) solutions can be specified by the parameter β in the similarity-based mating scheme. Next we demonstrated that the performance of NSGA-II was further improved by recombining extreme and similar parents. It was shown that the bias toward extreme solutions has a positive effect on the diversity of solutions and a negative effect on the convergence to the Pareto front (especially around the center of the Pareto front). The strength of the bias toward extreme solutions can be specified by the parameter α in the similarity-based mating scheme. Experimental results also showed the existence of the tradeoff between the diversity of solutions and the convergence to the Pareto front. Finally we demonstrated that good solution sets with respect to both the diversity and the convergence were obtained by changing the values of α and β during the execution of NSGA-II.

Experimental results in this paper demonstrated the usefulness of mating restriction especially with respect to the diversity of solutions. It was shown that the similarity-based mating scheme is a simple but powerful method for mating restriction. Of course, the similarity-based mating scheme does not always improve the performance of any EMO algorithms. Whereas the performance of NSGA-II was always improved by the similarity-based mating scheme as shown in Fig. 29, we did not observe the performance improvement of SPEA in some cases as shown in Fig. 28. After examining why the similarity-based mating scheme did not always work well for SPEA, we proposed a modified mating scheme using Pareto dominance relation between candidate solutions and its center vector. The modified mating scheme almost always improved the performance of SPEA in terms of the diversity of solutions as shown in Fig. 31.

Acknowledgment

This work was partially supported by Japan Society for the Promotion of Science (JSPS) through Grand-in-Aid for Scientific Research (B): KAKENHI (17300075).

References

Coello, C. A. C., Lamont, G. B., 2004. Applications of Multi-Objective Evolutionary Algorithms. World Scientific, Singapore.

Coello, C. A. C., Van Veldhuizen, D. A., Lamont, G. B., 2002. Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer, Boston.

Czyzak, P., Jaszkiewicz, A., 1998. Pareto-simulated annealing - A metaheuristic technique for multi-objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis 7 (1), 34-47.

Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester.

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

Fonseca, C. M., Fleming, P. J., 1993. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. Proc. of 5th International Conference on Genetic Algorithms (ICGA 1993), 416-423.

Fonseca, C. M., Fleming, P. J., 1995. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 3 (1), 1-16.

Fonseca, C. M., Fleming, P. J., 1996. On the performance assessment and comparison of stochastic multiobjective optimizers. Proc. of International Conference on Parallel Problem Solving from Nature (PPSN IV), 584-593.

Goldberg, D. E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading.

Hajela, P., Lin, C. Y., 1992. Genetic search strategies in multicriterion optimal design. Structural Optimization 4, 99-107.

Horn, J., Nafpliotis, N., Goldberg, D. E., 1994. A niched Pareto genetic algorithm for multi-objective optimization. Proc. of 1st IEEE International Conference on Evolutionary Computation (ICEC 1994), 82-87.

Hughes, E. J., 2005. Evolutionary many-objective optimization: many once or one many? Proc. of

2005 Congress on Evolutionary Computation (CEC 2005), 222-227.

Ishibuchi, H., Doi, T., Nojima, Y., 2006. Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms. Lecture Notes in Computer Science 4193 (PPSN IX), 493-502.

Ishibuchi, H., Kaige, S., Narukawa, K., 2005. Comparison between Lamarckian and Baldwinian repair on multiobjective 0/1 knapsack problems. Lecture Notes in Computer Science 3410 (EMO 2005), 370-385.

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

Ishibuchi, H., Narukawa, K., 2005. Recombination of similar parents in EMO algorithms. Lecture Notes in Computer Science 3410 (EMO 2005), 265-279.

Ishibuchi, H., Shibata, Y., 2003a. An empirical study on the effect of mating restriction on the search ability of EMO algorithms. Lecture Notes in Computer Science 2632 (EMO 2003), 433-447.

Ishibuchi, H., Shibata, Y., 2003b. A similarity-based mating scheme for evolutionary multiobjective optimization. Lecture Notes in Computer Science 2723 (GECCO 2003), 1065-1076.

Ishibuchi, H., Shibata, Y., 2004. Mating scheme for controlling the diversity-convergence balance for multiobjective optimization. Lecture Notes in Computer Science 3102 (GECCO 2004), 1259-1271.

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

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

Jaszkiewicz, A., 2002a. Genetic local search for multi-objective combinatorial optimization. European Journal of Operational Research 137 (1), 50-71.

Jaszkiewicz, A., 2002b. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - A comparative experiment. IEEE Transactions on Evolutionary Computation 6 (4), 402-412.

Jaszkiewicz, A., 2004. On the computational efficiency of multiple objective metaheuristics: The knapsack problem case study. European Journal of Operational Research 158, 418-433.

Kim, M., Hiroyasu, T., Miki, M., Watanabe, S., 2004. SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2. Lecture Notes in Computer Science 3242 (PPSN VIII), 742-751.

Knowles, J. D., Corne, D. W., 2000. A comparison of diverse approaches to memetic multiobjective combinatorial optimization. Proc. of 2000 Genetic and Evolutionary Computation Conference Workshop Program: WOMA I (GECCO 2000), 103-108.

Knowles, J. D., Corne, D. W., 2002. On metrics for comparing non-dominated sets. Proc. of 2002 Congress on Evolutionary Computation (CEC 2002), 711-716.

Kumar, R., Banerjee, N., 2005. Running time analysis of a multiobjective evolutionary algorithm on simple and hard problems. Lecture Notes in Computer Science 3469 (FOGA 2005), 112-131.

Mumford, C. L., 2003. Comparing representations and recombination operators for the multi-objective 0/1 knapsack problem. Proc. of 2003 Congress on Evolutionary Computation (CEC 2003), 854-861.

Murata, T., Ishibuchi, H., Tanaka, H., 1996. Genetic algorithms for flowshop scheduling problems. Computer and Industrial Engineering 30 (4), 1061-1071.

Okabe, T., Jin, Y., Sendhoff, B., 2003. A critical survey of performance indices for multi-objective optimization. Proc. of 2003 Congress on Evolutionary Computation (CEC 2003), 878-885.

Pelikan, M., Sastry, K., Goldberg, D. E., 2005. Multiobjective hBOA, clustering, and scalability. Proc. of 2005 Genetic and Evolutionary Computation Conference (GECCO 2005), 663-670.

Purshouse, R. C., Fleming, P. J., 2003. Evolutionary many-objective optimization: An exploratory analysis. Proc. of 2003 Congress on Evolutionary Computation (CEC 2003), 2066-2073.

Sato, H., Aguirre, H., Tanaka, K., 2004. Effects from local dominance and local recombination in enhanced MOEAs. Proc. of 5th International Conference on Simulated Evolution and Learning (SEAL 2004) CD-ROM Proceedings.

Schaffer, J. D., 1985. Multiple objective optimization with vector evaluated genetic algorithms. Proc. of 1st International Conference on Genetic Algorithms and Their Applications (ICGA 1985), 93-100.

Van Veldhuizen, D. A., 1999. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph. D dissertation, Air Force Institute of Technology, Dayton.

Van Veldhuizen, D. A., Lamont, G. B., 2000. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. Evolutionary Computation 8, 125-147.

Watanabe, S., Hiroyasu, T., Miki, M., 2002. Neighborhood cultivation genetic algorithm for multi-objective optimization problems. Proc. of 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2002), 198-202.

Zitzler, E., 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph. D dissertation, Swiss Federal Institute of Technology, Zurich, Shaker Verlag, Aachen.

Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation 8 (2), 173-195.

Zitzler, E., Thiele, L., 1998. Multiobjective optimization using evolutionary algorithms – A comparative case study. Proc. of 5th International Conference on Parallel Problem Solving from Nature (PPSN V), 292-301.

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

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., da Fonseca V. G., 2003. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7 (2), 117-132.

Zydallis, J. B., Lamont, G. B., 2003. Explicit building-block multiobjective evolutionary algorithms for NPC problems. Proc. of 2003 Congress on Evolutionary Computation (CEC 2003), 2685-2695.