Generalization of Dominance Relation-Based Replacement Rules for Memetic EMO Algorithms

Tadahiko Murata [1], Shiori Kaige [2], and Hisao Ishibuchi [2]

[1] Department of Informatics, Kansai University
2-1-1 Ryozenji-cho, Takatsuki, Osaka 569-1095, Japan
murata@res.kutc.kansai-u.ac.jp,
http: www.res.kutc.kansai-u.ac.jp/~murata/

[2] Department of Industrial Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
{hisaoi, shior}@ie.osakafu-u.ac.jp,
http://www.ie.osakafu-u.ac.jp~hisaoi/ci_lab_e/

Abstract. In this paper, we generalize the replacement rules based on the dominance relation in multiobjective optimization. Ordinary two replacement rules based on the dominance relation are usually employed in a local search (LS) for multiobjective optimization. One is to replace a current solution with a solution which dominates it. The other is to replace the solution with a solution which is not dominated by it. The movable area in the LS with the first rule is very small when the number of objectives is large. On the other hand, it is too huge to move efficiently with the latter. We generalize these extreme rules by counting the number of improved objectives in a candidate solution for LS. We propose a LS with the generalized replacement rule for existing EMO algorithms. Its effectiveness is shown on knapsack problems with two, three, and four objectives.

1 Introduction

Since Schaffer's study [1], evolutionary algorithms have been applied to various multiobjective optimization problems for finding their Pareto-optimal solutions. Recently evolutionary algorithms for multiobjective optimization are often referred to as EMO (evolutionary multiobjective optimization) algorithms. The task of EMO algorithms is to find Pareto-optimal solutions as many as possible. In recent studies (e.g., [2-6]), emphasis was placed on the convergence speed to the Pareto-front as well as the diversity of solutions. In those studies, some form of elitism was used as an important ingredient of EMO algorithms. It was shown that use of elitism improved the convergence speed to the Pareto-front [5].

One promising approach for improving the convergence speed to the Pareto-front is the use of local search in EMO algorithms. Hybridization of evolutionary algorithms with local search has already been investigated for single-objective optimization problems in many studies (e.g., [7], [8]). Such a hybrid algorithm is often referred to as a memetic algorithm. See Moscato [9] for an introduction to this field and [10]–[12] for recent developments. The hybridization with local search for multiobjective

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 1234–1245, 2003.

optimization was first implemented in [13], [14] as a multiobjective genetic local search (MOGLS) algorithm where *a scalar fitness function with random weights* was used for the selection of parents and the local search for their offspring. Jaszkiewicz [15] improved the performance of the MOGLS by modifying its selection mechanism of parents. While his MOGLS still used the scalar fitness function with random weights in selection and local search, it did not use the roulette wheel selection over the entire population. A pair of parents was randomly selected from a pre-specified number of the best solutions with respect to the scalar fitness function with the current weights. This selection scheme can be viewed as a kind of mating restriction in EMO algorithms. Knowles & Corne [16] combined their Pareto archived evolution strategy (PAES [2], [4]) with a crossover operation for designing a memetic PAES (M-PAES) [17]. In their M-PAES, *the Pareto-dominance relation* and *the grid-type partition of the objective space* were used for determining the acceptance (or rejection) of new solutions generated in genetic search and local search. The M-PAES had a special form of elitism inherent in the PAES. In those studies, the M-PAES was compared with the PAES, the MOGLS of Jaszkiewicz [15], and an EMO algorithm. In the above-mentioned hybrid EMO algorithms (i.e., multiobjective memetic algorithms [13]-[17]), local search was applied to individuals in *every generation*. In some studies [18], [19], local search is restrictedly applied to every generation by limited local search *only to non-dominated solutions* [18] or introducing *the tournament selection* and *the selection probability* of candidate solutions for local search [19]. Another way of application of local search is proposed in [20], [21], where local search was applied to individuals only in the final generation.

In order to design a local search for multiobjective optimization, a rule for replacing a current solution with another solution should be defined in advance. Murata *et al.* [22] showed experimental results on pattern classification problems where a scalar fitness function-based replacement rule was better than the dominance relation-based replacement rules. In the dominance relation-based replacement rules, the current solution is replaced with a solution which dominates the current one or a solution which is at least a non-dominated solution with the current one. Ishibuchi et al. [23] also pointed this matter by experimental results on scheduling problems. As mentioned in [17], [22] and [23], the replacement rule to accept a non-dominated solution has a weak search pressure since almost all pairs of solutions (a current solution and a candidate solution) will be non-dominated with respect to each other especially in problems with a large number of objectives. On the other hand, the replacement rule to accept a dominating solution does not work well because it is difficult to find a dominating solution for the current solution.

We generalize the replacement rules based on the dominance relation by counting the number of better objective values. Details of the dominance relation-based replacement rules are shown in the next section. We employ a local search using the proposed replacement rule to improve the performance of existing EMO algorithms such as SPEA [3] and NSGA-II [6]. We apply them with the local search to multi-objective knapsack problems as benchmark problems [3], and show the effectiveness of our generalization.

2 Dominance Relation-Based Replacement Rules

2.1 Previous Extensions of Dominance Relation

In order to improve the performance of local search using the dominance relation, several ideas to extend it have been already proposed in [17], [24], and [25]. Knowles and Corne [17] proposed a replacement rule by which a current solution is replaced with a non-dominated solution if it dominates other solutions in the set of non-dominated solutions obtained so far. Ikeda et al. [24] proposed their "α-dominance" where a small detriment in one or several of the objectives is permitted if an attractive improvement in the other objective(s) is achieved. While these two methods try to extend the area of dominating solutions of the current solution, Laumanns et al. [25] proposed their "ε-dominance" where a solution with a small improvement in every normalized objective does not dominate the current one.

Each of these three methods can be considered as an extension of the dominance relation. Before explaining these extensions, we show the dominance relation defined in multiobjective optimization. Without loss of generality, we assume the following N-objective maximization problem:

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

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

where $\mathbf{z}$ is the objective vector with N objectives to be maximized, $\mathbf{x}$ is the decision vector, and $\mathbf{X}$ is the feasible region in the decision space. A solution $\mathbf{x} \in \mathbf{X}$ is said to dominate another solution $\mathbf{y} \in \mathbf{X}$ if the following two conditions are satisfied.

$$f_i(\mathbf{x}) \geq f_i(\mathbf{y})\,,\ \forall i \in \{1, 2, ..., N\}\,, \tag{3}$$

$$f_i(\mathbf{x}) > f_i(\mathbf{y})\,,\ \exists i \in \{1, 2, ..., N\}\,. \tag{4}$$

If there is no solution which dominates $\mathbf{x}$ in $\mathbf{X}$, $\mathbf{x}$ can be said to be a Pareto-optimal solution. Fig. 1 shows that there are four areas of candidate solutions for the solution $\mathbf{x}$ in the case of two-objective problems. When we employ this dominance relation in local search, two replacement rules can be used in the local search as follows:

Rule A: Move to dominating solutions:
Replace the solution $\mathbf{x}$ with a solution which dominates it (Area A in Fig. 1).
Rule B: Move to non-dominated solutions:
Replace the solution $\mathbf{x}$ with a solution which is not dominated by $\mathbf{x}$ (Areas A - C).

The movable area in the local search with Rule A is very small when the number of objectives is large. On the other hand, it is too huge to move efficiently with Rule B. Therefore some extensions for the dominance relation should be considered.

As shown in Fig. 2, Knowles and Corne [17] extended the area of dominating solutions using non-dominated solutions obtained so far. Fig. 3 shows the dominating area of the current one defined by the α-dominance relation [24]. While these two methods enlarge the dominating area of the current solution, the ε-dominance relation [25] reduces the dominating area as shown in Fig. 4. Since the aim of the ε-dominance is reducing the number of non-dominated solutions obtained by this dominance relation, an opposite strategy is used. However, we can see that the area of non-dominated solutions (B and C) is also reduced by the hatched area which consists of dominated solutions (D in Fig. 4). Therefore we can see that this is also a method to reduce the area of non-dominated solutions.

As we observe from Figs. 2 - 4, the area of non-dominated solutions is reduced by these three methods. However, each of them needs more computational efforts. The method in [17] needs to compare the candidate solution with non-dominated solutions. Its performance may depend on the quality of the obtained set of non-dominated solutions. As for the α-dominance [24], the decision maker (DM) should define parameters β and γ for every pair of objectives in advance. The DM should also define a parameter ε in advance to use the ε-dominance [25], and ε should be determined carefully since a large ε makes a solution obtained by this method far from the true Pareto-front.

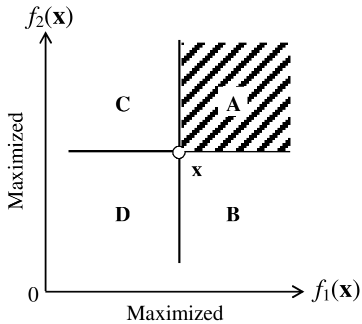

Fig. 1. The area of candidate solutions which replace the current solution **x** by the dominance relation.

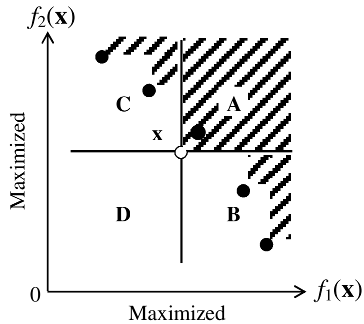

Fig. 2. The area of candidate solutions which replace the current solution **x** by the method in [17].

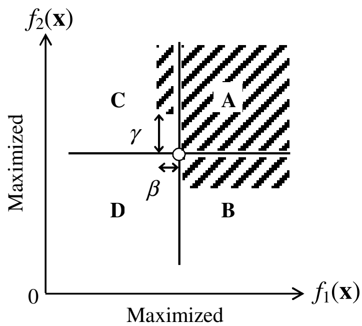

Fig. 3. The area of candidate solutions which replace the current solution **x** by the α-dominance [24].

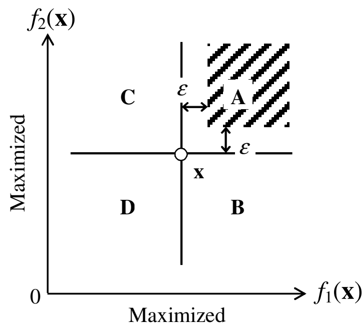

Fig. 4. The area of candidate solutions which replace the current solution **x** by the ε-dominance [25].

2.2 Generalization of the Dominance Relation

In this section, we generalize the two replacement rules (i.e., Rules A and B) shown in the previous section by counting the number of improved objectives. Fig. 5 shows that there are eight possible spaces for the solution **x** in the case of three-objective problems. Every solution in Space A dominates the solution **x**. On the other hand, the solution **x** dominates solutions in Space H. Therefore Rule A allows the current solution to move to a solution in only one space, Space A. On the other hand, Rule B enables it to move to neighborhood solutions in all spaces except a dominated space H in Fig. 5. This means that $(2^N - 1)$ spaces are allowed out of 2^N spaces in Rule B. We can see that the number of accepted spaces is extreme in each of both cases. That is, while the number of accepted spaces is only one as for Rule A, it is $(2^N - 1)$ for Rule B. We generalize these two extreme cases by counting the number of improved objectives.

The number of improved objectives for the solution **x** is different in each space. For example, Fig. 5 shows that the number of improved objectives for a solution in Space A is three. It is zero for a solution in Space H. There are other spaces where the number of improved objectives is one or two. That is, Spaces B, C, and E have two improved objectives, and Spaces D, F, and G have one.

In the case of *N*-objective problems, the number of possible spaces from the current solution is 2^N . The number of improved objectives varies from zero to *N* in this case. We generalize the replacement rules A and B by considering the number of improved objectives *d*. That is, the current solution **x** is replaced with a solution which has *d* or more improved objective values. We have the following generalized rule:

Rule *d*: Move to *d*-Improved Solutions:
Replace the current solution **x** with a solution which has *d* or more improved objectives.

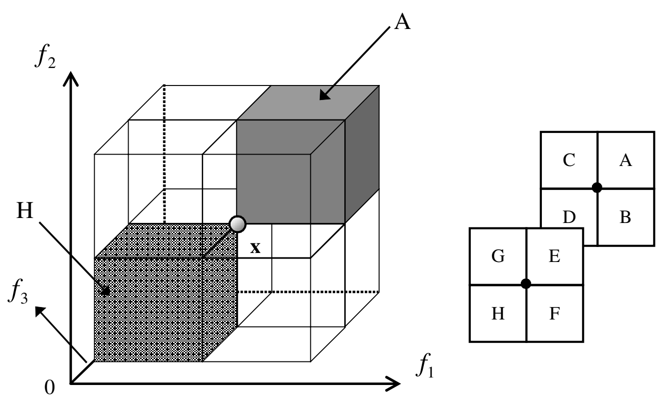

Fig. 5. Eight spaces for the current solution in the case of three-objective problems.

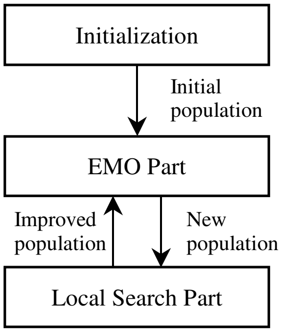

Fig. 6. Generic form of our local search part and EMO part.

By varying the value of d, we have the following rules where N is the number of objectives:

Rule N : Accept a solution which has N better objective values.
Rule $N-1$: Accept a solution which has $N-1$ or more better objective values.
⋮
Rule 2: Accept a solution which has at least two better objective values.
Rule 1: Accept a solution which has at least one better objective value.

Therefore, Rules A and B in Subsection 2.1 are Rule N and Rule 1 of the proposed rule, respectively.

3 Local Search Using the Generalized Replacement Rule

The outline of our local search can be written in a generic form as Fig. 6. This figure shows a basic structure of simple memetic algorithms. As shown in Fig. 6, our local search part can be applied to every EMO algorithm. For other types of memetic algorithms, see Krasnogor [26] where taxonomy of memetic algorithms was given using an index number D. This type of memetic algorithms is a $D=4$ memetic algorithm in his taxonomy (for details, see [26]).

We design our local search as follows:

[Proposed Local Search]

Iterate the following seven steps N_{pop} times, where N_{pop} is the number of populations to be governed by genetic operations such as crossover and mutation in an EMO algorithm. Then replace the current population with N_{pop} solutions obtained by the following steps.

Step 1: Randomly choose two individuals from the current population.
Step 2: Count the number of better objective values between the two solutions. Select a solution which has a larger number of better objective values.

Step 3: Select another solution from the current population, and back to Step 2 until t solutions are compared from the current population.

Step 4: Apply local search with the local search probability p_{LS} . If it is applied, go to Step 5. If not, go to Step 7.

Step 5: Generate a neighborhood solution of the current solution, and calculate the objectives of the generated solution. Count the number of improved objectives by the generated solution.

Step 6: If the number of improved objective values is d or more, replace the current solution with the generated solution, and back to Step 5 for examining the neighborhood solution for the generated solution. If not, back to Step 5 until the number of examined solution for the current solution becomes k. If there is no better solution among k neighborhood solutions, go to Step 7.

Step 7: Back to Step 1 until N_{pop} solutions are selected for the local search.

When local search is applied to the selected solution in Step 4, the final solution obtained by the local search is included in the next population. If local search is not applied, the selected solution is included in the next population. Therefore Steps 1 - 3 can be considered as the tournament selection for selecting candidate solutions for local search. In this tournament selection, we also employ the idea of the generalized replacement rule. That is, we select a solution with respect to the number of better objective values among t solutions. We use the local search probability p_{LS} for decreasing the number of solutions to which local search is applied. In this way, local search is not applied to all the selected solutions. If local search is applied to all the solution among the population, the computation time may be wasted to improve dominated solutions. Moreover, we also employ the number of examined solutions k in Step 6 in order to control the balance between genetic search and local search.

Since this proposed local search can be applied to any EMO algorithm, we apply our local search to SPEA [3] and NSGA-II [6]. In order to show its effectiveness, we employ multiobjective knapsack problems [3]. We show results of computer simulations in the next section.

4 Computer Simulations on Multiobjective Knapsack Problems

4.1 Multiobjective Knapsack Problems

We employ multiobjective knapsack problems [3] to which we applied EMO algorithms with the proposed local search. Those test problems are available from the web site (http://www.tik.ee.ethz.ch/~zitzler/). Generally, a 0/1 knapsack problem consists of a set of items, weight and profit associated with each item, and an upper bound for the capacity of the knapsack. The task is to find a subset of items which maximizes the total profits within the prespecified total weight of the items. This single-objective problem was extended to the multiobjective case by allowing an arbitrary number of knapsacks in [3]. In the multiobjective knapsack problem, there are m items and N knapsacks. Profits of items, weights of items, and capacities of knapsacks are denoted as follows:

p_{ij} : profit of item j according to knapsack i, (5)
w_{ij} : weight of item j according to knapsack i, (6)
c_i : capacity of knapsack i, (7)
where $j = 1,...,m$ and $i = 1,...,N$. (8)

The decision vector **x** of this problem is $\mathbf{x} = (x_1, x_2, ..., x_m)$, where x_j is a binary decision value. $x_j = 0$ and $x_j = 1$ mean that the item j is not included in the knapsacks, and it is included in the knapsacks, respectively. Thus N-objective problem can be written as follows:

$$\text{Maximize } \mathbf{z} = (f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ ...,\ f_N(\mathbf{x})) , \tag{9}$$

$$\text{subject to } \mathbf{x} \in \{0,1\}^m \text{ , and } \sum_{j=1}^{m} w_{ij} \cdot x_j \le c_i \text{ for } i = 1,...,N , \tag{10}$$

where each objective function is described in the following form:

$$f_i(\mathbf{x}) = \sum_{j=1}^{m} p_{ij} \cdot x_j \text{ for } i = 1,...,N . \tag{11}$$

On the website of the first author of [3], problems with 100, 250, 500, and 750 items for 2, 3 and 4 knapsacks are available. We employed the SPEA [3] and the NSGA-II [6] as representative EMO algorithms. These algorithms are known as high performance algorithms for multiobjective problems. We preliminarily defined the parameters in both the algorithms as shown in Table 1. We employed a one point crossover with the crossover rate 0.8, a bit-flip mutation with the mutation rate 0.01 to each bit. In our local search, we specified the tournament size as $t = 6$, the selection probability as $p_{\mathrm{LS}} = 0.1$, and the number of examined solution as $k = 3$. We commonly used these parameters to the SPEA and the NSGA-II. We generated 30 sets of different initial solutions and applied each of EMO algorithms to them.

After that we apply the three- and four-objective knapsack problems with 250, 500, and 750 items to show the effectiveness of generalizing the replacement rule based on the dominance relation.

4.2 Experimental Results on a Two-Objective Problem

We show the effectiveness of our local search on two-objective knapsack problems. We employed a 750-item problem for two knapsacks to show non-dominated solutions obtained by EMO algorithms with/without our local search. Therefore we applied each of four EMO algorithms to the problem. In this case, we can not show the effectiveness of generalization using the number of improved objective values d because the value of d can vary one or two for two-objective problems. We can see, however, the performance of the basic architecture of our local search with $d = 2$.

First we obtained 30 sets of non-dominated solutions by each algorithm. In order to depict figures clearly, we employed the 50%-attainment surface [27]. An attainment surface is a kind of trade-off surface obtained by a single run of an EMO algorithm. And 50%-attainment surface shows the estimated attainment surface obtained by at least 50% of multiple runs. Figs. 7 and 8 show the 50%-attainment surface of 30 sets of non-dominated solutions obtained by four EMO algorithms. Each axis of the figures shows the total profit of each knapsack. In a two objective problem, the total profit of each of two knapsacks is maximized.

From Figs. 7 and 8, we can see that the 50%-attaiment surface is improved by introducing the local search in the area of compromised solutions. Considering only a single objective, each of the original EMO algorithm (i.e., without the local search) found better solutions. Since we employed the local search with $d = 2$, the local search allows a solution to move to only a dominating solution in two objectives. Therefore the attainment surface in the area of compromised solutions is improved by the local search.

As for the local search using the generalized replacement rule with $d = 1$, we could not obtain better attainment surfaces than the original EMO algorithms. As explained in [23], a drawback of the replacement rule which allows a solution to move to non-dominated solutions is that the current solution can be degraded by multiple moves. That may be a reason why the local search with $d = 1$ was not effective to this problem.

Table 1. Parameter settings for EMO algorithms.

Problem (objectives, items)	# of evaluations	Population size	Secondary Population Size in SPEA
(2, 750)	125 000	250	100
(3, 250)	100 000	200	80
(3, 500)	125 000	250	100
(3, 750)	150 000	300	120
(4, 250)	125 000	250	100
(4, 500)	150 000	300	120
(4, 750)	175 000	350	140

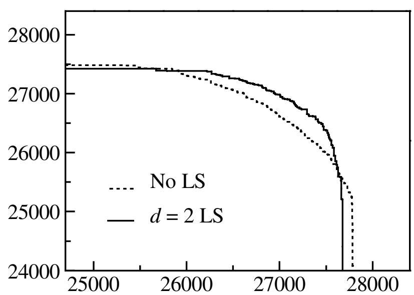

Fig. 7. 50%-attainment surface obtained by SPEA with/without local search.

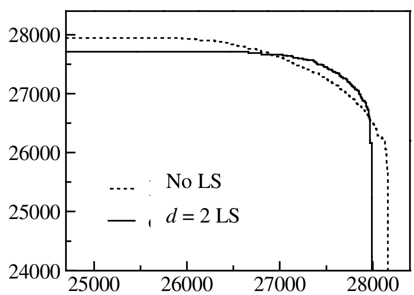

Fig. 8. 50%-attainment surface obtained by NSGA-II with/without local search.

4.3 Experimental Results on Three- and Four-Objective Problems

In the case of two-objective problems, we can depict the set distribution by two-dimensional graphs, it is difficult, however, to show the set distribution by figures for more than three-objective problems. In this paper, we use the coverage metric [3] to evaluate two sets of non-dominated solutions obtained by two EMO algorithms. Let $\mathbf{X'}, \mathbf{X''} \in \mathbf{X}$ be two sets of non-dominated solutions. The coverage metric can be defined as follows:

$$C(\mathbf{X'},\mathbf{X''}) = |\{\mathbf{a''} \in \mathbf{X''} | \exists \mathbf{a'} \in \mathbf{X'} : \mathbf{a'} \succeq \mathbf{a''}\}| / |\mathbf{X''}|. \quad (12)$$

The value $C(\mathbf{X'},\mathbf{X''}) = 1$ means that all points in $\mathbf{X''}$ are dominated by or equal to points in **X'**. On the other hand, $C(\mathbf{X'},\mathbf{X''}) = 0$ represents that no solutions in $\mathbf{X''}$ are covered by the set **X'**. It is noted that both $C(\mathbf{X'},\mathbf{X''})$ and $C(\mathbf{X''},\mathbf{X'})$ have to be considered, since $C(\mathbf{X'},\mathbf{X''})$ is not necessarily equal to $C(\mathbf{X''},\mathbf{X'})$.

Table 2. SPEA for 3-objective problems (250, 500, and 750 items).

3 objectives	No LS	$d = 1$	$d = 2$	$d = 3$
No LS	--	0.0311	0.0088	0.0179
$d = 1$	0.7444	--	0.1960	0.2123
$d = 2$	0.8437	0.5142	--	0.3671
$d = 3$	0.8044	0.4484	0.2796	--

Table 3. NSGA-II for 3-objective problems (250, 500, and 750 items).

3 objectives	No LS	$d = 1$	$d = 2$	$d = 3$
No LS	--	0.0961	0.0862	0.1147
$d = 1$	0.4373	--	0.2274	0.2369
$d = 2$	0.4797	0.3283	--	0.3127
$d = 3$	0.4466	0.3326	0.2473	--

Table 4. SPEA for 4-objective problems (250, 500, and 750 items).

4 objectives	No LS	$d = 1$	$d = 2$	$d = 3$	$d = 4$
No LS	--	0.0038	0.0010	0.0001	0.0012
$d = 1$	0.8820	--	0.1488	0.1369	0.1697
$d = 2$	0.9353	0.4231	--	0.2394	0.2904
$d = 3$	0.9299	0.4356	0.2792	--	0.2976
$d = 4$	0.9161	0.3756	0.2122	0.2011	--

We applied the EMO algorithms with/without our local search to three three-objective problems and three four-objective problems using the parameters in Table 1. We varied the number of improved objective values d as $d = 1,2,3$ in three-objective problems, and $d = 1,2,3,4$ in four-objective problems. Therefore we had four variants for each EMO algorithm in the case of three-objective problems, and five in the case of four-objective problems. We compare two sets of non-dominated solutions using the coverage metric, and calculate an average value over 30 trials. Tables 2–5 show the summarization of the results for each EMO algorithm. We average the values of the coverage over three different items. The second column of Tables 2 and 3 shows that the sets obtained original algorithm is covered by those obtained by algorithms with the proposed local search. For example, 0.8437 in the cell of the 4th row and the second column in Table 2 shows that 84.37 % solutions obtained by the original EMO algorithm (SPEA with No LS) are covered by solutions obtained by SPEA with the local search ($d = 2$). From Tables 2–5, we can see that the better sets obtained by our local search with $d = 2$ for three-objective problems, and $d = 2,3$ for four-objective problems.

Table 5. NSGA-II for 4-objective problems (250, 500, and 750 items).

4 objectives	No LS	$d = 1$	$d = 2$	$d = 3$	$d = 4$
No LS	--	0.0431	0.0114	0.0114	0.0155
$d = 1$	0.4701	--	0.1165	0.1233	0.1112
$d = 2$	0.5898	0.3736	--	0.2418	0.2475
$d = 3$	0.5719	0.3551	0.2214	--	0.2122
$d = 4$	0.6022	0.3643	0.2447	0.2325	--

Due to the page limitation, we don't show detail results on each problem. Further information is shown in the first authors web (http://www.res.kutc.kansai-u.ac.jp/~murata/research.html). Through the experiments, we found that the proposed local search was effective in the case of larger problems. That is, it was more effective in 500-item problem than 250, and 750 than 500 for each of three- and four-objective problems.

5 Conclusion and Future Works

In this paper, we generalized the replacement rules based on the dominance relation for local search in multiobjective optimization. Simulation results on knapsack problems with three- and four-objectives showed the effectiveness of the generalized replacement rule by introducing the number of improved objectives. As shown in the experimental results for the two-objective problems, the proposed local search is weak to improve each objective value. We can improve such weakness of the local search with the generalization rule. Since we employed the proposed only to knapsack problems in this paper, we can also apply other types of problems such as permutation problems, and function approximation problems.

References

[1] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *Proc. of 1st Int'l Conf. on Genetic Algorithms and Their Applications*, pp. 93–100, 1985.

[2] J. D. Knowles and D. W. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization," *Proc. of 1999 Congress on Evolutionary Computation*, pp. 98–105, 1999.

[3] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. on Evolutionary Computation*, **3** – Algorithms: Empirical Results," *Evolutionary Computation*, **8** (2), pp. 173–195, 2000.

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

[7] P. Merz and B. Freisleben, "Genetic local search for the TSP: New results," *Proc. of 4th IEEE Int'l Conf. on Evolutionary Computation*, pp. 159–164, 1997.

[8] N. Krasnogor and J. Smith, "A memetic algorithm with self-adaptive local search: TSP as a case study," *Proc. of 2000 Genetic and Evolutionary Computation Conf.*, pp. 987–994, 2000.

[9] P. Moscato, "Memetic algorithms: A short introduction," in D. Corne, F. Glover, and M. Dorigo (eds.), *New Ideas in Optimization*, McGraw-Hill, pp. 219-234, Maidenhead, 1999.

[10] W. E. Hart, N. Krasnogor, and J. Smith (eds.), *First Workshop on Memetic Algorithms* (WOMA I)*,* in *Proc. of 2000 Genetic and Evolutionary Computation Conf. Workshop Program,* pp. 95–130, 2000.
[11] W. E. Hart, N. Krasnogor, and J. Smith (eds.), *Second Workshop on Memetic Algorithms* (WOMA II)*,* in *Proc. of 2001 Genetic and Evolutionary Computation Conf. Workshop Program,* pp. 137–179, 2001.
[12] W. E. Hart, N. Krasnogor, and J. Smith (eds.), *Proc. of Third Workshop on Memetic Algorithms* (WOMA III)*,* 2002.
[13] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm," *Proc. of 3rd IEEE Int'l Conf. on Evolutionary Computation*, pp. 119–124, 1996.
[14] H. Ishibuchi and T. Murata, "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), pp. 392–403, 1998.
[15] A. Jaszkiewicz, "Genetic local search for multi-objective combinatorial optimization," *European Journal of Operational Research*, **137** (1), pp. 50–71, 2002.
[16] J. D. Knowles and D. W. Corne, "M-PAES: A memetic algorithm for multiobjective optimization," *Proc. of 2000 Congress on Evolutionary Computation*, pp. 325–332, 2000.
[17] J. D. Knowles and D. W. Corne, "A comparison of diverse approaches to memetic multiobjective combinatorial optimization," *Proc. of 2000 Genetic and Evolutionary Computation Conf. Workshop Program,* pp. 103–108, 2000.
[18] T. Murata, H. Nozawa, Y. Tsujimura, M. Gen, and H. Ishibuchi, "Effect of local search on the performance of cellular multi-objective genetic algorithms for designing fuzzy rule-based classification systems," *Proc. of 2002 Congress on Evolutionary Computation*, pp. 663–668, 2002.
[19] H. Ishibuchi, T. Yoshida, and T. Murata, "Selection of initial solutions for local search in multi-objective genetic local search," *Proc. of 2002 Congress on Evolutionary Computation*, pp. 950–955, 2002.
[20] K. Deb and T. Goel, "A hybrid multi-objective evolutionary approach to engineering shape design," *Proc. of 1st Int'l Conf. on Evolutionary Multi-Criterion Optimization*, pp. 385–399, 2001.
[21] E. Talbi, M. Rahoual, M. H. Mabed, and C. Dhaenens, "A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop," *Proc. of 1st Int'l Conf. on Evolutionary Multi-Criterion Optimization*, pp. 416–428, 2001.
[22] T. Murata, H. Nozawa, H. Ishibuchi, and M. Gen, "Modification of local search directions for non-dominated solutions in cellular multiobjective genetic algorithms for pattern classification problems," *Proc. of 2nd Int'l Conf. on Evolutionary Multi-Criterion Optimization*, 2003 (to appear).
[23] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. on Evolutionary Computation* (to appear).
[24] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?" *Proc. of 2001 Congress on Evolutionary Computation*, pp. 957–962, 2001.
[25] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, **10** (3), pp. 263–282, 2002.
[26] N. Krasnogor, "Studies on the theory and design space of memetic algorithms," *Ph. D. Thesis*, University of the West of England, Bristol, June 2002.
[27] C. M. Fonseca and P. J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," *Proc. of Parallel Problem Solving from Nature IV*, pp. 584–593, 1996.