Balance between Genetic Search and Local Search in Hybrid Evolutionary Multi-Criterion Optimization Algorithms

Hisao Ishibuchi
Dept. of Industrial Engineering
Osaka Prefecture University
Sakai, Osaka 599-8531, JAPAN
hisaoi@ie.osakafu-u.ac.jp
Phone: +81-72-254-9350

Tadashi Yoshida
Dept. of Industrial Engineering
Osaka Prefecture University
Sakai, Osaka 599-8531, JAPAN
yossy@ie.osakafu-u.ac.jp
Phone: +81-72-254-9351

Tadahiko Murata
Dept. of Informatics
Kansai University
Takatsuki, Osaka 569-1095, JAPAN
murata@res.kutc.kansai-u.ac.jp
Phone: +81-726-90-2429

Abstract

The aim of this paper is to clearly demonstrate the importance of finding a good balance between genetic search and local search in the implementation of hybrid evolutionary multi-criterion optimization (EMO) algorithms. We first modify the local search part of an existing multi-objective genetic local search (MOGLS) algorithm. In the modified MOGLS algorithm, the computation time spent by local search can be decreased by two tricks: to apply local search to only selected solutions (not all solutions) and to terminate local search before all neighbors of the current solution are examined. Next we show that the local search part of the modified MOGLS algorithm can be combined with other EMO algorithms. We implement a hybrid version of a strength Pareto evolutionary algorithm (SPEA). Using the modified MOGLS algorithm and the hybrid SPEA algorithm, we examine the balance between genetic search and local search through computer simulations on a two-objective flowshop scheduling problem. Computer simulations are performed using various specifications of parameter values that control the computation time spent by local search.

1. INTRODUCTION

One promising trick for improving the search ability of evolutionary multi-criterion optimization (EMO) algorithms is the hybridization with local search. Such a hybrid EMO algorithm was first implemented as a multi-objective genetic local search (MOGLS) algorithm in Ishibuchi & Murata (1996) together with a simple idea of elitism. The MOGLS algorithm was successfully applied to multi-objective flowshop scheduling problems in Ishibuchi & Murata (1998). While their MOGLS algorithm implemented two promising tricks for improving the search ability of EMO algorithms (i.e., hybridization and elitism), its search ability is not high if compared with recently proposed EMO algorithms such as a strength Pareto evolutionary algorithm (SPEA) of Zitzler & Thiele (1999) and a revised non-dominated sorting genetic algorithm (NSGA-II) of Deb et al. (2000).

Jaszkiewicz (1998) and Jaszkiewicz et al. (2001) improved the performance of the MOGLS algorithm by modifying its selection mechanism of parent solutions. While his MOGLS algorithm uses a scalar fitness function with random weight values for selection and local search as in the original MOGLS algorithm in Ishibuchi & Murata (1996), it does not use the roulette wheel selection. A pair of parent solutions is randomly selected from a pre-specified number of the best solutions (i.e., a kind of subpopulation) with respect to the scalar fitness function with the current weight values. The weight values are randomly updated whenever a pair of parent solutions is selected as in the original MOGLS algorithm. In the above-mentioned two MOGLS algorithms, local search is applied to all solutions generated by genetic operations in every generation. In some hybrid EMO algorithms, local search is used only when the execution of EMO algorithms is terminated. Deb & Goel (2001) applied local search to final solutions obtained by EMO algorithms for decreasing the number of non-dominated solutions (i.e., for decreasing the variety of final solutions). On the other hand, Talbi (2001) intended to increase the variety of final solutions by the application of local search.

The performance of the original MOGLS algorithm in Ishibuchi & Murata (1996) can be improved by carefully addressing the following issues:

Choice of initial solutions for local search: Local search was applied to all solutions in the current population in the original MOGLS algorithm. Its performance can be improved by choosing only good solutions from the current population as initial solutions for local search.

Specification of local search directions: The local search direction for each solution was specified by the scalar fitness function used in the selection of its parent solutions in the original MOGLS algorithm. Its performance can be improved by specifying an appropriate local search direction for each solution independent of the scalar fitness function used in the selection of its parent solutions.

Balance between genetic search and local search: If we simply combine local search with EMO algorithms, almost all the available computation time is spent by local search. This is because a large number of solutions are usually examined by local search for a single initial solution until a locally optimal solution is found. As a result, the global search ability of EMO algorithms is deteriorated by the hybridization with local search. In the original MOGLS algorithm, the balance between genetic search and local search was controlled by the number of neighbors examined by local search around the current solution. Local search was terminated if a better solution was not found among a pre-specified number of neighbors examined around the current solution. The balance can be also controlled by the number of solutions in the current population to which local search is applied. The performance of the original MOGLS algorithm can be improved by finding a good balance between genetic search and local search.

In this paper, first we briefly discuss the first two issues: choice of initial solutions for local search and specification of a local search direction for each initial solution. Then the balance between genetic search and local search is discussed through computer simulations on a two-objective flowshop scheduling problem.

2. MULTI-CRITERION OPTIMIZATION

Let us consider the following n-objective minimization problem:

$$\text{Minimize } \boldsymbol{z} = (f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x}), ...,\ f_n(\boldsymbol{x})), \quad (1)$$

$$\text{subject to } \boldsymbol{x} \in \boldsymbol{X}, \quad (2)$$

where $\boldsymbol{z}$ is the objective vector, $\boldsymbol{x}$ is the decision vector, and $\boldsymbol{X}$ is the feasible region in the decision space. Usually, there is no optimal solution $\boldsymbol{x}^*$ that satisfies the following inequality condition:

$$f_i(\boldsymbol{x}^*) \leq f_i(\boldsymbol{x}) \text{ for } \forall i \in \{1, 2, ..., n\} \text{ and } \forall \boldsymbol{x} \in \boldsymbol{X}. \quad (3)$$

Thus the task of EMO algorithms is not to find a single final solution but to find all solutions that are not dominated by any other solutions. Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be two decision vectors ($\boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{X}$). Then $\boldsymbol{b}$ is said to be dominated by $\boldsymbol{a}$ (i.e., $\boldsymbol{a} \prec \boldsymbol{b}$) if and only if the following two conditions hold:

$$f_i(\boldsymbol{a}) \leq f_i(\boldsymbol{b}) \text{ for } \forall i \in \{1, 2, ..., n\}, \quad (4)$$

$$f_i(\boldsymbol{a}) < f_i(\boldsymbol{b}) \text{ for } \exists i \in \{1, 2, ..., n\}. \quad (5)$$

When $\boldsymbol{b}$ is not dominated by any other solutions in $\boldsymbol{X}$, $\boldsymbol{b}$ is said to be a Pareto-optimal solution. That is, $\boldsymbol{b}$ is a Pareto-optimal solution when there is no solution $\boldsymbol{a}$ in $\boldsymbol{X}$ that satisfies the above two conditions.

While the task of EMO algorithms is to find all Pareto-optimal solutions, it is impractical to try to find true Pareto-optimal solutions of large problems. Thus EMO algorithms usually present non-dominated solutions among examined ones to decision makers as a result of their execution. In this case, the task of EMO algorithms is to drive populations to true Pareto-optimal solutions as close as possible.

3. HYBRID EMO ALGORITHMS

3.1 MOGLS ALGORITHM

In the original MOGLS algorithm, local search is applied to all solutions in every generation. The following scalar fitness function was used for both the selection of a pair of parent solutions and the local search for their offspring.

$$f(\boldsymbol{x}) = w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}) + \cdots + w_n f_n(\boldsymbol{x}), \quad (6)$$

where w_i is a non-negative weight. The point is to randomly specify the weight values whenever a pair of parent solutions is selected. This weight specification mechanism generates various search directions in the n-dimensional objective space. The MOGLS algorithm also uses a kind of elitism where all non-dominated solutions obtained during its execution are stored as a secondary population separately from the current population. A few non-dominated solutions are randomly selected from the secondary population and their copies are added to the current population.

The main characteristic feature of local search in the MOGLS algorithm is that all neighbors of the current solution are not examined. For decreasing the computation time spent by local search, only k neighbors of the current solution are randomly chosen and examined. If no better solution is found among the examined k neighbors, local search for the current solution is terminated. The first move strategy is used in local search. That is, the current solution is replaced as soon as a better neighbor is found.

Figure 1 shows the general outline of hybrid EMO algorithms discussed in this paper. In hybrid EMO algorithms, a new population is generated by genetic operations in the EMO algorithm part. Then the new population is improved by local search. The improved population is handled as the current population in the

EMO algorithm part. In this manner, the population update is iterated by genetic operations and local search until a pre-specified stopping condition is satisfied.

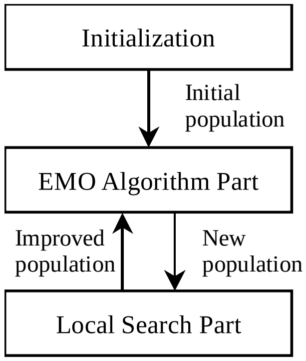

Figure 1: Outline of hybrid EMO algorithms.

3.2 MODIFICATION OF LOCAL SEARCH PART

In the original MOGLS algorithm, local search was applied to all solutions in the current population. The drawback of this scheme is computational inefficiency. That is, the application of local search to poor solutions seems to be mere waste of computation time. The efficiency of the original MOGLS algorithm can be improved by applying local search to only good solutions in the current population. The local search direction for each solution was specified by the scalar fitness function used in the selection of its parents. The drawback of this scheme is that the local search direction for each solution is not always appropriate.

As a remedy for these two drawbacks, we modify the local search part of the original MOGLS algorithm as follows:

[Modified Local Search Part]

Step 1. Iterate the following two procedures for constructing a local search pool of N_{pop} solutions:

(a) Randomly specify the weight values $w_1, ..., w_n$.

(b) Select a solution to be included in the local search pool from the current population (i.e., new population generated by genetic operations in Fig. 1) using the size four tournament selection with replacement based on the scalar fitness function with the current weight values specified in (a). That is, four solutions are randomly selected from the current population and a copy of the best one is added to the local search pool. The four solutions are returned to the current population for further selection to construct the local search pool.

Step 2. Randomly select N_{LS} solutions from the local search pool without replacement. Local search is applied to only the selected N_{LS} solutions. The local search direction of each solution is specified by the weight values used in the selection of that solution for constructing the local search pool. The next population consists of the improved N_{LS} solutions and the other $(N_{\mathrm{pop}} - N_{\mathrm{LS}})$ solutions in the local search pool.

3.3 HYBRIDIZATION WITH EMO ALGORITHMS

In the modified local search part, the local search direction of each solution is not inherited from its parent solutions. The local search direction is specified in the local search part independent of the EMO algorithm part in Fig. 1. Thus the modified local search part can be combined with any EMO algorithms even if they do not use the scalar fitness function in (6) for the selection of parent solutions. As shown in Fig. 1, the local search part of hybrid EMO algorithms receives a new population updated in the EMO algorithm part and returns an improved population by local search. It is an advantage of the modified MOGLS algorithm over the original one that the modified local search part can be combined as a module with any EMO algorithms.

In this paper, we examine the original MOGLS algorithm and its modified version. We also examine a hybrid version of the SPEA algorithm because high search ability of the SPEA algorithm to find Pareto-optimal solutions has been reported in the literature (see Zitzler & Thiele 1999 and Zitzler et al. 2000).

4. COMPUTER SIMULATIONS

4.1 EFFECT OF MODIFICATION OF MOGLS

We examined the effect of the modification of the local search part on the performance of the MOGLS algorithm. In the same manner as in Ishibuchi & Murata (1998), we generated a 40-job and 20-machine flowshop scheduling problem with two objectives: to minimize the makespan and to minimize the maximum tardiness. We applied the original MOGLS algorithm and its modified version to this test problem. Each algorithm was terminated when 60000 solutions were examined. As in Ishibuchi & Murata (1998), we used the position-based two-point crossover and the shift mutation as genetic operations. The neighborhood structure was defined by the shift mutation in local search.

We used the following parameter specifications. Population size: 20, crossover probability: 0.9, mutation probability for each string: 0.3, the number of elite solutions: 3, the number of neighbors examined for improving the current solution in local search (i.e., k): 2. These specifications are almost the same as Ishibuchi & Murata (1998). In the modified MOGLS algorithm, the tournament size was specified as four for constructing the local search pool from the current population. The number of selected initial solutions for local search was specified as $N_{\mathrm{LS}} = 20$ (i.e., the same as the population size).

Each algorithm was applied to the test problem 150 times. Fig. 2 and Fig. 3 show all solutions obtained by the 150 runs of each algorithm. From the comparison between Fig. 2 and Fig. 3, we can see that the modified MOGLS algorithm in Fig. 3 outperformed the original one in Fig. 2. That is, the performance of the original MOGLS algorithm was improved by the modification of the local search part.

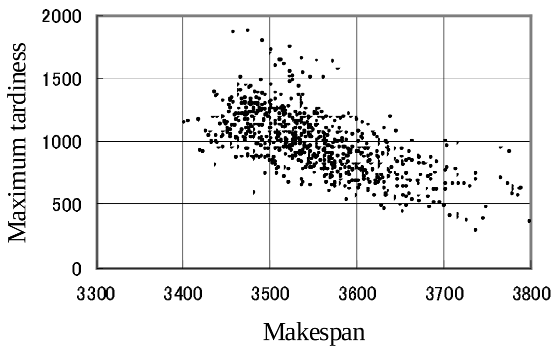

Figure 2: Obtained solutions by 150 runs of the original MOGLS algorithm.

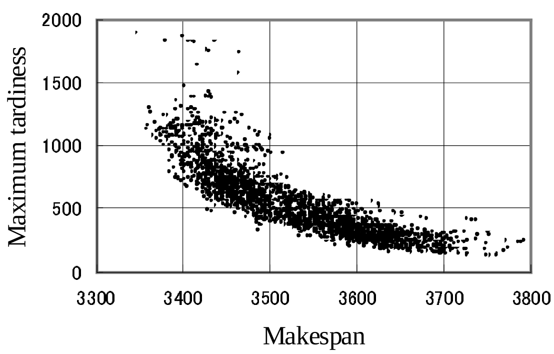

Figure 3: Obtained solutions by 150 runs of the modified MOGLS algorithm. Only the choice of initial solutions for local search is different from the original MOGLS algorithm.

4.2 EFFECT OF LOCAL SEARCH

In the same manner as in the previous subsection, we examined the effect of the hybridization with local search on the performance of EMO algorithms. In Fig. 4, we show simulation results by a simple EMO algorithm implemented by removing the local search part from the original MOGLS algorithm. Since the performance of this algorithm was very poor, many solutions are out of the range of Fig. 4. From the comparison of Fig. 4 with Fig. 2 and Fig. 3, we can see that the hybridization with local search significantly improved the performance of the simple EMO algorithm.

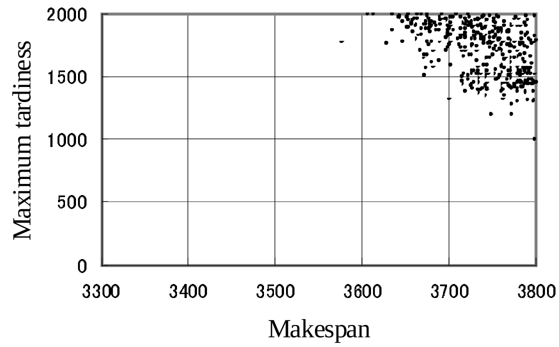

Figure 4: Obtained solutions by 150 runs of the original MOGLS algorithm with no local search. Many non-dominated solutions are out of the range of this figure.

While the comparison between Fig. 3 and Fig. 4 clearly shows that the simple EMO algorithm was significantly improved by the hybridization with local search, one may think that the improvement is mainly due to the poor performance of the simple EMO algorithm in Fig. 4. So we also implemented a hybrid version of the SPEA in the same manner as the modified MOGLS algorithm. Simulation results are summarized in Fig. 5 and Fig. 6. The maximum number of stored non-dominated solutions was specified as 20 in the computer simulations. The other parameters were specified in the same manner as in the previous subsection.

From the comparison between Fig. 5 and Fig. 6, we can see that the performance of the SPEA was slightly improved by the hybridization with local search. For example, more solutions were obtained in the region $[3300, 3400]\times[500, 1500]$ of Fig. 6 by the hybrid SPEA than the original SPEA in Fig. 5.

For further examining the effect of the hybridization with local search on the performance of the SPEA, a solution set obtained by the SPEA was compared with another solution set obtained by the hybrid SPEA. In this comparison, solutions obtained by one algorithm were examined whether they were dominated by other solutions obtained by the other algorithm. This comparison was performed over 150 runs of these two algorithms. Then the average number of non-dominated solutions was calculated. Simulation results are summarized in Table 1. This table shows the average number of obtained solutions by each algorithm, the average number of solutions that were not dominated by other solutions obtained by the other algorithm, the ratio of non-dominated solutions to obtained solutions, and the average CPU time. From this table, we can see that the hybrid SPEA outperformed the original SPEA in terms of the ratio of non-dominated solutions. We can also see from Table 1 that the CPU time was decreased by the hybridization with local search. This is because local search can be executed more efficiently than genetic search. If these two algorithms are compared under the same CPU time, it is more clearly shown that the hybrid SPEA outperforms the original SPEA (compare Fig. 7 with Fig. 5).

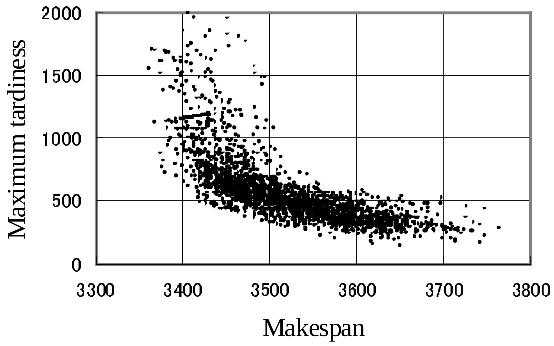

Figure 5: Obtained solutions by 150 runs of the original SPEA.

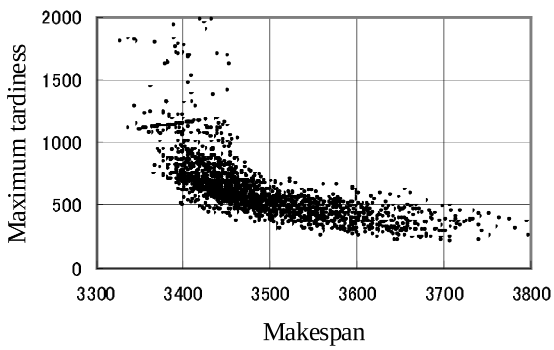

Figure 6: Obtained solutions by 150 runs of the hybrid SPEA.

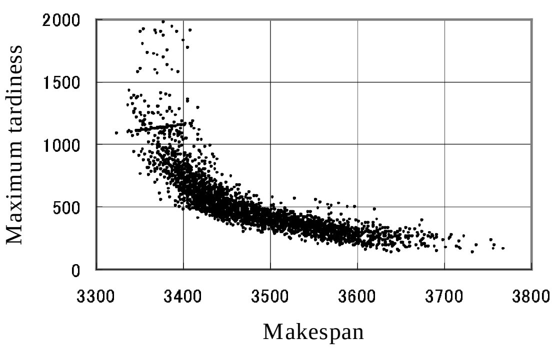

Figure 7: Obtained solutions by 150 runs of the hybrid SPEA with $k = 3$ using the same CPU time as the original SPEA.

Table 1: Comparison between SPEA and its hybrid version.

Algorithm	Obtained solutions	Non-dominated	Ratio of non-dominated	CPU time (Sec.)
SPEA	17.34	10.08	58.13%	17.47
Hybrid	14.69	9.85	67.05%	13.26

4.3 BALANCE BETWEEN GENETIC SEARCH AND LOCAL SEARCH IN THE MODIFIED MOGLS

The performance of hybrid EMO algorithms depends on parameter specifications. The point is to find a good balance between genetic search and local search. The balance is controlled by two parameters *k* and N_{LS} in the modified MOGLS algorithm (*k*: the number of neighbors examined for improving the current solution by local search, N_{LS}: the number of solutions in each population to which local search is applied). Table 2 shows simulation results with various values of *k*. In this table, N_{LS} was specified as $N_{LS} = 20$. Good results were not obtained from large values of *k* (see the column labeled as "Non-dominated"). Good specifications of *k* in Table 2 are $k = 1$~5.

Table 2: Simulation results by the modified MOGLS algorithm with various values of *k*. The value of N_{LS} was specified as N_{LS} = 20. Good results are highlighted by boldface letters. "Generation updates" means the number of generations.

k	Generation updates	Obtained solutions	Non-dominated	CPU time (seconds)
0	3000.00	17.78	1.08	19.81
1	1490.56	18.09	**4.29**	14.75
2	952.24	17.01	**4.60**	13.21
3	696.80	16.90	**3.41**	12.46
4	543.65	16.99	**3.44**	12.10
5	440.84	17.08	**3.13**	11.78
10	212.56	17.78	1.99	10.84
20	92.43	17.03	1.57	10.40
30	55.25	17.19	1.41	10.25
40	37.90	16.88	1.02	10.19
50	28.39	17.03	1.36	10.16
100	11.83	14.91	1.08	10.09
1521	1.00	5.28	1.36	13.75

On the other hand, Table 3 shows simulation results with various values of N_{LS}. In this table, *k* was specified as $k = 2$. Good results were not obtained from small values of N_{LS}. Good specifications of N_{LS} in Table 3 are $N_{LS} = 8$~20.

For further examining the balance between genetic search and local search, we examined various combinations of *k* and N_{LS}. A solution set obtained from each combination was compared with other solution sets obtained from other combinations in the same manner as in the previous computer simulations. Simulation results are summarized in Table 4. This table shows the number of solutions that were not dominated by any other solutions obtained from other parameter specifications. Table 4 shows average results over 150 trials as in the previous computer simulations. From this table, we can see that appropriate specifications of N_{LS} and k are related to each other. An appropriate value of N_{LS} decreases as the specified value of k increases in Table 4. In general, larger values of these two parameters mean longer computation time spent by local search. Thus the increase of one parameter value needs the decreases of the other parameter value for keeping a good balance between genetic search and local search.

Table 3: Simulation results by the modified MOGLS algorithm with various values of N_{LS}. The value of k was specified as $k = 2$. Good results are highlighted by boldface letters.

N_{LS}	Generation updates	Obtained solutions	Non-dominated	CPU time (seconds)
0	3000.00	17.35	0.77	18.90
2	2467.09	17.37	1.46	17.26
4	2094.59	17.77	2.12	16.26
6	1819.94	17.93	2.56	15.53
8	1611.89	16.93	**3.33**	14.92
10	1444.13	16.81	**3.11**	14.49
12	1306.81	17.70	2.71	14.14
14	1193.71	17.51	**3.27**	13.85
16	1100.39	17.27	2.93	13.59
18	1019.59	17.58	**3.09**	13.37
20	952.24	17.01	**3.62**	13.21

Table 4: The average number of solutions that were not dominated by any other solutions from other combinations of parameter values. The execution of the modified MOGLS algorithm was terminated when 60000 solutions were examined. Good results are highlighted by boldface letters.

N_{LS}	The value of k										
	0	1	2	3	4	5	10	20	30	40	50
0	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
2	0.17	0.06	0.33	0.43	0.45	**0.67**	**0.58**	0.35	**0.58**	**0.51**	0.47
4	0.17	0.39	0.45	0.41	**0.59**	**0.55**	**0.73**	0.24	0.40	0.31	0.37
6	0.17	0.35	**0.55**	**0.51**	**0.51**	**0.81**	**0.50**	0.29	0.32	0.29	0.23
8	0.17	0.48	**0.83**	**0.61**	**0.65**	0.37	**0.52**	0.27	0.11	0.25	0.17
10	0.17	**0.50**	**0.66**	0.39	**0.64**	**0.59**	0.31	0.21	0.21	0.20	0.19
12	0.17	0.49	0.47	**0.57**	**0.57**	**0.64**	0.30	0.17	0.16	0.13	0.13
14	0.17	**0.75**	**0.62**	0.42	0.42	0.46	0.31	0.25	0.17	0.15	0.09
16	0.17	**0.61**	**0.74**	**0.53**	**0.67**	0.29	0.21	0.12	0.11	0.06	0.13
18	0.17	**0.68**	**0.58**	**0.65**	0.41	0.42	0.35	0.15	0.09	0.04	0.04
20	0.17	**0.59**	**0.61**	0.35	0.45	0.37	0.32	0.15	0.09	0.09	0.07

We also performed the same computer simulation using different specifications of the stopping condition. In one specification, we decreased the available computation time from the examination of 60000 solutions to 20000 solutions. Simulation results are shown in Table 5. In the other specification, it was increased to 120000 solutions. Simulation results are summarized in Table 6. Table 5 and Table 6 show that appropriate values of N_{LS} and k are related to each other as in Table 4. From the comparison among the three tables, we can see that larger values of k can be used when the available computation resource is larger (i.e., Table 6). This means that we can use a larger portion of the computation time for local search when the available computation time is longer.

Table 5: The average number of solutions that were not dominated by any other solutions from other combinations of parameter values. The execution of the modified MOGLS algorithm was terminated when 20000 solutions were examined. Good results are highlighted by boldface letters.

N_{LS}	The value of k										
	0	1	2	3	4	5	10	20	30	40	50
0	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
2	0.02	0.04	0.37	0.39	**0.51**	**0.52**	0.44	**0.63**	**0.68**	0.49	**0.92**
4	0.02	0.11	0.23	0.47	0.48	**0.57**	0.49	0.37	0.39	0.43	0.39
6	0.02	0.35	0.45	0.49	0.43	**0.51**	0.39	0.33	0.30	0.35	0.23
8	0.02	0.18	**0.71**	**0.54**	0.35	0.23	0.24	0.25	0.22	0.18	0.29
10	0.02	0.43	0.43	0.43	0.29	0.41	0.24	0.20	0.16	0.17	0.24
12	0.02	0.40	**0.52**	0.30	0.45	0.41	0.17	0.11	0.15	0.11	0.18
14	0.02	0.41	**0.52**	0.48	0.27	0.28	0.11	0.06	0.16	0.17	0.05
16	0.02	0.32	**0.53**	0.41	0.32	0.26	0.20	0.05	0.05	0.07	0.09
18	0.02	0.35	0.33	**0.53**	0.31	0.23	0.12	0.04	0.04	0.05	0.02
20	0.02	0.49	**0.53**	0.32	0.40	0.29	0.11	0.07	0.01	0.01	0.07

Table 6: The average number of solutions that were not dominated by any other solutions from other combinations of parameter values. The execution of the modified MOGLS algorithm was terminated when 120000 solutions were examined. Good results are highlighted by boldface letters.

N_{LS}	The value of k										
	0	1	2	3	4	5	10	20	30	40	50
0	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
2	0.11	0.12	0.37	0.48	**0.65**	**0.57**	**0.52**	**0.52**	0.42	**0.61**	**0.51**
4	0.11	0.41	**0.61**	**0.52**	**0.61**	**0.57**	**0.53**	0.47	0.47	0.25	0.29
6	0.11	0.27	**0.70**	**0.55**	**0.54**	**0.73**	**0.51**	0.37	0.35	0.31	0.35
8	0.11	**0.51**	**0.54**	**0.65**	**0.59**	**0.55**	0.45	0.27	0.28	0.27	0.21
10	0.11	0.38	**0.51**	**0.51**	**0.71**	**0.77**	**0.54**	0.39	0.19	0.29	0.27
12	0.11	**0.53**	**0.85**	**0.77**	**0.51**	**0.73**	0.41	0.18	0.15	0.26	0.15
14	0.11	**0.71**	**0.53**	**0.69**	**0.56**	**0.55**	0.38	0.31	0.15	0.14	0.19
16	0.11	**0.56**	**0.62**	**0.69**	**0.73**	0.45	0.38	0.15	0.10	0.13	0.14
18	0.11	**0.61**	**0.53**	**0.72**	**0.58**	**0.53**	0.31	0.28	0.15	0.11	0.11
20	0.11	**0.69**	**0.57**	**0.70**	**0.74**	**0.60**	0.11	0.13	0.16	0.13	0.16

4.4 BALANCE BETWEEN GENETIC SEARCH AND LOCAL SEARCH IN THE HYBRID SPEA

We also examined the balance between genetic search and local search using the hybrid SPEA in the same manner as in the previous subsection. Table 7 shows simulation results by the hybrid SPEA with $N_{\mathrm{LS}} = 20$ and various values of k. Good results were not obtained from large values of k. Good specifications of k in Table 7 are $k = 2$~5. On the other hand, Table 8 shows simulation results by the hybrid SPEA with $k = 2$ and various values of N_{LS}. Good results were not obtained from small values of N_{LS}. Good specifications of N_{LS} in Table 8 are N_{LS} = 18~20. We can also see that the simulation results by the hybrid SPEA in Table 7 and Table 8 are similar to those by the modified MOGLS algorithm in Table 2 and Table 3, respectively.

Table 7: Simulation results by the hybrid SPEA with various values of k. The value of N_{LS} was specified as N_{LS} = 20. Good results are highlighted by boldface letters.

k	Generation updates	Obtained solutions	Non-Dominated	CPU time (seconds)
0	3000.00	15.46	1.26	19.03
1	1483.57	15.69	2.91	14.91
2	976.18	14.69	**3.12**	13.26
3	719.59	15.85	**3.73**	12.59
4	565.81	15.43	**3.50**	12.21
5	462.69	15.35	**3.95**	11.99
10	227.53	15.68	2.93	10.95
20	97.62	16.69	1.77	10.47
30	57.25	16.68	1.62	10.31
40	38.53	16.76	1.29	10.23
50	28.95	16.38	1.15	10.19
100	11.58	15.35	1.11	10.13
1521	2.00	5.10	1.35	13.89

Table 8: Simulation results by the hybrid SPEA with various values of N_{LS}. The value of k was specified as $k = 2$. Good results are highlighted by boldface letters.

N_{LS}	Generation Updates	Obtained solutions	Non-Dominated	CPU time (seconds)
0	3000.00	15.46	0.78	19.03
2	2486.16	15.64	1.99	17.50
4	2123.08	15.13	1.99	16.41
6	1850.80	16.26	2.33	15.74
8	1641.54	15.63	2.49	15.06
10	1474.84	16.01	2.56	14.63
12	1338.12	15.45	2.53	14.28
14	1223.87	15.80	2.62	14.06
16	1128.85	15.39	2.70	13.77
18	1046.31	15.75	**3.19**	13.54
20	976.18	15.44	**3.23**	13.26

We also examined various combinations of k and N_{LS} using the hybrid SPEA. Simulation results are summarized in Table 9 ~ Table 11. As in the previous subsection, these tables show simulation results using different stopping conditions. From these tables, we can see that appropriate specifications of N_{LS} and k are related to each other. An appropriate value of N_{LS} decreases as the specified value of k increases. From the comparison between the simulation results in this subsection by the hybrid SPEA and those in the previous subsection by the modified MOGLS algorithm, we can see that appropriate specifications of N_{LS} and k depend on the algorithm. For example, appropriate values of k for the hybrid SPEA are larger than those for the modified MOGLS algorithm. This is observed from the comparison between Table 6 and Table 11.

Table 9: The average number of solutions that were not dominated by any other solutions. The execution of the hybrid SPEA was terminated when 60000 solutions were examined. Good results are highlighted by boldface letters.

N_{LS}	The value of k										
	0	1	2	3	4	5	10	20	30	40	50
0	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
2	0.07	0.09	0.29	0.23	0.38	0.37	**0.76**	**1.03**	**0.57**	**0.76**	**0.61**
4	0.07	0.25	0.37	0.29	0.37	0.29	**0.63**	**0.55**	**0.53**	**0.50**	0.35
6	0.07	0.29	0.49	0.34	0.46	**0.62**	0.48	**0.50**	0.39	0.33	0.33
8	0.07	0.29	0.31	0.49	0.26	**0.72**	0.49	0.30	0.41	0.41	0.35
10	0.07	0.23	0.43	0.31	**0.59**	**0.56**	**0.57**	0.29	0.41	0.22	0.21
12	0.07	0.32	0.35	0.41	0.33	**0.66**	**0.58**	0.49	0.15	0.18	0.16
14	0.07	0.19	0.37	0.47	0.49	**0.57**	**0.57**	0.37	0.19	0.17	0.12
16	0.07	0.32	0.31	0.44	**0.65**	0.39	0.37	0.39	0.22	0.06	0.05
18	0.07	0.31	0.34	0.44	**0.65**	**0.51**	0.33	0.23	0.14	0.17	0.07
20	0.07	0.24	**0.67**	0.43	0.41	0.34	0.33	0.12	0.15	0.08	0.08

Table 10: The average number of solutions that were not dominated by any other solutions. The execution of the hybrid SPEA was terminated when 20000 solutions were examined. Good results are highlighted by boldface letters.

N_{LS}	The value of k										
	0	1	2	3	4	5	10	20	30	40	50
0	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
2	0.06	0.10	0.28	0.17	0.23	0.26	**0.61**	**0.68**	0.47	**0.69**	**0.72**
4	0.06	0.22	0.21	0.23	0.39	0.32	**0.73**	0.47	0.41	**0.53**	0.34
6	0.06	0.25	0.29	0.29	0.41	**0.56**	**0.58**	0.33	0.35	0.28	0.29
8	0.06	0.38	0.37	0.47	0.24	**0.65**	0.44	0.15	0.25	0.27	0.33
10	0.06	0.23	0.23	0.31	0.24	0.35	0.38	0.14	0.25	0.14	0.13
12	0.06	0.37	0.29	0.38	0.28	0.37	0.41	0.25	0.13	0.16	0.10
14	0.06	0.18	0.46	0.45	0.40	0.35	0.23	0.12	0.11	0.11	0.09
16	0.06	0.36	0.30	0.29	0.40	0.41	0.20	0.14	0.09	0.07	0.04
18	0.06	0.43	**0.51**	0.19	0.39	0.35	0.13	0.11	0.03	0.06	0.04
20	0.06	0.41	0.43	0.42	0.24	0.40	0.17	0.03	0.07	0.06	0.03

Table 11: The average number of solutions that were not dominated by any other solutions. The execution of the hybrid SPEA was terminated when 120000 solutions were examined. Good results are highlighted by boldface letters.

N_{LS}	The value of k										
	0	1	2	3	4	5	10	20	30	40	50
0	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
2	0.09	0.11	0.23	0.25	0.21	0.21	**0.63**	**0.62**	**0.65**	**0.77**	**0.56**
4	0.09	0.17	0.28	0.21	0.38	0.34	0.43	**0.61**	**0.65**	**0.53**	0.48
6	0.09	0.27	**0.57**	0.36	0.45	**0.57**	0.45	**0.57**	0.46	**0.57**	**0.52**
8	0.09	0.37	0.45	0.39	0.43	0.45	**0.57**	0.27	**0.51**	0.39	0.39
10	0.09	0.39	0.45	0.31	0.43	0.47	**0.54**	**0.51**	0.44	**0.51**	0.41
12	0.09	0.35	0.42	0.49	0.28	**0.54**	**0.53**	0.40	0.31	0.31	0.24
14	0.09	0.32	0.36	0.36	**0.56**	**0.53**	**0.69**	0.34	0.33	0.27	0.13
16	0.09	0.33	0.37	0.47	**0.67**	**0.56**	0.49	**0.55**	0.29	0.19	0.15
18	0.09	0.34	0.47	**0.58**	**0.58**	**0.54**	0.47	0.25	0.21	0.23	0.16
20	0.09	0.42	0.45	**0.59**	0.49	0.45	**0.50**	0.20	0.13	0.21	0.19

5. CONCLUSIONS

In this paper, we first modified the local search part of the MOGLS algorithm of Ishibuchi & Murata (1996) for applying local search only to good solutions in the current population and assigning an appropriate local search direction to each solution. The local search direction of each solution is specified in the modified local search part independent of genetic operations in the EMO algorithm part. Thus the modified local search part can be combined with other EMO algorithms. We implemented a hybrid SPEA by combining local search with the SPEA. Using the modified MOGLS algorithm and the hybrid SPEA, we examined the balance between genetic search and local search. Simulation results in this paper showed that the performance of the hybrid EMO algorithms strongly depends on this balance. When a good balance is achieved by appropriate parameter specifications, the hybrid EMO algorithms outperform the corresponding non-hybrid EMO algorithms.

It was also shown through computer simulations with different stopping conditions that appropriate parameter specifications for achieving a good balance between genetic search and local search depend on the amount of the available computation time. When long computation time was available, good results were obtained from parameter specifications that increase the ratio of the computation time spent by local search. On the other hand, good results were obtained in the case of a small ratio of the computation time spent by local search when we did not have long computation time. Simulation results also showed that different hybrid EMO algorithms require different parameter specifications for achieving a good balance. An appropriate ratio of the computation time spent by local search in the hybrid SPEA was larger than that in the modified MOGLS algorithm. Implication of this observation is not clear. One possible explanation is that genetic search in the hybrid SPEA may require shorter computation time than that in the modified MOGLS algorithm because the EMO algorithm part of the hybrid SPEA is more powerful than that of the modified MOGLS algorithm (compare Fig. 5 by the original non-hybrid SPEA with Fig. 4 by the EMO algorithm part in the modified MOGLS algorithm).

REFERENCES

K. Deb and T. Goel, "A hybrid multi-objective evolutionary approach to engineering shape design," *Proc. of 1st International Conference on Evolutionary Multi-Criterion Optimization*, pp. 385-399, Zurich, Switzerland, March 7-9, 2001.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast elitist multi-objective genetic algorithm: NSGA-II," *KanGAL Report* 200001, Indian Institute of Technology Kanpur, 2000.

H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm," *Proc. of 3rd IEEE International Conference on Evolutionary Computation*, pp. 119-124, 1996.

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*, vol. 28, no. 3, pp. 392-403, 1998.

A. Jaszkiewicz, "Genetic local search for multiple objective combinatorial optimization," *Working Paper*, RA-014/98, Poznan University of Technology, 1998.

A. Jaszkiewicz, M. Hapke, and P. Kominek, "Performance of multiple objective evolutionary algorithms on a distribution system design problem - Computational experiment," *Proc. of 1st International Conference on Evolutionary Multi-Criterion Optimization*, pp. 241-255, 2001.

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 International Conference on Evolutionary Multi-Criterion Optimization*, pp. 416-428, Zurich, Switzerland, March 7-9, 2001.

E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.

E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.