IMPLEMENTATION OF LOCAL SEARCH IN HYBRID MULTI-OBJECTIVE GENETIC ALGORITHMS: A CASE STUDY ON FLOWSHOP SCHEDULING

Hisao Ishibuchi and Tadashi Yoshida
Department of Industrial Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka, 599-8531, Japan

ABSTRACT

This paper examines the following issues related to the implementation of local search in hybrid multi-objective genetic algorithms: specification of an objective function to be optimized by local search, early termination of local search before finding a locally optimum solution, choice of individuals to which local search is applied, and timing of the application of local search. These issues are examined through computer simulations on a flowshop scheduling problem using a hybrid version of a well-known multi-objective genetic algorithm: the strength Pareto evolutionary algorithm (SPEA). Simulation results show that the hybridization with local search degrades the search ability of the SPEA when the implementation of local search is not appropriate. It is also shown that the hybridization has the possibility to improve the convergence speed of the SPEA to the Pareto front.

1. INTRODUCTION

Since Shaffer's work [1], genetic algorithms have been applied to multi-objective optimization problems in many studies [2-4]. One approach to the design of multi-objective genetic algorithms with high search ability is the efficient use of non-dominated solutions stored in a secondary population separately from the current population [5-8]. Another approach is the hybridization with local search [9-11]. Hybridization with local search, however, often degrades the global search ability of multi-objective genetic algorithms when the available computation time is limited. This is because almost all the available computation time is spent by local search. A large number of solutions are examined for finding a locally optimum solution by local search from each initial (i.e., starting) solution generated by genetic operations.

For decreasing the computation time spent by local search, we examined the following two tricks in our former studies [12]: Early termination of local search before finding locally optimum solutions and the restriction on the number of solutions to which local search is applied. Local search, which was applied to only a few solutions in the current population, was terminated before locally optimum solutions were found. We demonstrated the importance of finding a good balance between local search and genetic search in hybrid multi-objective genetic algorithms [12]. In addition to these two tricks, we also examine the timing of the application of local search in this paper. While local search was usually applied to solutions in every generation [9-12], its application is executed in every T generations in this paper where T is a user-definable parameter.

When hybrid algorithms are applied to a multi-objective optimization problem, we have to specify an objective function to be optimized by local search. This specification is straightforward in the application to a single-objective optimization problem because the single objective can be used for both genetic search and local search. On the other hand, the specification of an objective function for local search is not straightforward in the case of multi-objective optimization. A weighed sum of multiple objectives was often used for local search in hybrid multi-objective genetic algorithms [9-12]. In this paper, we examine the following six alternatives:

(a) The weighted sum of multiple objectives with random weights. Initial (i.e., starting) solutions for local search are randomly selected from the current population.
(b) The weighted sum of multiple objectives with random weights. An appropriate initial solution is selected from the current population for each weight vector.
(c) The weighted sum of multiple objectives with pseudo weights [4]. A pseudo-weight vector is specified for each of randomly selected initial solutions based on its

location in the objective space.

(d) The direct use of the fitness function in the strength Pareto evolutionary algorithm (SPEA [5]).

(e) Move to a non-dominated neighbor that is not dominated by the current solution.

(f) Move to a better neighbor that dominates the current solution.

In the last two alternatives, no objective function is explicitly defined. The objective function in (d), which is defined by the dominance relation in the primary and secondary populations, cannot be simply written.

In this paper, we first examine the above-mentioned six alternatives in the implementation of local search. Then we examine the balance between genetic search and local search using the three tricks: Early termination of local search, its application to only a limited number of solutions, and its application to solutions in every T generations. Simulation results on a flowshop scheduling problem clearly show that inappropriate hybridization of the SPEA with local search significantly degrades its global search ability to find a variety of non-dominated solutions. We also show that a hybrid algorithm with a good balance between local search and global search can outperform the non-hybrid SPEA.

2. OBJECTIVE FUNCTION FOR LOCAL SEARCH

For simplicity of explanation, let us consider the following two-objective optimization problem:

$$\text{Minimize} \quad f_1(\boldsymbol{x}) \text{ and } f_2(\mathbf{x}). \tag{1}$$

2.1 Weighted Sum with Random Weights

While we can use many multi-objective genetic algorithms in the literature to our two-objective optimization problem, their hybridization with local search is not straightforward because local search is an iterative improvement procedure for optimizing a single objective function. That is, we have to construct a scalar objective function to be optimized by local search.

A simple approach to the implementation of local search for our two-objective problem in (1) is the use of the following weighted sum as an objective function.

$$f(\boldsymbol{x}) = w_1 \cdot f_1(\boldsymbol{x}) + w_2 \cdot f_2(\boldsymbol{x}), \tag{2}$$

where w_1 and w_2 are non-negative weights satisfying the following relations:

$$w_1 + w_2 = 1, \; w_1 \geq 0, \; w_2 \geq 0. \tag{3}$$

The specification of the weight vector $\boldsymbol{w} = (w_1, w_2)$ corresponds to that of the local search direction in the two-dimensional objective space. For finding a variety of non-dominated solutions, a different weight vector was used for each solution in [9,10]. In Fig. 1, the weight vector for each solution is randomly specified.

2.2 Selection of Initial Solutions with Random Weights

As we can see in Fig. 1, a randomly specified local search direction for each solution is not always appropriate. Moreover the application of local search to poor solutions seems to be mere waste of CPU time. For decreasing the inefficiency of the random weight specification, local search was applied to only good solutions in our former study [12] as shown in Fig. 2. In [12], first a local search direction was randomly specified. Then an initial solution was selected from the current population using the tournament selection of the size four with replacement where the weighted sum in (2) with the current weight vector is used for evaluating each individual.

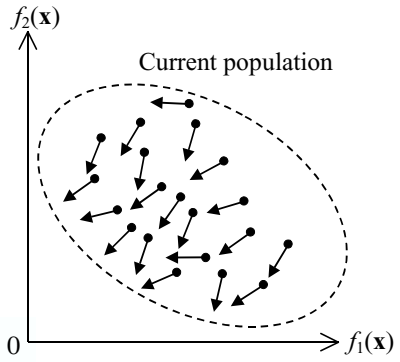

Fig. 1 Randomly specified local search directions.

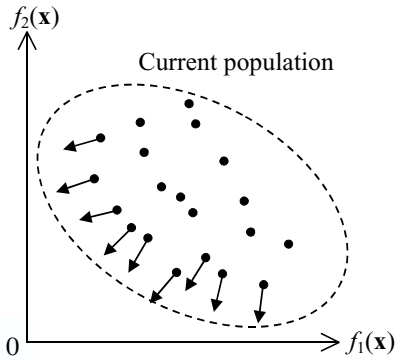

Fig. 2 Application of local search only to good solutions.

2.3 Use of Pseudo-Weights

In Deb [4], the pseudo-weight w_i for the i-th objective is defined for the current solution **x** as

$$w_i = \frac{f_i^{\max} - f_i(\mathbf{x})}{f_i^{\max} - f_i^{\min}} \Bigg/ \sum_{j=1}^{n} \frac{f_j^{\max} - f_j(\mathbf{x})}{f_j^{\max} - f_j^{\min}}, \qquad (4)$$

where $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum values of the i-th objective function in the current population, respectively. The weighted sum in (2) is used in local search where a pseudo-weight vector is specified using this formulation for each initial solution randomly selected from the current population.

2.4 Direct Use of a Fitness Function

Another approach to the implementation of local search is the direct use of fitness functions in multi-objective genetic algorithms. In this paper, we use the fitness function in the SPEA [5], which is defined by the dominance relation among solutions in the primary and secondary populations. A drawback of this approach is that longer CPU time is required for evaluating each neighbor in local search than the above-mentioned approaches with the weighted sum of multiple objectives.

2.5 Move to Non-dominated Solutions

For comparison, we examine other alternatives based on the dominance relation between the current solution and its neighbors. One implementation of local search is to replace the current solution with its neighbor that is not dominated by the current solution. This approach is illustrated in Fig. 3 where the current solution (closed circle) can be replaced with any neighbor (open circle) in the shaded region. A drawback of this approach is that the current solution may be degraded by multiple moves.

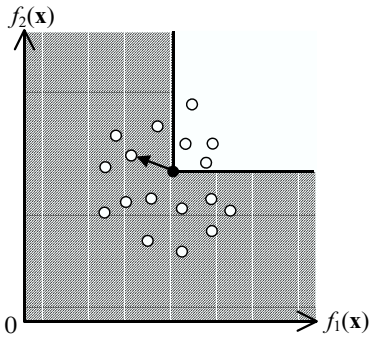

Fig. 3 Move to a non-dominated solution.

2.6 Move to Better Solutions

Another approach is to move to a better neighbor that dominates the current solution. A drawback of this approach is that the movable region from the current solution is very small especially in the case of many objectives (i.e., in a high-dimensional objective space).

3. LOCAL SEARCH AND GENETIC SEARCH

For finding a good balance between local search and genetic search, we examine the following three tricks that can control the CPU time spent by local search.

3.1 Early termination of local search

In local search of our former studies [9,10,12], a neighboring solution was randomly generated from the current solution. The current solution was replaced with the generated neighbor if the neighbor was better than the current solution. When no better solution was found among randomly generated k neighbors of the current solution, local search was terminated.

3.2 Restriction on the Number of Initial Solutions

In [12], an initial solution for local search was selected from the current population using the tournament selection based on a randomly specified weight vector. We use the same idea in the second alternative of the objective function for local search. In the other alternatives, we randomly choose a pre-specified number of initial solutions from the current population. The next population consists of the improved solutions by local search and the other solutions in the current population to which local search is not applied. The number of solutions to which local search is applied is denoted by N.

3.3 Timing of Local Search

In this paper, local search is applied to solutions in every T generations ($T = 1$ in our former studies [9, 10, 12]).

4. COMPUTER SIMULATIONS

4.1 Test Problem

As in our former studies [9,10,12], we generated a two-objective flowshop scheduling problem with 40 jobs and 20 machines. The two objectives are to minimize the makespan and the maximum tardiness. We used the shift mutation in local search for generating neighbors of the current solution. The total number of neighbors for each solution is 1521 (i.e., 39×39).

4.2 Parameter Specification

We implemented a hybrid algorithm of the SPEA with local search where parameter values were specified as

Size of the primary population: 100,
Size of the secondary population: 100,
Crossover rate: 0.9 (two-point order crossover),
Mutation rate: 0.3 (shift mutation).

We examined several specifications of the parameters in local search (i.e., k, N and T). The hybrid algorithm was terminated when 100,000 solutions were examined.

4.3 Simulation Results

First we compared the six alternatives for specifying the objective function to be optimized by local search. The parameters in local search were specified as $k = 5$, $N = 5$ and $T = 1$. Non-dominated solutions obtained from five runs for each alternative are depicted in Fig. 4 and Fig. 5. From these figures, we can see that slightly better results were obtained in Fig. 4 with the weighted sum in (2) than Fig. 5. The average CPU time was about 6 seconds (i.e., 5.21 ~ 6.88) for the five alternatives except for the direct use of the fitness function where much longer CPU time was required (i.e., 17.50 seconds).

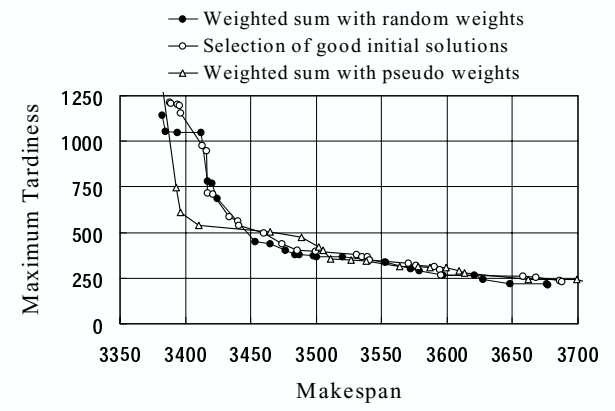

Fig. 4 Simulation results by the first three alternatives.

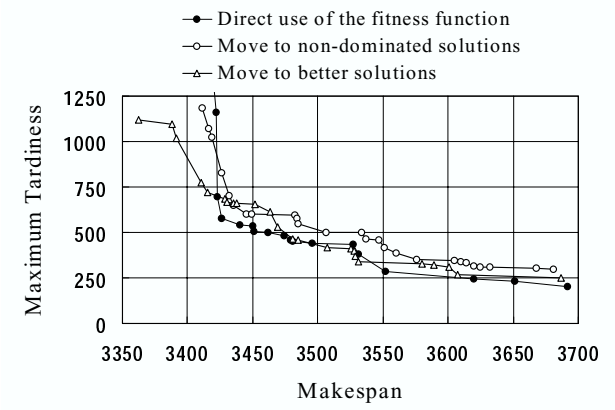

Fig. 5 Simulation results by the last three alternatives.

In the above computer simulations, early termination of local search was realized by the small value of k (i.e., $k = 5$). In Fig. 6, we show simulation results for several values of k (i.e., $k = 5,100,1000$). The second alternative of the objective function was used in the computer simulations. For comparison, we also show simulation results by the non-hybrid SPEA (i.e., the original SPEA) in Fig. 6. From this figure, we can see that much worse results were obtained from large values of k than the original SPEA. This means that the search ability of the SPEA was deteriorated by the hybridization with local search when we did not use the early termination trick.

In Fig. 7, we show simulation results for some combinations of k and N (N: the number of solutions to which local search is applied in each generation). Good results could not be obtained when both k and N were large (also see [12]).

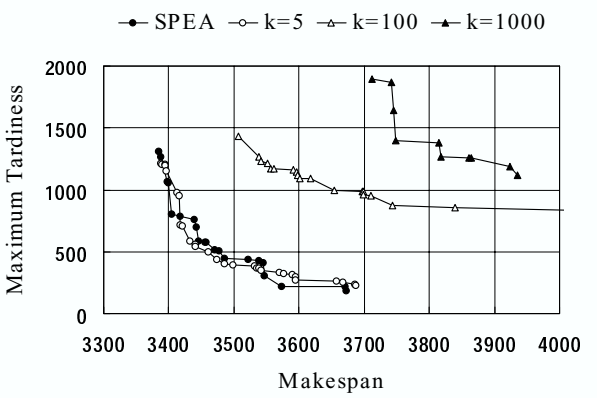

Fig. 6 Effect of early termination of local search

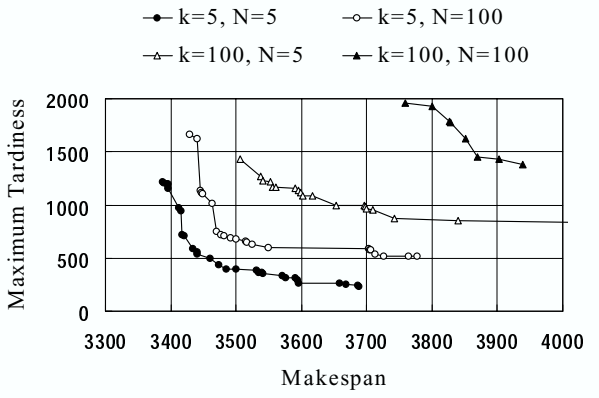

Fig. 7 Effect of the specifications of k and N.

In Fig. 8, we show the effect of the new parameter T on the performance of the hybrid algorithm. The other

parameter values were the same as the case of $k = 5$ in Fig. 6. We also show the result of the original SPEA, which can be viewed as the case of $T = \infty$. This figure shows that the search ability of the hybrid algorithm can be improved by appropriately specifying the value of T (i.e., the timing of the local search application). We can also see from Fig. 8 that slightly better results were obtained from the hybrid algorithm with $T < \infty$ than the non-hybrid SPEA with $T = \infty$.

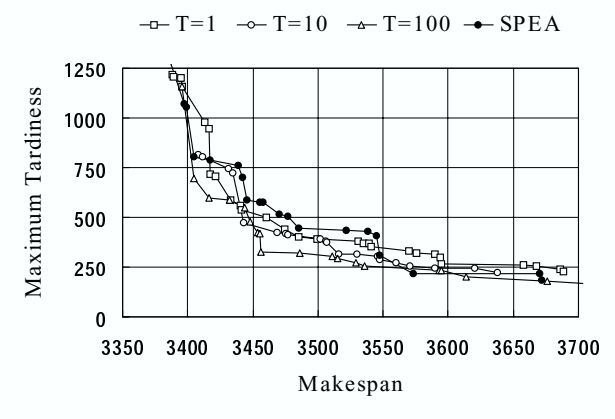

Fig. 8 Effect of the specification of *T*.

5. CONCLUSIONS

We examined six alternatives for specifying the objective function used by local search. Simulation results showed that better results were obtained from the three alternatives with the weighted sum approach than the two alternatives based on the dominance relation. It was also shown that the direct use of the fitness function of the SPEA as the objective function in local search needed longer CPU time than the other alternatives. Through computer simulations with various parameter specifications, we demonstrated that the hybridization with local search deteriorated the search ability of the SPEA when local search spent too much CPU time. In our computer simulations, we used only a single test problem. Thus further computer simulations on a number of test problems are required in order to obtain convincing conclusions.

Acknowledgement

The authors would like to thank the financial support from Japan Society for the Promotion of Science (JSPS) through Grand-in-Aid for Scientific Research (B): KAKENHI (14380194).

References

[1] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *Proc. of 1st International Conference on Genetic Algorithms and Their Applications*, pp. 93-100, 1985.

[2] C. A. Coello Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information Systems*, vol. 1, no. 3, pp. 269-308, 1999.

[3] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125-147, 2000.

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

[5] 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.

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

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

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002

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

[10] 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.

[11] A. Jaszkiewicz, "Genetic local search for multi-objective combinatorial optimization," *European J. of Operational Research*, vol. 137, pp. 50-71, 2002.

[12] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in hybrid evolutionary multi-criterion optimization algorithms," *Proc. of Genetic and Evolutionary Computation Conference*, pp. 1301-1308, July 9-13, 2002.