Incorporation of Decision Maker's Preference into Evolutionary Multiobjective Optimization Algorithms

Hisao Ishibuchi, Yusuke Nojima, Kaname Narukawa, and Tsutomu Doi
Department of Computer Science and Intelligent Systems
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
+81-72-254-9350
{hisaoi, nojima}@cs.osakafu-u.ac.jp, {kaname, doi}@ci.cs.osakafu-u.ac.jp

ABSTRACT
The main characteristic feature of evolutionary multiobjective optimization (EMO) is that no *a priori* information about the decision maker's preference is utilized in the search phase. EMO algorithms try to find a set of well-distributed Pareto-optimal solutions with a wide range of objective values. It is, however, very difficult for EMO algorithms to find a good solution set of a multiobjective combinatorial optimization problem with many decision variables and/or many objectives. In this paper, we propose an idea of incorporating the decision maker's preference into EMO algorithms to efficiently search for Pareto-optimal solutions of such a hard multiobjective optimization problem.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic Methods.*

General Terms
Algorithms.

Keywords
Evolutionary multiobjective optimization (EMO), many-objective optimization, multiobjective combinatorial optimization, decision maker's preference, balance between convergence and diversity.

1. INTRODUCTION
Evolutionary multiobjective optimization (EMO) algorithms have been successfully applied to multiobjective optimization problems in various application areas [1]. EMO algorithms are designed to find a set of well-distributed Pareto-optimal solutions with a wide range of objective values. It is, however, very difficult for EMO algorithms to find such a good solution set of a large-scale combinatorial multiobjective optimization problem with many decision variables and/or many objectives as pointed out in the literature. This is the case even when multiobjective optimization problems have only two objectives [6], [7]. Of course, it is more difficult for EMO algorithms to find good solution sets of many-objective optimization problems [4], [5].

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

In this paper, we propose an idea of incorporating the decision maker's preference in EMO algorithms. More specifically, we implement a hybrid algorithm of NSGA-II [2] and the decision maker's preference. In our hybrid algorithm, the decision maker's preference is used for parent selection whereas Pareto ranking and a crowding measure are used for generation update as in NSGA-II.

2. HYBRID ALGORITHM
A number of optimization techniques have been proposed in the area of multi-criteria decision making (MCDM) to search for a single final solution of a multiobjective optimization problem using the decision maker's preference. The basic idea of our hybrid approach is to use the decision maker's preference to improve the convergence of solutions to the desired area while keeping the diversity of solutions. We implement this idea by modifying only the parent selection phase of NSGA-II in the following manner.

Parent Selection: We use a scalarizing function defined by the given preference information to select a pair of parent solutions. For example, we use a weighted sum fitness function when the relative importance of each objective is given as the preference information. On the other hand, when the minimum requirement level for each objective is given, we use a penalized objective function in the parent selection phase. We can also use the distance from a reference solution for parent selection when the decision maker's preference is given as the reference solution.

3. COMPUTATIONAL EXPERIMENTS
We applied NSGA-II to the two-objective 500-item 0/1 knapsack problem of Zitzler & Thiele [8] using the following parameter specifications:

Population size: 200 individuals,
Crossover probability: 0.8 (uniform crossover),
Mutation probability: 1/500 (bit-flip mutation),
Termination condition: 2000 generations.

When infeasible solutions were generated by genetic operations, we used a repair method based on a maximum profit/weight ratio as in Zitzler & Thiele [8]. In Figure 1, we show the 50% attainment surface [3] over 100 runs of NSGA-II and an example of a solution set by its single run. For comparison, we also show the true Pareto front.

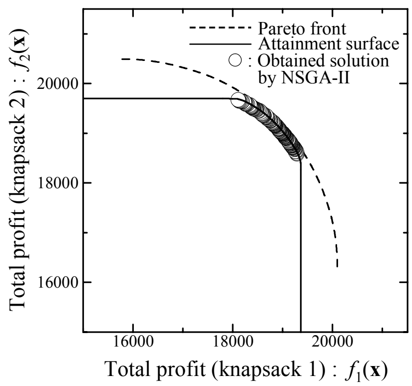

Figure 1. Experimental results on the 2-500 test problem.

From Figure 1, we can see that the obtained solutions do not have enough diversity if compared with the true Pareto front. If the decision maker wants to have a solution around the center area of the Pareto front, NSGA-II can provide good candidate solutions. On the other hand, if the decision maker wants to have an extreme solution with a very good value of one objective, NSGA-II can not provide good candidate solutions in Figure 1.

Let us assume that the decision maker wants to have a solution around the reference solution G in Figure 2. Whereas NSGA-II did not work well in this case, a good solution was obtained by minimizing the distance from G using a single-objective genetic algorithm (SOGA) as shown by the bold circle B in Figure 2. Other circles were obtained by our hybrid algorithm. Our hybrid algorithm found a number of good candidate solutions in Figure 2 (compare Figure 2 with Figure 1). In Figure 3, we used the scalar fitness function with the weight vector (0.1, 0.9). On the other hand, we used the minimum requirement level ε_1=17000 for the first objective in Figure 4. Our hybrid algorithm found a number of good candidate solutions in Figure 3 and Figure 4.

4. CONCLUDING REMARKS

In this paper, we proposed an idea of incorporating *a priori* information about the decision maker's preference into EMO algorithms. The point of the proposed idea is to hybridize the two approaches to multiobjective optimization: EMO and MCDM. In our approach, *a priori* information about the decision maker's preference is used to efficiently search for Pareto-optimal solutions as in MCDM. In this sense, our approach is different from EMO. On the other hand, our approach presents a number of candidate solutions to the decision maker as in EMO. In this sense, our approach is different from MCDM.

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

5. REFERENCES

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

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

[3] Fonseca, C. M., and Fleming, P. J. On the performance assessment and comparison of stochastic multiobjective optimizers. *Lecture Notes in Computer Science 114*, Springer, Berlin (September 1996) 584-593.

[4] Hughes, E. J. Evolutionary many-objective optimization: many once or one many?. *Proc. of 2005 IEEE Congress on Evolutionary Computation* (Edinburgh, UK, September 2-5, 2005) 222-227.

[5] Jaszkiewicz, A. On the computational efficiency of multiple objective metaheuristics: The knapsack problem case study. *European Journal of Operational Research 158*, 2, (October 2004) 418-433.

[6] Kumar, R., and Banerjee, N. Running time analysis of a multiobjective evolutionary algorithm on simple and hard problems. *Lecture Notes in Computer Science* 3469, Springer, Berlin (January 2005) 112-131.

[7] Pelikan, M., Sastry, K., and Goldberg, D. E. Multiobjective hBOA, clustering, and scalability. *Proc. of 2005 Genetic and Evolutionary Computation Conference* (Washington DC, June 25-29, 2005) 663-670.

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

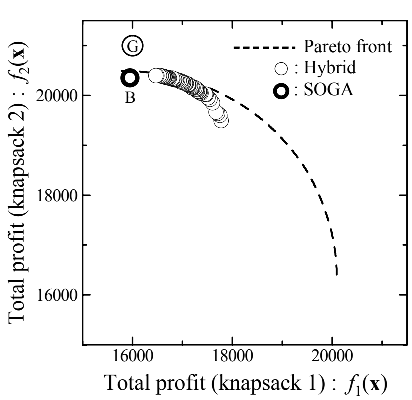

Figure 2. Reference solution G.

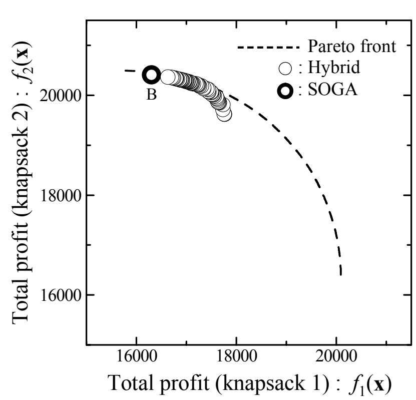

Figure 3. Weight vector (0.1, 0.9).

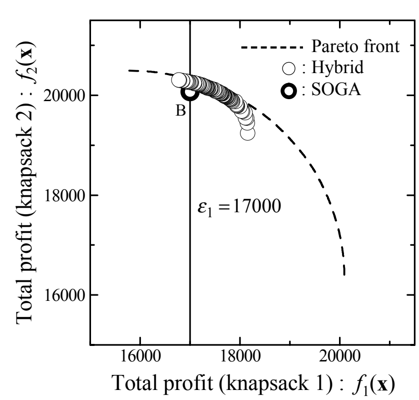

Figure 4. Requirement level 17000.