The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Hybridization of Evolutionary Multiobjective Optimization Algorithms by the Adaptive Use of Scalarizing Fitness Function

Noritaka Tsukamoto, Yuji Sakane, Yusuke Nojima, and Hisao Ishibuchi

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
(Tel : 81-72-254-9198; Fax : 81-72-254-9915)
({nori@ci., sakane@ci., nojima@, hisaoi@}cs.osakafu-u.ac.jp)

Abstract: This paper proposes an idea of adaptively using a scalarizing fitness function in evolutionary multiobjective optimization (EMO) algorithms. In our former study, we proposed a hybrid EMO algorithm, where we introduced two probabilities to specify how often the scalarizing fitness function is used for parent selection and generation update in EMO algorithms. In this paper, we use the ratio of non-dominated solutions to specify how often the scalarizing fitness function is used in our hybrid EMO algorithm. Through computational experiments on multiobjective 0/1 knapsack problems, we show the effectiveness of adaptively using a scalarizing fitness function.

Keywords: Evolutionary multiobjective optimization (EMO) algorithms, multiobjective problems, many-objective problems, scalarizing fitness function, multiobjective 0/1 knapsack problems.

I. INTRODUCTION

Evolutionary multiobjective optimization (EMO) is one of the most active research areas of evolutionary computation. Most EMO algorithms use Pareto ranking to evaluate the fitness of each solution. Whereas Pareto ranking-based EMO algorithms usually work well on two-objective problems, their search ability is severely degraded by the increase in the number of objectives. Well-known Pareto ranking-based EMO algorithms such as NSGA-II [1] and SPEA [2] do not work well on many-objective problems with four or more objectives. This is because almost all solutions in each population become non-dominated with each other when they are compared using many objectives. That is, almost all solutions have the same fitness with respect to Pareto ranking. On the other hand, Hughes [3] showed that multiple runs of single-objective evolutionary algorithms (SOEAs) outperformed a single run of EMO algorithms in their applications to many-objective problems. In some studies, EMO algorithms can outperform SOEAs even when they are used to solve single-objective problems. It was also reported that better results were obtained from transforming single-objective problems into multi-objective ones.

Some related studies suggest that SOEAs and EMO algorithms have their own advantages and disadvantages. In our former study [4], we hybridized them into a single algorithm in order to simultaneously utilize their advantages. Following this idea, we implemented a hybrid EMO algorithm using NSGA-II and a weighted sum fitness function. The weighted sum fitness function is probabilistically used for parent selection and generation update in our hybrid EMO algorithm. We introduced two probabilities to specify how often the scalarizing fitness function is used for parent selection and generation update in our hybrid EMO algorithms. We showed that the use of the weighted sum fitness function improved the performance of NSGA-II for multiobjective optimization.

In this paper, we propose an idea of adaptively using the weighted sum fitness function in our hybrid EMO algorithm. We use the ratio of non-dominated solutions to specify how often the weighted sum fitness function is used in our hybrid EMO algorithm. The weighted sum fitness function is adaptively used for parent selection and generation update in this paper. Through computational experiments on multiobjective 0/1 knapsack problems, we show the effectiveness of adaptively using the weighted sum fitness function.

II. HYBRID EMO ALGORITHM

In this section, we explain about our hybrid EMO algorithm. In our former study [4], we implemented a hybrid EMO algorithm by incorporating a weighted sum fitness function into NSGA-II [1]. We introduced two probabilities P_{PS} and P_{GU} to specify how often the weighted sum fitness function is used for parent selection and generation update, respectively.

1. NSGA-II

NSGA-II [1] is a well-known and frequently-used EMO algorithm with the ($\mu+\lambda$)-ES generation update mechanism. The outline of NSGA-II is as follows:

[NSGA-II]
Step 1: P = Initialize (P)
Step 2: While the stopping condition is not satisfied, do
Step 3: P' = Parent Selection (P)
Step 4: P'' = Genetic Operations (P')
Step 5: P = Generation Update ($P \cup P''$)
Step 6: End while
Step 7: Return Non-dominated (P)

In NSGA-II, each solution in the current population is evaluated using Pareto ranking and crowding distance in the following manner in Step 3. First, Rank 1 is assigned to all the non-dominated solutions in the current population. Solutions with Rank 1 are tentatively removed from the current population. Next, Rank 2 is assigned to all the non-dominated solutions in the remaining population. Solutions with Rank 2 are tentatively removed from the remaining population. In the same manner, ranks are assigned to all solutions in the current population. Solutions with smaller rank values are viewed as being better than those with larger rank values. A crowding distance is used to compare solutions with the same rank. The crowding distance of a solution is the Manhattan distance between its two adjacent solutions in the objective space (for details, see [1]). When two solutions have the same rank, one solution with a larger value of crowding distance is viewed as being better than the other with a smaller value.

A prespecified number of pairs of parent solutions are selected from the current population by binary tournament selection to form a parent population P' in Step 3. An offspring solution is generated from each pair of parent solutions by genetic operations to form an offspring population P'' in Step 4. The current population and the offspring population are merged to form an enlarged population. Each solution in the enlarged population is evaluated by Pareto ranking and the crowding distance as in the parent selection phase. A prespecified number of the best solutions are chosen from the enlarged population as the next population P in Step 5.

2. Weighted sum fitness function

The weighted sum fitness function of the k objectives is as follows:

$$fitness(\mathbf{x}) = w_1 \cdot f_1(\mathbf{x}) + w_2 \cdot f_2(\mathbf{x}) + \ldots + w_k \cdot f_k(\mathbf{x}) \quad (1)$$

where $f_i(\mathbf{x})$ is the i-th objective value of $\mathbf{x}$, w_i is a non-negative weight value.

We generate a set of non-negative integer vectors satisfying the following relation: $w_1 + w_2 + \ldots + w_k = d$ where d is a prespecified integer. In this paper, we specify $d = 4$ for two, three, and four-objective problems. On the other hand, we specify $d = k$ for five and six-objective problems where k is the number of objectives. For example, we have five integer vectors: (4, 0), (3, 1), (2, 2), (1, 3), and (0, 4) for two-objective problems. For three-objective problems, we have 15 integer vectors: (4, 0, 0), (3, 1, 0), …, (0, 0, 4). For four, five, and six-objective problems, we have 35, 126 and 462 integer vectors, respectively.

3. Hybrid EMO algorithm

Our hybrid EMO algorithm is the same as NSGA-II except for parent selection in Step 3 and generation update in Step 5. When a pair of parent solutions is selected from the current population, the weighted sum fitness function and the NSGA-II fitness evaluation mechanism are used with the probabilities P_{PS} and (1− P_{PS}), respectively. When another pair of parent solutions is to be selected, the probabilistic choice between two fitness evaluation schemes is performed again.

As in the parent selection phase in Step 3, we probabilistically use the weighted sum fitness function in generation update phase in Step 5. When one solution is to be selected and added to the next population, the weighted sum fitness function and the NSGA-II fitness evaluation mechanism are used with the probabilities P_{GU} and (1− P_{GU}), respectively. When another solution is to be selected, the probabilistic choice between two fitness evaluation schemes is performed again.

It should be noted that our hybrid EMO algorithm with P_{PS} = 0.0 and P_{GU} = 0.0 is the same as the pure NSGA-II [1] since the weighted sum fitness function is never used. On the other hand, our hybrid EMO algorithm with P_{PS} = 1.0 and P_{GU} = 1.0 is a weighted sum-based genetic algorithm with the ($\mu+\lambda$)-ES generation update mechanism. In our former study [4], we used fixed values P_{PS} and P_{GU} during the execution of our

Table 1. Hyprevolume measure over 50 runs in subsection 3.2.

	2-500 ($\times 10^8$)	3-500 ($\times 10^{12}$)	4-500 ($\times 10^{17}$)	5-500 ($\times 10^{21}$)	6-500 ($\times 10^{25}$)
NSGA-II	3.80 (0.0160)	6.53 (0.0440)	1.01 (0.0085)	1.74 (0.0185)	2.60 (0.0286)
Hybrid$_{PS}$	3.78 (0.0152)	6.48 (0.0388)	1.00 (0.0081)	1.74 (0.0138)	2.62 (0.0237)
Hybrid$_{GU}$	3.85 (0.0179)	6.71 (0.0410)	1.08 (0.0070)	1.89 (0.0161)	2.91 (0.0297)
Hybrid$_{adapt}$	**3.91 (0.0145)**	**6.87 (0.0409)**	**1.11 (0.0094)**	**1.93 (0.0214)**	**2.92 (0.0381)**

hybrid algorithm.

In this paper, we adaptively use the weighted sum fitness function. In short, we use various values P_{PS} and P_{GU} during the execution of our hybrid algorithm. P_{PS} is the ratio of non-dominated solutions in the current population in Step 3 whereas P_{GU} is the ratio of non-dominated solutions in the enlarged population in Step 5.

III. PERFORMANCE EVALUATION

In this section, we examine the performance of adaptively using the weighted sum fitness function. First, we compare three versions of our idea in subsection 3.2. Next, we compare our idea with our original hybrid algorithm [4] in subsection 3.3.

1. Parameter settings

As test problems, we use a two-objective 500-item, a three-objective 500-item, and a four-objective 500-item knapsack problem in [2]. These test problems are denoted as 2-500, 3-500, and 4-500, respectively. As many-objective problems, we generate a five-objective 500-item (i.e., 5-500) and a six-objective 500-item (i.e., 6-500) knapsack problem in the same manner as [2]. We use the following parameter specifications in this section:

Population size: 200 (i.e., $\mu = \lambda = 200$),
Crossover probability: 0.8 (uniform crossover),
Mutation probability: 1/500 (bit-flip mutation),
Stopping condition: 400,000 fitness evaluations.

Each algorithm is applied to each test problem 50 times. As a performance measure, we use the hypervolume measure [5] that calculates the volume of the dominated region by the non-dominated solution set in the objective space. The hypervolume measure is used as the diversity and the convergence measure. For the 2-500 problem, we show the 50% attainment surface [6] in order to visually show the behavior of each algorithm.

2. Comparison among three versions of our idea

In this subsection, we examine the following versions of our idea:

Hybrid$_{PS}$: The weighted sum fitness function is adaptively used for parent selection in Step 3.

Hybrid$_{GU}$: The weighted sum fitness function is adaptively used for generation update in Step 5.

Hybrid$_{adapt}$: The weighted sum fitness function is adaptively used for parent selection in Step 3 and generation update in Step 5.

We also use the pure NSGA-II [1] to compare three versions of our idea. In Table 1, we show the average value and the standard deviation value (in the parentheses) of the hypervolume measure over 50 runs for each problem. The best value and the worst value are highlighted by bold and underline, respectively.

Table 1 shows that the best result for each problem was obtained from Hybrid$_{adapt}$. From this table, we show the effectiveness of adaptively using the weighted sum fitness function for parent selection phase and generation update phase. On the other hand, Hybrid$_{PS}$ did not obtain good results for each problem. We cannot observe the effect of adaptively using the weighted sum fitness function only for parent selection phase. In Fig. 1, we show the 50% attainment surface over 50 runs of each algorithm: Hybrid$_{PS}$, Hybrid$_{GU}$, Hybrid$_{adapt}$, and NSGA-II. From Fig. 1, Hybrid$_{adapt}$ improved the diversity of obtained non-dominated solutions. On the other hand, Hybrid$_{PS}$ obtained the similar result with NSGA-II. From these results, it is shown that adaptively using the weighted sum fitness function only for parent selection phase did not improve the search ability of NSGA-II whereas adaptively using the weighted sum fitness function for generation update phase had a positive effect.

Table 2. Hyprevolume measure over 50 runs in subsection 3.3.

	2-500 ($\times 10^8$)	3-500 ($\times 10^{12}$)	4-500 ($\times 10^{17}$)	5-500 ($\times 10^{21}$)	6-500 ($\times 10^{25}$)
NSGA-II	3.80 (0.0160)	6.53 (0.0440)	1.01 (0.0085)	1.74 (0.0185)	2.60 (0.0286)
Hybrid$_{0.5_0.5}$	3.89 (0.0154)	6.78 (0.0400)	1.08 (0.0069)	1.88 (0.0131)	2.86 (0.0309)
Weighted Sum	**3.95 (0.0121)**	**6.95 (0.0416)**	**1.12 (0.0086)**	**1.93 (0.0193)**	2.86 (0.0377)
Hybrid$_{adapt}$	3.91 (0.0145)	6.87 (0.0409)	1.11 (0.0094)	**1.93 (0.0214)**	**2.92 (0.0381)**

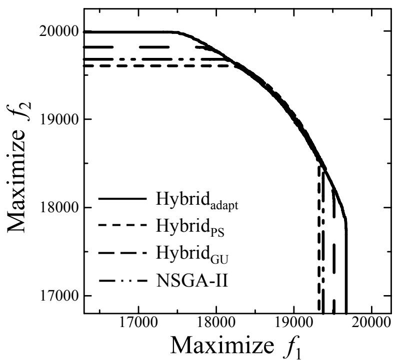

Fig. 1. 50% attainment surface in subsection 3.2.

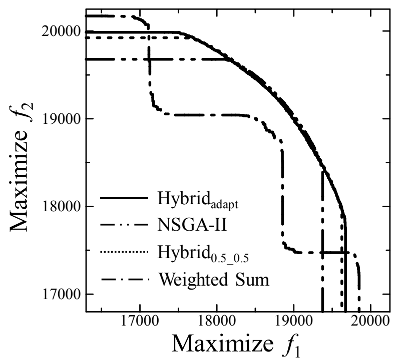

Fig. 2. 50% attainment surface in subsection 3.3.

3. Comparison with original hybrid algorithm

In this subsection, we compare our idea Hybrid$_{adapt}$ with our original hybrid algorithm [4]. In our original hybrid algorithm, we use the following parameters for P_{PS} and P_{GU}:

P_{PS}: 0.0, 0.5, 1.0,
P_{GU}: 0.0, 0.5, 1.0.

We examine the 3 × 3 combinations of the 3 values of P_{PS} and P_{GU}. Due to the page limitation, we use 3 combinations of P_{PS} and P_{GU}: $P_{PS} = P_{GU} = 0.0$, $P_{PS} = P_{GU} = 0.5$, $P_{PS} = P_{GU} = 1.0$. As stated in subsection 2.3, our original hybrid algorithm with $P_{PS} = P_{GU} = 0.0$ is the same as the pure NSGA-II whereas our original hybrid algorithm with $P_{PS} = P_{GU} = 1.0$ is a weighted sum-based genetic algorithm with ($\mu+\lambda$)-ES generation update mechanism. We denote the combination of $P_{PS} = P_{GU} = 0.5$ as Hybrid$_{0.5_0.5}$. In Table 2, we show the average value and the standard deviation value of the hypervolume measure over 50 runs for each problem. From Table 2, Hybrid$_{adapt}$ obtained relatively good results for each problem. Especially, Hybrid$_{adapt}$ obtained the best result for two problems with five or six objectives. In Fig. 2, we show the 50% attainment surface over 50 runs of each algorithm: NSGA-II, Hybrid$_{0.5_0.5}$, Weighted Sum, and Hybrid$_{adapt}$. From Fig. 2, adaptively using the weighted sum fitness function had relatively good effect on the search ability of our original hybrid algorithm.

VI. CONCLUSION

This paper proposed an idea of adaptively using a scalarizing fitness function in our hybrid EMO algorithm. We used the ratio of the non-dominated solutions to specify how often the scalarizing function is used in our hybrid EMO algorithm. Through computational experiments, we showed the effectiveness of our idea.

REFERENCES

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. on Evolutionary Computation*, 6, 2, 182-197.
[2] E. Zitzler and L. Thiele (1999), Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, *IEEE Trans. on Evolutionary Computation*, 3, 4, 257-271.
[3] E. J. Hughes (2005), Evolutionary many-objective optimization: Many once or one many?, *Proc. of CEC 2005*, 222-227.
[4] H. Ishibuchi, T. Doi, and Y. Nojima (2006), Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization, *LNCS 4193: PPSN IX*, 493-502, Springer, Berlin.
[5] E. Zitler and L. Thiele (1998), Multiobjective optimization using evolutionary algorithms – A comparative case study, *LNCS: 1498: PPSN V*, 292-301, Springer, Berlin.
[6] C. M. Fonseca and P. J. Fleming (1996), On the performance assessment and comparison of stochastic multiobjective optimizers, *LNCS 1141: PPSN IV*, 584-593, Springer, Berlin.