A Distribution Control of Weight Vector Set for Multi-objective Evolutionary Algorithms

Tomoaki Takagi(✉), Keiki Takadama, and Hiroyuki Sato

The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
tomtkg@uec.ac.jp, keiki@inf.uec.ac.jp, sato@hc.uec.ac.jp

Abstract. For solving multi-objective optimization problems with evolutionary algorithms, the decomposing the Pareto front by using a set of weight vectors is a promising approach. Although an appropriate distribution of weight vectors depends on the Pareto front shape, the uniformly distributed weight vector set is generally employed since the shape is unknown before the search. This work proposes a simple way to control the weight vector distribution appropriate for several Pareto front shapes. The proposed approach changes the distribution of the weight vector set based on the intermediate objective vector in the objective space. A user-defined parameter determines the intermediate objective vector in the static method, and the objective values of the obtained solutions dynamically determine the intermediate objective vector in the dynamic method. In this work, we focus on MOEA/D as a representative decomposition-based multi-objective evolutionary algorithm and apply the proposed static and dynamic methods for it. The experimental results on WFG test problems with different Pareto front shapes show that the proposed static and dynamic methods improve the uniformity of the obtained solutions for several Pareto front shapes and the dynamic method can find an appropriate intermediate objective vector for each Pareto front shape.

Keywords: Multi-objective optimization · Evolutionary computation

1 Introduction

Real-world optimization problems often involve multiple conflicting objectives. These problems are said to be multi-objective optimization problems. The aim of multi-objective optimization is to acquire a set of solutions approximating the Pareto front, the optimal trade-off among conflicting objectives. So far, evolutionary algorithms have been intensively studied for solving multi-objective optimization problems [1,2]. As an evolutionary algorithm solving multi-objective problems, we focus on MOEA/D which decomposes the Pareto front in the

A. Compagnoni et al. (Eds.): BICT 2019, LNICST 289, pp. 70–80, 2019.
https://doi.org/10.1007/978-3-030-24202-2_6

objective space [3]. MOEA/D simultaneously optimize a number of scalarizing functions with different weight vectors and tries to approximate the Pareto front by the obtained solutions paired with weight vectors. MOEA/D needs to prepare the set of weight vectors before the search. Since each weight vector determines an approximation part of the Pareto front, the distribution of weight vectors strongly affects the distribution of the obtained solutions in the objective space. The appropriate distribution of weight vectors depends on the shape of the Pareto front. However, the shape of the Pareto front is generally unknown before the search. The conventional MOEA/D uses a uniformly distributed weight vectors based on the simplex-lattice design method. Although several methods arranging the weight vector set have been studied recently [4–7], we are investigating another simple way to re-arrange the weight vector set.

In this work, we propose an approach to re-arrange the weight vector set based on the intermediate objective value in the population. The approach is employed in the two proposed methods. The first one is the static method which determines the intermediate objective value by using an user-defined parameter. The another one is the dynamic method which determines the intermediate objective value by the objective values of the obtained solutions during the search. The effects of the two proposed methods are verified on tWFG4 problems [8] with different Pareto front shapes.

2 Evolutionary Multi-objective Optimization

A multi-objective optimization problem is defined as follows:

$$\text{Minimize} \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \tag{1}$$

Since these minimizations of m kinds of objective functions f_i $(i = 1, 2, \ldots, m)$ often conflict, there is no single optimal solution $\boldsymbol{x}$ which minimizes all objective values simultaneously. The optimal trade-off among objectives arises instead, and it is said to be the Pareto front. Therefore, the task of multi-objective optimization is to acquire a solution set approximating the Pareto front. Approximation qualities of the Pareto front can be evaluated by the convergence of solutions toward the Pareto front, the spread of them to cover the entire Pareto front, and the distribution uniformity of them in the objective space.

Although there are several approaches to address multi-objective optimization, population-based evolutionary algorithms have an advantage that a set of solutions approximating the Pareto front can be obtained in a single run. In this work, we focus on MOEA/D [3] as a representative evolutionary algorithm for solving multi-objective optimization problems.

3 MOEA/D

MOEA/D specifies N approximating parts of the Pareto front with a weight vector set $\mathcal{L} = (\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N)$. Each weight vector $\boldsymbol{\lambda}^i$ has m kinds of elements $(\lambda_1^i, \lambda_2^i, \ldots, \lambda_m^i)$. The decomposition parameter H and the number of objectives m are used to generate the weight vector set $\mathcal{L}$. Each element of $\boldsymbol{\lambda}^i = (\lambda_1^i, \lambda_2^i, \ldots, \lambda_m^i)$ is one of $\{0/H, 1/H, \ldots, H/H\}$, and all $N = C_{H+m-1}^{m-1}$ kinds of weight vectors satisfying $\sum_{j=1}^{m} \lambda_j^i = 1.0$ are employed in the search. These weight vectors are uniformly distributed on the m-dimensional hyperplane, and it is said to be the simplex-lattice design.

Since each weight vector $\boldsymbol{\lambda}^i$ is paired with one solution $\boldsymbol{x}^i$, the size of the population $\mathcal{P} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$ becomes N which is equivalent to the size of the weight vector set $\mathcal{L}$. MOEA/D compares solutions based on their scalarizing function values calculated with their objective function vectors and a weight vector. For each weight vector, MOEA/D maintains the best solution with the minimum scalarizing function value in the population. Although there are several scalarizing functions, in this work, we use the reciprocal weighted Tchebycheff function (rTCH) [9]. The rTCH scalarizing function is formulated as

$$\text{Minimize} \quad g(\boldsymbol{x}|\boldsymbol{\lambda}^i) = \max_{1 \leq j \leq m} \left| \frac{f_j(\boldsymbol{x}) - z_j}{\lambda_j^i} \right|. \tag{2}$$

where, z_j $(j = 1, 2, \ldots, m)$ is an element of the obtained ideal point $\boldsymbol{z}$. z_j is set to the minimum (best) objective function value f_j found during the search.

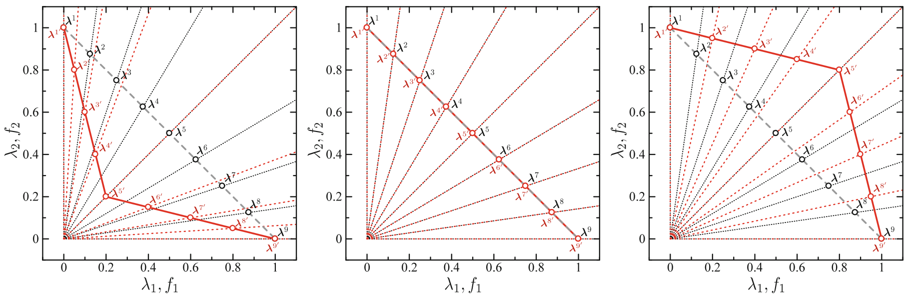
4 Issue Focus: Weight Vector Distribution

The distribution of the obtained solutions in the objective space is affected by the distribution of the weight vectors. Since the Pareto front shape is generally unknown before the search, the conventional MOEA/D uses the weight vector set uniformly distributed on the m-dimensional hyperplane. However, the weight distribution is appropriate only when the shape of the Pareto front is the hyperplane. For other Pareto front shapes such as convex and concave shapes, their appropriate distributions of weight vectors are different. If the distribution of the weight vectors can be changed according to the Pareto front shape, the distribution uniformity of the obtained solutions can be improved.

Several approaches varying the distribution of weight vectors have been studied so far [4–7]. Jiang et al. proposed a uniformly distributed weight vectors based on the mixture uniform design method [4]. This method changes the curvature of the hyperplane on the weight vectors. Deb et al. proposed a method to change weight vector (reference point) distribution by deleting and adding weight vectors during the search [5]. On the other hand, Hamada et al. proposed adding and re-arranging mechanism of the weight vectors during the search [6,7]. In this work, we propose an alternative way to change the weight vector distribution.

5 Proposed Method: Weight Vector Distribution Control Based on Intermediate Objective Value

The proposed method varies the conventional weight vector set $\mathcal{L}$ to $\mathcal{L}' = \{\boldsymbol{\lambda}^{1'}, \boldsymbol{\lambda}^{2'}, \ldots, \boldsymbol{\lambda}^{N'}\}$ by using the intermediate objective value p. p is a real value in the range $[0, 1]$.



(a) $p = 0.2$ (b) $p = 0.5$ (c) $p = 0.8$

Fig. 1. Weight vector distributions by varying intermediate objective value p

In this work, we proposed two methods. One is the static approach using p as a static user-defined parameter, and another one is the dynamic approach determining p based on the objective values in the population during the search.

5.1 Static Approach

In the static approach, we transform each weight vector $\boldsymbol{\lambda}^i = (\lambda_1^i, \lambda_2^i, \ldots, \lambda_m^i)$ into $\boldsymbol{\lambda}^{i'} = (\lambda_1^{i'}, \lambda_2^{i'}, \ldots, \lambda_m^{i'})$ with an user-defined intermediate objective value p in the range $[0, 1]$. Each element is transformed by

$$\lambda_j^{i'} = \begin{cases} \lambda_j^i \cdot p \cdot m, & \text{if } \lambda_j^i \leq \frac{1}{m}, \\ 1 - (1 - \lambda_j^i) \cdot \frac{1-p}{m-1} \cdot m, & \text{otherwise.} \end{cases} \quad (3)$$

Figure 1 shows three examples of transformed weight vector sets with different p on $m = 2$ objective space. We can see that the weight distribution with $p = 1/m = 0.5$ is the equivalent to the conventional weight distribution. We can see that the weights with $p < 0.5$ get close to the origin point. This distribution is appropriate for the convex Pareto front. Also, we can see that the weights with $p > 0.5$ get away from the origin point. This distribution is appropriate for the concave Pareto front.

5.2 Dynamic Approach

The dynamic approach determines the intermediate value p based on the objective values of solutions in the population during the search.

In each generation, the proposed dynamic approach calculates the intermediate objective value p by

$$p = \frac{1}{\sqrt{m}} \cdot d_1(\boldsymbol{x}_L), \quad (4)$$

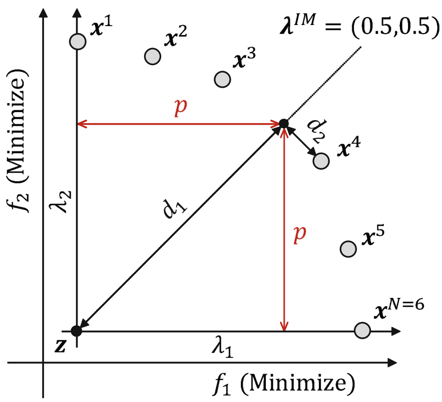


Fig. 2. Calculation of the intermediate value p in the proposed dynamic approach

where,

$$\boldsymbol{x}_L = \underset{\boldsymbol{x}\in\{\boldsymbol{x}^1,\boldsymbol{x}^2,\ldots,\boldsymbol{x}^N\}}{\arg\min} \quad d_2(\boldsymbol{x}) \tag{5}$$

$$d_1(\boldsymbol{x}) = \frac{\|(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z})^T \boldsymbol{\lambda}^{IM})\|}{\|\boldsymbol{\lambda}^{IM}\|}, \tag{6}$$

$$d_2(\boldsymbol{x}) = \left\| \boldsymbol{f}(\boldsymbol{x}) - \left(\boldsymbol{z} - d_1(\boldsymbol{x}) \frac{\boldsymbol{\lambda}^{IM}}{\|\boldsymbol{\lambda}^{IM}\|} \right) \right\|. \tag{7}$$

The two distances d_1 and d_2 are employed from the concept of the penalty-based boundary intersection (PBI) [3]. $\boldsymbol{\lambda}^{IM} = (1/m, 1/m, \cdots, 1/m)$ is the intermediate weight vector specifying the center of the Pareto front.

In Eq. (4), first we find the landmark solution $\boldsymbol{x}_L$ with the minimum distance d_2 among the all solutions $\mathcal{P} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$. Figure 2 shows an example on an $m = 2$ objective problem. In this example, the landmark solution is $\boldsymbol{x}^4$ since it has the minimum distance d_2 to the intermediate weight vector $\boldsymbol{\lambda}^{IM} = (0.5, 0.5)$. In Eq. (4), next we calculate the distance d_1 of the landmark solution. Then, we calculate p as the side length of a hypercube which the length of the diagonal line is d_1. In the example of Fig. 2, p is the side length of a square which the length of the diagonal line is d_1 of $\boldsymbol{x}^4$. In the proposed dynamic approach, the intermediate objective value p is repeatedly updated every generation. That is, the weight distribution is repeatedly changed every generation.

5.3 Expected Effects

For a convex Pareto front, we can expect the weight distribution with $p < 0.5$ obtains more uniformly distributed solutions than the conventional weight vectors. For a concave Pareto front, we can expect the weight distribution with $p > 0.5$ obtains more uniformly distributed solutions than the conventional weight vectors. In the case of the static approach, we need to set a p value

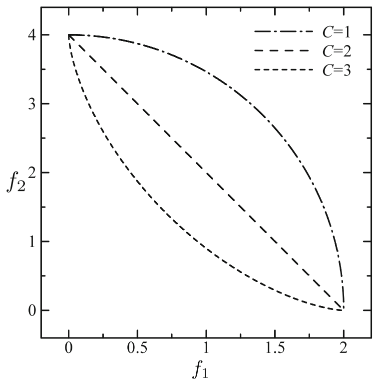


Fig. 3. Pareto fronts with different C values

before the search. To find an appropriate p, we need the parameter tuning of p by repeating run of the algorithm with different p. On the other hand, in the dynamic approach, we can expect to obtain more uniformly distributed solutions than the conventional weights from the convex to the concave Pareto front while searching an appropriate intermediate objective value p during the search.

6 Experimental Setting

6.1 Test Problems

To verify the effectiveness of the proposed methods, in this work, we use tWFG4 [8] problems extended from WFG4 [10] problems. tWFG4 is formulated by

$$f_j(\boldsymbol{x}) = 2j \cdot \left(\frac{f_j^{WFG4}(\boldsymbol{x})}{2j} \right)^C \quad (j = 1, 2, \ldots, m), \tag{8}$$

where, f_j^{WFG4} is j-th objective function of WFG4 problem [10], C is a parameter specifying the Pareto front shape. tWFG4 with $C = 2$ has a liner Pareto front. tWFG4 with $C < 2$ has a concave Pareto front. tWFG4 with $C > 2$ has a convex Pareto front. In this work, we use tWFG4 problems with $C = \{1, 2, 3\}$. Figure 3 shows three Pareto fronts with different C.

f_j^{WFG4} $(j = 1, 2, \ldots, m)$ involves two problem parameters. First one is the distance parameter L to control the convergence difficulty. Another one is the spread parameter K to control the spread difficulty. To evaluate the uniformity of the obtained solutions, in this work, we respectively set them $K = 1+(m-1)-1 \mod (m-1)$ and $L = 1$. Consequently, the number of variables becomes $n = K + L = \{3, 3, 4\}$ for $m = \{2, 3, 4\}$. That is, the uniformity of the solutions strongly affect the search performance.

6.2 Parameters

We use tWFG4 problems with $m = \{2, 3, 4\}$ objectives and $C = \{1, 2, 3\}$. For $m = \{2, 3, 4\}$ objective problems, the decomposition parameters are respectively set to $H = \{200, 19, 9\}$, and the population sizes become $N = \{201, 210, 220\}$ in MOEA/D. Also, we employ commonly used SBX with the distribution index $\eta_c = 20$ and the crossover ratio 0.8 and the polynomial mutation with the distribution index $\eta_m = 20$ and the mutation ratio $1/n$. Also, the neighborhood size is set to $T = 20$, and the total number of generations is set to 3,000.

6.3 Metric

To evaluate the obtained solutions, we employ Hypervolume (HV) [11]. HV is a volume determined by the obtained solutions and the reference point $\boldsymbol{r}$ in the objective space. The convergence of solutions toward the Pareto front, the spread of solutions in the objective space, and the uniformity of solutions in the objectives space affects to the HV value. Before we calculate HV, we each normalize objective value as $f'_j(\boldsymbol{x}) = f_j(\boldsymbol{x})/2j$ $(j = 1, 2, \ldots, m)$. Then we calculate HV with the reference point $\boldsymbol{r} = \{1.1, 1.1, \ldots, 1.1\}$. In this work, we compare the median HV of all independent 31 runs.

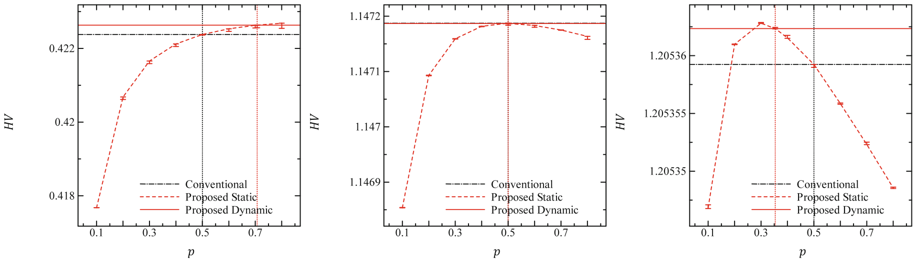
7 Experimental Results and Discussion

7.1 Search Performance *HV*

Figures 4, 5 and 6 show results of HV on problems with different Pareto front shapes $C = \{1, 2, 3\}$ and number of objectives $m = \{2, 3, 4\}$, respectively. In each graph, the horizontal axis indicates the intermediate objective value p of the static approach. The black vertical line on $p = 1/m$ indicates the conventional weight distribution based on the simplex-lattice design. The black horizontal line indicates the HV value obtained by the conventional weight distribution. Also, the red horizontal line indicates the HV value of the proposed dynamic approach. The red vertical line indicates the intermediate objective value p of the proposed dynamic approach at the final generation. The red dot line with makers indicates HV values of the proposed static approach. In all graphs, the error bars are the maximum and the minimum HV values among 31 runs.

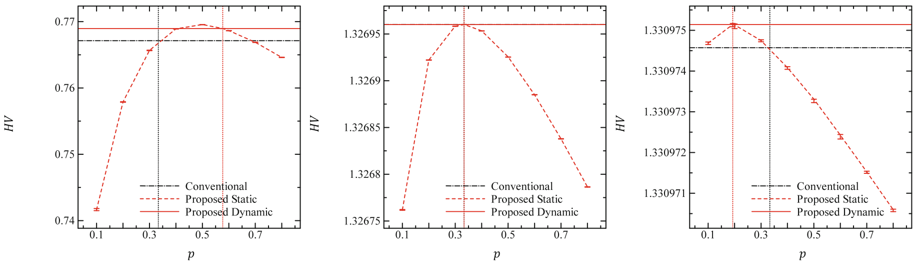
First, from Fig. 4 on $m = 2$ objective problems with a plane Pareto front ($C = 2$), we can see that the proposed static approach achieves the highest HV when we use $p^* = 0.5$ which is the equivalent to the conventional weight vector set by simplex-lattice design. We can see that the proposed dynamic approach shows HV value comparable with the one obtained by the proposed static approach with $p^* = 0.5$.

Next, form the results on $m = 2$ objective problems with a concave Pareto front ($C = 1$), we can see that the proposed static approach achieves the highest HV when we use $p^* = 0.8$ which the transformed weight vectors get away from the origin point. Also, we can see that HV achieved by the static approach with



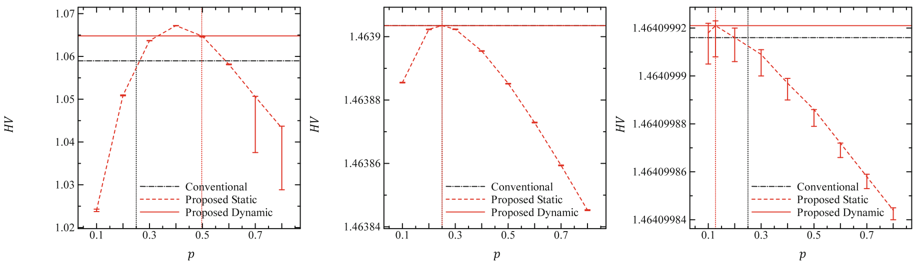
(a) $C = 1$ (Concave) (b) $C = 2$ (Plane) (c) $C = 3$ (Convex)

Fig. 4. Relationship between intermediate objective values p and HV in 2 objectives (Color figure online)



(a) $C = 1$ (Concave) (b) $C = 2$ (Plane) (c) $C = 3$ (Convex)

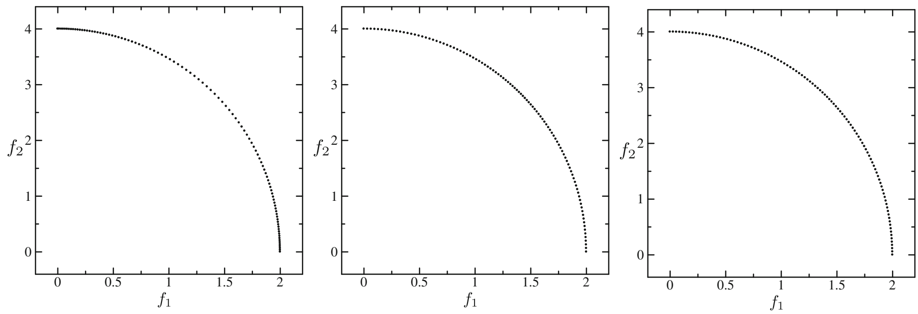
Fig. 5. Relationship between intermediate objective values p and HV in 3 objectives (Color figure online)



(a) $C = 1$ (Concave) (b) $C = 2$ (Plane) (c) $C = 3$ (Convex)

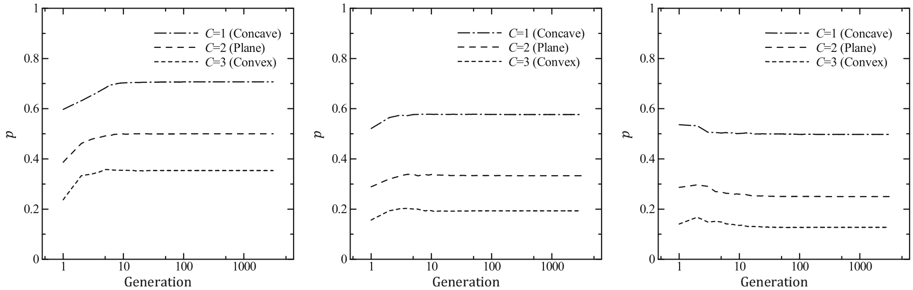
Fig. 6. Relationship between intermediate objective values p and HV in 4 objectives (Color figure online)

$p^* = 0.8$ is higher than HV obtained by the conventional weights with $p = 0.5$. This result reveals that the proposed re-arrangement of weight vectors improves HV value. Also, we can see that the proposed dynamic approach shows HV value close to HV values obtained by the static approaches with $p = \{0.7, 0.8\}$. The static approach has to determine an intermediate objective value p before the



(a) Conventional weights ($p = 0.5$) (c) Proposed static ($p^* = 0.8$) (c) Proposed dynamic

Fig. 7. Obtained solutions on the problem a concave Pareto front ($C = 1$)



(a) 2 objectives (b) 3 objectives (c) 4 objectives

Fig. 8. Transition of the dynamically determined intermediate objective value p

search. To find out the appropriate p^*, we need to repeatedly run the algorithm while slightly changing p. On the other hand, the proposed dynamic approach searches while adjusting the intermediate objective value p during the search. Therefore, the result reveals that the proposed dynamic approach achieves higher HV than the conventional method in a single run. Next, from the results on $m = 2$ objective problems with a convex Pareto front ($C = 3$), we can see that the proposed static approach achieves the highest HV when we use $p^* = 0.3$ which the weight vectors get close to the origin point. Thus, the optimal p^* maximizing HV decreases by increasing C and changing the Pareto front from a concave to a convex. Also, although the proposed dynamic approach cannot achieve the maximum HV obtained by the static approach with the optimal p^*, the proposed dynamic approach achieves higher HV than the conventional method in a single run. On $m = \{3, 4\}$ objective problems, we can see that the similar tendency observed on the $m = 2$ objective problem.

7.2 Obtained Solution Set

Figure 7 shows the obtained solution sets at the final generation on $m = 2$ objective problem with a concave Pareto front ($C = 1$). From the results obtained by the conventional weights, we can see that the approximation granularity around the edges of the Pareto front is high but one in the center of the Pareto front is low. On the other hand, we can see that two proposed methods obtain more uniformly distributed solutions than the conventional weights.

7.3 Transition of p in Dynamic Approach

Figure 8 shows transitions of the intermediate objective value p determined by the proposed dynamic approach. From the result, we can see that p values converge around ten generation on each of the problems with different Pareto front shapes C and the number of objectives. These stable transition of p reveals the stable adaptation of the weight vector distribution by the proposed dynamic approach.

8 Conclusions

To improve the uniformity of obtained solutions on problems with different Pareto front shapes, in this work, we proposed a method to control weight vector distribution based on the intermediate objective value. Experimental results showed that the proposed static approach improves the approximation performance on problems with a convex, a plane, and a concave Pareto fronts by setting an appropriate intermediate objective value. Also, we showed that the dynamic approach achieved higher approximation performance than the conventional weight vector distribution on problems with different Pareto front shapes without parameter tuning of the intermediate objective value. We focus on MOEA/D as a representative evolutionary algorithm for solving multi-objective optimization problems, I believe it is necessary to check whether it applies not only to MOEA/D but also to other decomposition base MOEAs including NSGA-III.

As future works, we will apply the proposed method to other decomposition-based algorithms such as NSGA-III and verify the effects on the search performance. We are designing an algorithm converting weight vector set based on multiple objective vectors.

References

1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley, New York (2001)
2. Coello, C.A.C., Veldhuizen, D.A.V., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, Boston (2002)
3. Zhang, Q., Li, H.: MOEA/D: a multi-objective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)

4. Jiang, S., Cai, Z., Zhang, J., Ong, Y.S.: Multiobjective optimization by decomposition with Pareto-adaptive weight vectors. In: Proceedings of 2011 Natural Computation (ICNC), pp. 1260–1264 (2011)
5. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: handling constraints and extending to an adaptive approach. IEEE Trans. Evol. Comput. **18**(4), 602–622 (2014)
6. Hamada, N., Nagata, Y., Kobayashi, S., Ono, I.: On the stopping criterion of adaptive weighted aggregation for multiobjective continuous optimization. Trans. Jpn. Soc. Evol. Comput. **4**(1), 13–27 (2013)
7. Hamada, N., Nagata, Y., Kobayashi, S., Ono, I.: BS-AWA: a more scalable adaptive weighted aggregation for continuous multiobjective optimization. Trans. Jpn. Soc. Evol. Comput. **5**(1), 1–15 (2014)
8. Sato, H.: Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs. J. Heuristics **21**(6), 819–849 (2015)
9. Li, K., Zhang, Q., Kwong, S., Li, M., Wang, R.: Stable matching based selection in evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **18**(6), 909–923 (2013)
10. Huband, S., Hingston, P., Barone, L., While, L.: A review of multi-objective test problems and a scalable test problem toolkit. IEEE Trans. Evol. Comput. **10**(5), 477–506 (2006)
11. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, Swiss Federal Institute of Technology, Zurich (1999)