Affinity Based Search Amount Control in Decomposition Based Evolutionary Multi-Objective Optimization

Hiroyuki Sato
The University of
Electro-Communications
1-5-1 Chofugaoka, Chofu,
Tokyo, 182-8585, Japan
h.sato@uec.ac.jp

Minami Miyakawa
Hosei University, JSPS
Research Fellow (PD)
3-7-2 Kajino-cho, Koganei-shi
Tokyo, 184-8584, Japan
miyakawa@cis.k.hosei.ac.jp

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

ABSTRACT

This work proposes a search amount control method on each search part of the Pareto front in decomposition based evolutionary multi-objective optimization. The conventional MOEA/DC decomposes the Pareto front with a set of weight vectors and pairs one solution with each weight vector to approximate the entire Pareto front with the set of solutions. Well-matched pairs of weight vector and solution contribute to uniformly approximating the Pareto front, and mismatched pairs having a long distance between weight vector and solution in the objective space deteriorate the approximation quality and the search. To eliminate mismatched pairs and improve the search performance, this work proposes affinity based search amount control method for MOEA/DC. Experimental results using continuous WFG4 test problems with 2-5 objectives show that the proposed method improves the well-matched pair ratio in all pairs of weight vector and solution and the search performance.

CCS Concepts

•Computing methodologies → Optimization algorithms; •Artificial intelligence → Search methodologies;

Keywords

Multi-objective optimization, evolutionary algorithms, MOEA/D

1. INTRODUCTION

Most of real-world optimization problems are multi-objective optimization problems involving several objectives. In the multi-objective problems, there is no single optimal solution generally since objectives are conflicted. Therefore, the goal of multi-objective problems is to find a set of solutions approximating the Pareto front which is the optimal trade-off among conflicting objectives. The population-based evolutionary algorithms are suited to solve multi-objective problems since a set of solutions to approximate the Pareto front can be obtained from the population in a single run [1, 2].

MOEA/D [3] is one of effective evolutionary algorithms for solving multi-objective problems and an important framework in recent algorithmic studies. MOEA/D uses a set of weight vectors to search the Pareto front. Each weight vector specifies a part of the Pareto front. One solution is assigned to each weight vector. In other words, each weight vector is paired with one solution. Each solution has the role to approximate a part of the Pareto front specified by the paired weight vector. Therefore, it is desirable that generated offspring (solutions) during the search process are paired with their nearest weight vectors in the objective space. MOEA/D determines pairs of weight vector and solution based on the scalarizing function values. However, since MOEA/D predetermines the target solutions to be updated before offspring generations, generated offspring might not be paired with their near weight vectors, and it causes an inefficient search. To overcome this problem, MOEA/DC (MOEA/D with Chain-reaction update) has been proposed [4]. MOEA/DC adaptively determines the target solution order to be updated based on positions of offspring in the objective space after each of offspring generated. Concretely, this method tries to update existing solutions paired with weight vectors near to the offspring in the objective space. As the results, matching relations between weight vector and solution are improved, and the search performance is also improved. The previous study [4] reported that MOEA/DC achieves higher search performance than the conventional MOEA/D and NSGA-III [7] on combinatorial multi-objective knapsack problems [5] having discrete variable spaces and WFG problems [6] having continuous variable spaces.

MOEA/DC is able to improve the matching relations of weight vector and solution, however, mismatched pairs having a long distance between weight vector and solution in the objective space remain. If the mismatched pairs of weight vector and solution can be further eliminated, the approximation performance of the Pareto front is further improved since the solution distribution close to the uniformly distributed weight vectors. To overcome this problem, this work focuses on the number of offspring generations on each weight vector.

In the conventional MOEA/D and MOEA/DC, each weight vector has the equal number of offspring generations. In other words, the rights to generate offspring are uniformly allocated to each weight vector. However, what should be uniform for all weight vectors is the optimality of solutions for each weight vector but the number of offspring generations. For example, if we have an optimal solution well-matched with a weight vector, the necessity to generate new offspring around its well-matched weight vector is lower than mismatched weight vectors. There is possibility that the approximation performance of the Pareto front is improved by allocating more search amount for mismatched weight vectors rather than spending search amount for well-matched weight vectors.

BICT 2017, March 15-16, Hoboken, United States
ISBN 978-1-63190-148-5
DOI: 10.4108/eai.22-3-2017.152400

To improve the search performance of MOEA/D based algorithms, this work focuses on MOEA/DC and proposes a method to control search amount for each weight vectors. To generate a new offspring, we focus on a weight vector. In the cases of the conventional MOEA/D and MOEA/DC, the focused weight vector has the right to generate offspring around it. On the other hand, the proposed method considers to transfer rights to generate offspring of well-matched weight vectors to mismatched weight vectors. Concretely, when mismatched weight vectors exist in the neighboring weight vectors of the focused weight, its right to generate offspring is transferred to the mismatched weight vector. In this way, the proposed method enhances the elimination of mismatched pairs and the improvement of pair matchings of weight vector and solution. The proposed method is combined with MOEA/DC, and its effectiveness is verified on continuous WFG4 test problems with 2-5 objectives.

2. MOEA/DC: AN IMPROVED MOEA/D

2.1 MOEA/D

MOEA/DC (MOEA/D with Chain-reaction update) is designed based on MOEA/D [3]. The basic MOEA/D decomposes a multi-objective optimization problem into a number of single-objective scalarizing function optimization problems using a set of weight vectors $\mathcal{L} = \{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \dots, \boldsymbol{\lambda}^N\}$ and simultaneously optimizes them to approximate the Pareto front. Each weight vector $\boldsymbol{\lambda}^i$ is paired with one solution $\boldsymbol{x}^i$, the set of solutions becomes the population $\mathcal{P} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^N\}$. To generate one offspring (new solution), MOEA/D focuses on a weight vector $\boldsymbol{\lambda}^i$. According to T-neighbor weight vector indices $\mathcal{B}_i = \{i_1, i_2, \dots, i_T\}$ of the focused weight vector $\boldsymbol{\lambda}^i$, MOEA/D randomly selects two parents from the set of solutions $\boldsymbol{x}^{i_1}, \boldsymbol{x}^{i_2}, \dots, \boldsymbol{x}^{i_T}$ and generates an offspring $\boldsymbol{y}$ by applying crossover and mutation operators. Then, MOEA/D tries to replace the existing solutions $\boldsymbol{x}^{i_1}, \boldsymbol{x}^{i_2}, \dots, \boldsymbol{x}^{i_T}$ with the newly generated offspring $\boldsymbol{y}$ based on their scalarizing function values using the focused weight vector $\boldsymbol{\lambda}^i$. MOEA/D searches the entire Pareto front by repeating the above process while sequentially changing the focused weight vector $\boldsymbol{\lambda}^i$ among the set of weight vectors $\mathcal{L} = \{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \dots, \boldsymbol{\lambda}^N\}$.

2.2 MOEA/DC

MOEA/DC (MOEA/D with Chain-reaction update) [4] is a variant of MOEA/D employing an alternative existing solution update mechanism with generated offspring. The original MOEA/D focuses on a weight vector $\boldsymbol{\lambda}^i$, selects parents, generates an offspring and tries to update the existing solution set $\boldsymbol{x}^{i_1}, \boldsymbol{x}^{i_2}, \dots, \boldsymbol{x}^{i_T}$ paired with T-neighbor weight vectors of the focused weight $\boldsymbol{\lambda}^i$. Actually, the appropriate weight vector should be paired with each offspring is unknown before it is generated and evaluated. However, the original MOEA/D predetermines the target existing solutions tried to be updated with each offspring before it is generated and evaluated. It causes inefficient search.

MOEA/DC also randomly selects parents by sequentially focusing on each of weight vectors in the same manner as the original MOEA/D. However, the solution update mechanism of MOEA/DC is different from the original MOEA/D. MOEA/DC adaptively determines the existing solution order tried to be updated with each generated offspring based on its location in the objective space while the original MOEA/D predetermines the existing solutions tried to be updated before each offspring is generated and evaluated. Concretely, for each generated offspring $\boldsymbol{y}$, MOEA/DC calculates the objective balance vector $\boldsymbol{b}(\boldsymbol{y})$ of objective function value vector $\boldsymbol{f}(\boldsymbol{y})$ as follows.

$$b_j(\boldsymbol{y}) = \frac{f_j(\boldsymbol{y}) - z_j}{\sum_{\ell=1}^{m} \{f_\ell(\boldsymbol{y}) - z_\ell\}} \quad (j = 1, 2, \dots, m) \tag{1}$$

where, $\boldsymbol{z}$ is the obtained ideal vector, and each element z_j ($j = 1, 2, \dots, m$) is the best (minimum) objective value obtained during the search for the j-th objective function. Then, MOEA/DC calculates Euclidean distances between $\boldsymbol{b}(\boldsymbol{y})$ and each of all weight vectors $\mathcal{L} = \{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \dots, \boldsymbol{\lambda}^N\}$ and tries to update solutions increasing order of the paired weight distance.

2.3 Problem: Mismatched Pair of Weight Vector and Solution

MOEA/DC tries to update existing solutions by considering not only the scalarizing function value but also matching relation between weight and solution in the objective space. Since MOEA/DC preferentially updates existing solutions paired with close weight vectors to the objective balance vector of newly generated offspring, the decrease of distances between weight vectors and their paired solutions are enhanced compared with the original MOEA/D. MOEA/DC is able to improve the matching relations between weight vector and solution, however, it is difficult to make the all pairs the best matchings. That is, mismatched pairs of weight vector and solution having a long distance in the objective space remain. If the number of mismatched pairs of weight vector and solution could be decreased, the approximation performance of the Pareto front will be improved since the solution distribution approaches uniformly distributed weight distribution.

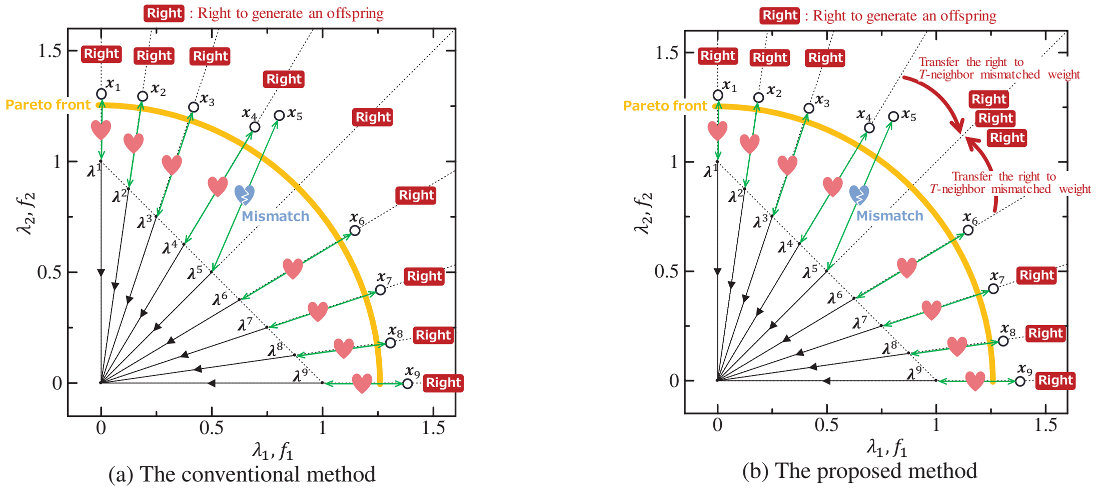
The both MOEA/D and MOEA/DC generate offspring by sequentially focusing on each weight vector to uniformly search each part of the entire Pareto front. Therefore, the equal number of offspring generations is allocated to each weight vector. However, what should be uniform for each weight vector will not be the number of offspring generations but the optimality for the specified part of the entire Pareto front. Generally, the optimal levels of solutions for their paired weight vector are not uniform. Therefore, to uniformly improve the optimal levels of solutions all over the entire Pareto front, allocating more chances to generate offspring for mismatched weight vectors would be effective rather than allocating chances to generate offspring for well-matched pairs.

As a method to dynamically control the number of offspring generations for each weight vector, MOEA/D-DRA (MOEA/D with Dynamical Resource Allocation) using the convergence level of the scalarizing function value has been proposed so far [8]. However, the convergence detection equation and its time window parameter would depend on problems. In this work, we do not use them and design a search amount control method compatible with MOEA/DC.

3. PROPOSAL: AFFINITY BASED SEARCH AMOUNT CONTROL

3.1 Concept

On the basis of MOEA/DC, this work proposes a search amount control method for each weight vector. Concretely, the proposed method finds out mismatched pairs of weight vector and solution, and gives more chance to generate offspring to them with high priority. In this work, pairs of weight and solution can be divided into two types. They are *well-matched pairs* and *mismatched pairs*. If a solution is paired with its nearest weight vector, they are a well-matched pair. If a solution is not paired with its nearest weight vector, they are a mismatched pair. **Fig. 1** shows an example. $\boldsymbol{x}^5$



(a) The conventional method

(b) The proposed method

Figure 1: Rights to generate offspring for well-matched and mismatched pairs of weight vecotor and solution

and $\boldsymbol{\lambda}^5$ are the mismatched pair because the nearest weight vector of $\boldsymbol{x}^5$ is $\boldsymbol{\lambda}^4$ but $\boldsymbol{x}^5$ is paired with 2nd nearest $\boldsymbol{\lambda}^5$. Other pairs of weight vector and solution are the well-matched pairs since all the solutions are paired with their nearest weight vectors. In the conventional MOEA/D and MOEA/DC, as shown in **Fig. 1 (a)**, rights to generate new offspring are evenly assigned to each weight vector even if both well-matched and mismatched weight vectors exist in the set of weigh vector. In the proposed method, although each weight vector has its own right to generate offspring, well-matched weight vectors transfer their right to generate offspring to mismatched weight vectors in T-neighbor weight vectors. **Fig. 1 (b)** shows a conceptual figure of the proposed search amount control with $T = 3$-neighbors. Since $\boldsymbol{\lambda}^5$ is mismatched weight vector, neighbors $\boldsymbol{\lambda}^4$ and $\boldsymbol{\lambda}^6$ transfer their rights to generate offspring to mismatched $\boldsymbol{\lambda}^5$ to find more appropriate $\boldsymbol{x}^5$.

This work proposes two algorithms MOEA/DC-CS and MOEA/DC-CST respectively employing different parent selections. MOEA/DC-CS (MOEA/DC with Controlling Search amount) randomly selects parents from the existing solution set paired with T-neighbor weight vectors of the focused weight vector. On the other hand, MOEA/DC-CST (MOEA/DC with Controlling Search amount and Tournament) selects parents by using g-tournament selection [9] based on scalarizing function value for the focused weight vector.

Algorithm 1 shows the pseudo-code of the proposed MOEA/DC-CS and MOEA/DC-CST. They are based on MOEA/DC [4]. Red lines indicate the codes for proposed search amount control process.

3.2 Controlling Search Amount: Transfer of Offspring Generation Authority

To control the search amount for each weight vector, the proposed method introduces the set of the well-matched pair flags $\mathcal{W}m = \{wm^1, \ldots, wm^N\}$ at 2nd line of **Algorithm 1**. $wm^i = true$ indicates the nearest weight vector of $\boldsymbol{x}^i$ is $\boldsymbol{\lambda}^i$, and they are the well-matched pair. On the other hand, $wm^i = false$ indicates the closest weight vector of $\boldsymbol{x}^i$ is not $\boldsymbol{\lambda}^i$, and they are mismatched pair.

To generate an offspring, we focus on a weight vector $\boldsymbol{\lambda}^i$ (8th line). Then, the proposed method sequentially checks the well-matched pair flags $wm^{i_1}, wm^{i_2}, \ldots, wm^{i_T}$ of T-neighbor weight vectors of the focused weight $\boldsymbol{\lambda}^i$. If a mismatched weight vector (false) is founded, the focused weight to generate an offspring is changed to it (9-14th lines). Thus, the proposed method controls the search amount on each weight vector by transferring rights to generate offspring for mismatched weight vector. Note that, T-neighbor weight vectors $\boldsymbol{\lambda}^{i_1}, \boldsymbol{\lambda}^{i_2}, \ldots, \boldsymbol{\lambda}^{i_T}$ are ordered in ascending order of distance from the focused weight vector $\boldsymbol{\lambda}^i$, and $\boldsymbol{\lambda}^i = \boldsymbol{\lambda}^{i_1}$.

Algorithm 1 The Proposed MOEA/DC-CS and MOEA/DC-CST

```
Input: the number of objectives $m$, the decomposition parameter $H$,
    the number of weight vectors and solutions in the population $N$,
    the neighborhood size $T$
Output: the non-dominated set of solutions
 1: $\mathcal{L} = \{\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^N\} \leftarrow$ Generate weight vectors ($H$, $m$)
 2: $\mathcal{W}m = \{wm^1, \ldots, wm^N\} \leftarrow \{false, false, \ldots, false\}$
 3: for each $\boldsymbol{\lambda}^i \in \mathcal{L}$ do
 4:     $\mathcal{B}_i = \{i_1, \ldots, i_T\} \leftarrow$ Find nearest neighbor weight indices
 5: end for
 6: $\mathcal{P} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\} \leftarrow$ Randomly generate the population
 7: repeat
 8:     for each $i \in \{1, 2, \ldots, N\}$ do
 9:         for each $j \in \{i_1, \ldots, i_T\} = \mathcal{B}_i$ do
10:             if $wm^j$ is $false$ then
11:                 $i \leftarrow j$
12:                 break
13:             end if
14:         end for
15:         $\boldsymbol{x}^p, \boldsymbol{x}^q \leftarrow$ Select parents ($i$)
16:                                     ▷ Algorithm 3 for MOEA/DC-CS
17:                                     ▷ Algorithm 4 for MOEA/DC-CST
18:         $\boldsymbol{y} \leftarrow$ Generate offspring ($\boldsymbol{x}^p, \boldsymbol{x}^q$)
19:         Chain-Reaction Solution Update ($\boldsymbol{y}$, $i$)    ▷ Algorithm 2
20:     end for
21: until The termination criterion is satisfied
22: return The non-dominated solutions picked from $\mathcal{P}$
```

The well-matched pair flags are set in the solution update procedure at 19th line. **Algorithm 2** shows the pseudo-code of the solution update procedure with a newly generated offspring $\boldsymbol{y}$. Red lines indicate the codes for proposed search amount control process. The objective balance vector $\boldsymbol{b}(\boldsymbol{y})$ of the offspring $\boldsymbol{y}$ is cal-

Algorithm 2 Chain-Reaction Solution Update

```
Input: solution $\boldsymbol{y}$
 1: procedure CHAIN-REACTION UPDATE ($\boldsymbol{y}$)
 2:     $\boldsymbol{b}$ ←Calculate balance of objective values ($\boldsymbol{y}$)
 3:     $\mathcal{D} \leftarrow \{d^1, d^2, \dots, d^N\}$     ▷ Distances from all weight vectors
 4:     for each $i \in \{1, 2, \dots, N\}$ do
 5:         $d^i \leftarrow$ Calculate Euclidean distance ($\boldsymbol{b}, \boldsymbol{\lambda}^i$)
 6:     end for
 7:     $\mathcal{D} \leftarrow$ Sort elements in ascending order ($\mathcal{D}$)
 8:     for each $d^j \in \mathcal{D}$ do
 9:         if $g(\boldsymbol{y}|\boldsymbol{\lambda}^j)$ is better than $g(\boldsymbol{x}^j|\boldsymbol{\lambda}^j)$ then
10:             $\boldsymbol{tmp} \leftarrow \boldsymbol{x}^j$                ▷ Preserve temporally
11:             $\boldsymbol{x}^j \leftarrow \boldsymbol{y}$
12:             if $d^j$ is the minimum in $\mathcal{D}$ then
13:                 $wm^j \leftarrow true$
14:             Else
15:                 $wm^j \leftarrow false$
16:             End if
17:             CHAIN-REACTION UPDATE ($\boldsymbol{tmp}$) ▷ Call recursively
18:             break
19:         end if
20:     end for
21: end procedure
```

culated at 2nd line. Next, we calculate distances $d^1, d^2, \dots, d^N$ between the objective balance vector $\boldsymbol{b}(\boldsymbol{y})$ and all weight vectors and sort the distances in ascending order (4-7th lines). Then, we try to update the existing solutions paired with weight vectors in order of sorted weight vector distances by comparing their scalarizing function values. In the case that the offspring $\boldsymbol{y}$ updates an existing solution $\boldsymbol{x}^j$, if $\boldsymbol{\lambda}^j$ is the nearest weight vector to $\boldsymbol{b}(\boldsymbol{y})$ among all weight vectors, $true$ is set to the well-matched pair flag wm^j. If $\boldsymbol{\lambda}^j$ is not the nearest weight vector to $\boldsymbol{b}(\boldsymbol{y})$, $false$ is set to the well-matched pair flag wm^j since $\boldsymbol{y}$ and $\boldsymbol{\lambda}^j$ are mismatched pair (12-16th lines).

3.3 Parent Selection: Two methods

This work combines the proposed search amount control with two kinds of parent selections at 15th line of **Algorithm 1**.

Random Selection (for MOEA/DC-CS)

The random selection [3] is employed in the conventional MOEA/D and MOEA/DC. Its pseudo-code is shown in **Algorithm 3**. In this method, two solutions are randomly selected from solutions paired with T-neighbor weight vectors of the focused weight vector as parents.

g-Tournament Selection (for MOEA/DC-CST)

The g-tournament selection [9] is alternative parent selection which can be used with MOEA/D based algorithms. Its pseudo-code is shown in **Algorithm 4**. This method first randomly selects four candidate solutions from solutions paired with T-neighbor weight vectors of the focused weight vector. For the first two candidate solutions, we compare their scalarizing function values for the focused weight vector and select the winner as the first parent. For the other two candidate solutions, we select the second parent in the same manner. Since appropriate parents for the focused mismatched weight vector can be selected compared with the random selection, it enhances to eliminate mismatched pairs and improves pair relations of weight vector and solution.

Algorithm 3 Random Selection [3] (for MOEA/DC-CS)

```
Input: The focused weight vector index $i$
Output: Two parents $\boldsymbol{x}^p, \boldsymbol{x}^q$
$p, q \leftarrow$ Randomly select two indices ($\mathcal{B}_i$)
return $\boldsymbol{x}^p, \boldsymbol{x}^q$
```

Algorithm 4 g-Tournament Selection [9] (for MOEA/DC-CST)

```
Input: The focused weight vector index $i$
Output: Two parents $\boldsymbol{x}^p, \boldsymbol{x}^q$
$p_1, p_2, q_1, q_2 \leftarrow$ Randomly select four indices ($\mathcal{B}_i$)
if $g(\boldsymbol{x}^{p_1}|\boldsymbol{\lambda}^i)$ is better than $g(\boldsymbol{x}^{p_2}|\boldsymbol{\lambda}^i)$ then
    $p \leftarrow p_1$
else
    $p \leftarrow p_2$
end if
if $g(\boldsymbol{x}^{q_1}|\boldsymbol{\lambda}^i)$ is better than $g(\boldsymbol{x}^{q_2}|\boldsymbol{\lambda}^i)$ then
    $q \leftarrow q_1$
else
    $q \leftarrow q_2$
end if
return $\boldsymbol{x}^p, \boldsymbol{x}^q$
```

4. EXPERIMENTAL SETUP

4.1 Four Algorithms

To verify the effectiveness of the proposed search amount control for each weight vector, in this work, we compare four algorithms summarized in **Table 1**. The basis of the four algorithms is MOEA/DC, and four algorithms are combinations of with/without the proposed search amount control method and the g-tournament selection, respectively. The first one is the conventional MOEA/DC, the second one is MOEA/DC-T employing only the g-tournament selection, the third one is MOEA/DC-CS employing only the proposed search amount control, and the last one is the proposed MOEA/DC-CST employing both the proposed search amount control and the g-tournament selection.

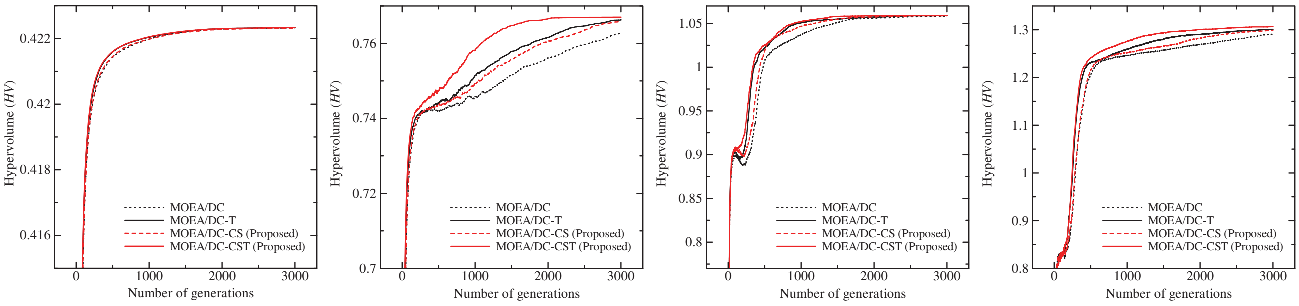
The four algorithms use the same parameters and settings. All algorithms use the reciprocal weighted Tchebycheff scalarizing function [10, 4]. The decomposition parameters are set to $H = \{200, 19, 9, 6\}$ for $m = \{2, 3, 4, 5\}$ objective problems, respectively. Therefore, the population sizes become $N = \{201, 210, 220, 210\}$ for $m = \{2, 3, 4, 5\}$ objective problems, respectively. The neighborhood size is set to $T = 20$. To generate offspring, the SBX crossover [11] with a crossover ratio 0.8 and a distribution index $\eta_c = 20$ and the polynomial mutation with a mutation ratio $1/n$ and the distribution index $\eta_m = 20$ are employed. Also, the termination condition of algorithms is set to totally 3,000 generations.

4.2 Test problems

This work uses WFG4 problem framework [6]. In this problem framework, the difficulty to obtain spread solutions to approximate the entire Pareto front is increased with increasing the position parameter k. In this work, WFG4 problems with $k = \{2, 6, 10\}$ are employed. Also, the number of objectives m is varied from two to five. The distance parameter is set to $l = 10$, and the total number of variables becomes $n = l + k + (m - 1) - k \mod (m - 1)$.

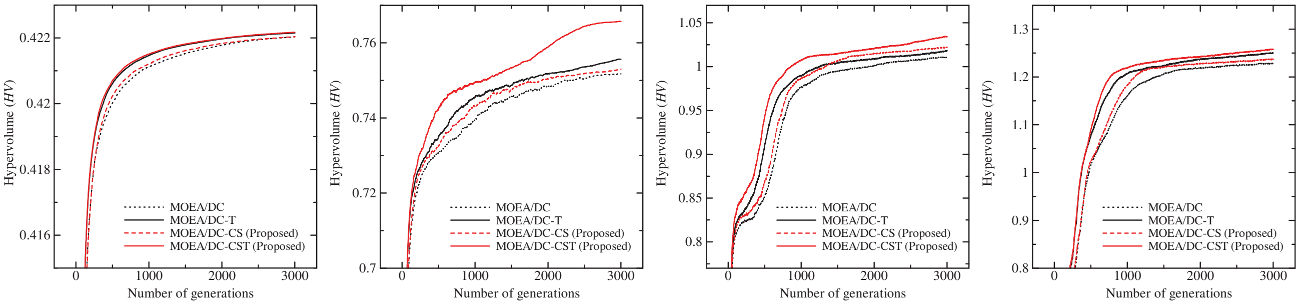
4.3 Performance Metric

To evaluate the obtained solutions approximating the Pareto front, this work uses Hypervolume (HV) [5]. HV is m-dimensional



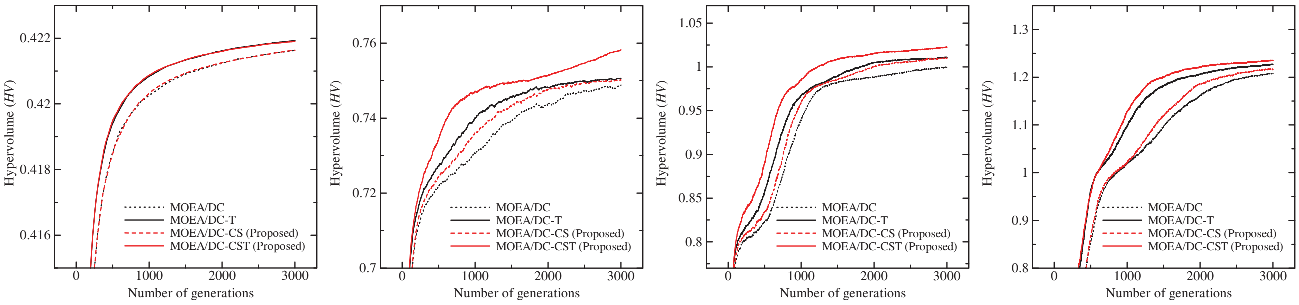
(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 2: Transition of HV on WFG4 with the problem difficulty parameter $k = 2$



(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 3: Transition of HV on WFG4 with the problem difficulty parameter $k = 6$



(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 4: Transition of HV on WFG4 with the problem difficulty parameter $k = 10$

Table 1: Four algorithms

	Controlling Search Amount	g-Tournament Selection
Conventional MOEA/DC	-	-
Conventional MOEA/DC-T	-	✓
Proposed MOEA/DC-CS	✓	-
Proposed MOEA/DC-CST	✓	✓

volume enclosing the obtained solutions and the reference point $\boldsymbol{r}$. HV is increased when the obtained solutions approach to the Pareto front and widely cover the Pareto front. That is, HV involves the contributions of both the convergence of the obtained solutions toward the Pareto front and the diversity of them to widely approximate the Pareto front simultaneously. Therefore, the higher HV, the better approximation performance of the Pareto front.

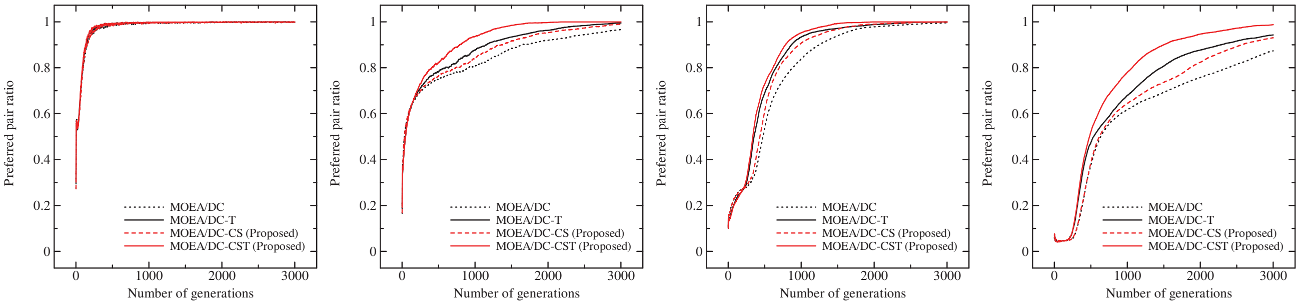
Since each objective function value in the WFG4 problems has different range, we normalize them as $f'_i(\boldsymbol{x}) = f_i(\boldsymbol{x})/2i$ $(i = 1, 2, \dots, m)$ and calculate HV with the normalized objective value vector and the reference point $\boldsymbol{r} = \{1.1, 1.1, \dots, 1.1\}$.

5. RESULTS AND DISCUSSION

5.1 Search Performance

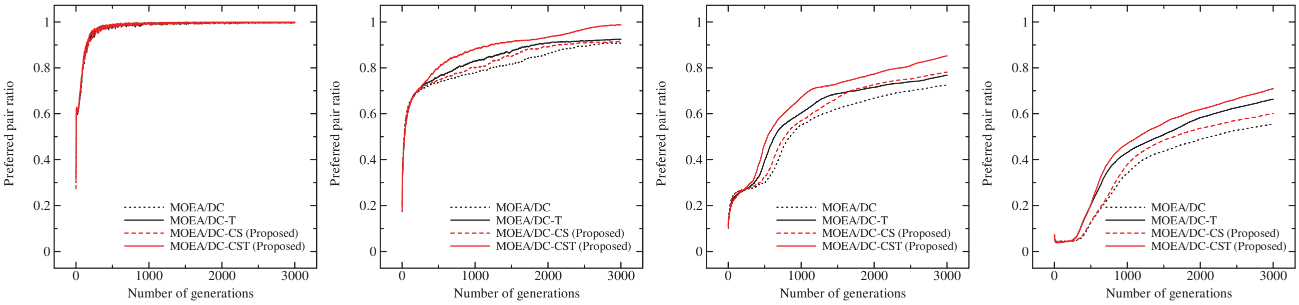
The transitions of average HV values obtained by the four algorithms are shown in **Fig. 2-4**.

From the results on $m = 2$ objective problems, we can see that the proposed MOEA/DC-CS achieves higher HV than the conventional MOEA/DC in the early generations but the difference disappears at the final generation. Also, we can see that the conventional MOEA/DC-T and the proposed MOEA/DC-CST with the



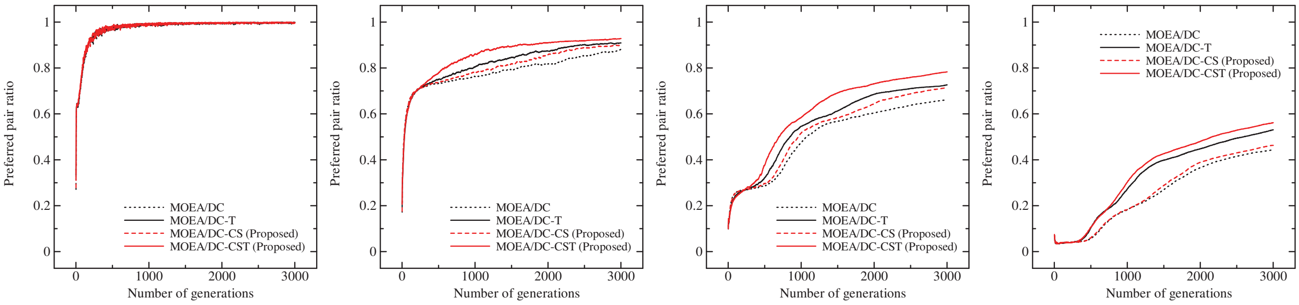
(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 5: Transiton of well-matched pair ratio among all pairs of weight vector and solution on WFG4 with $k = 2$



(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 6: Transiton of well-matched pair ratio among all pairs of weight vector and solution on WFG4 with $k = 6$



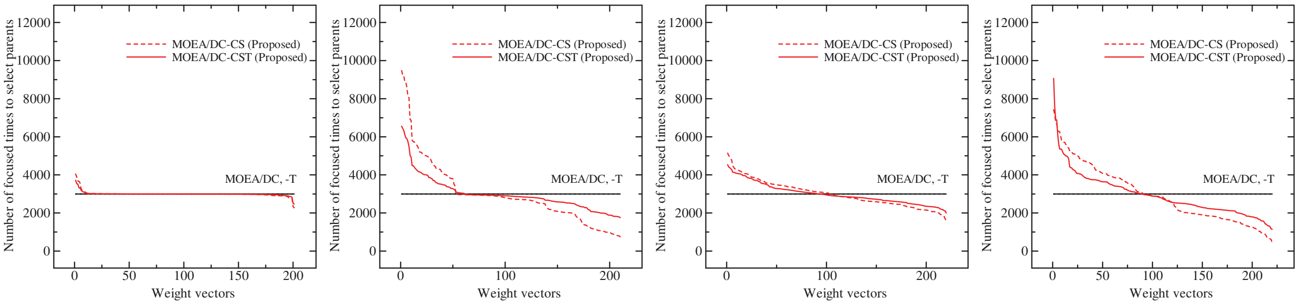
(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 7: Transiton of well-matched pair ratio among all pairs of weight vector and solution on WFG4 with $k = 10$

g-tournament selection achieve higher HV than the conventional MOEA/DC and the proposed MOEA/D-CS with the random selection especially on WFG4 problems with $k = \{6, 10\}$. Thus, in $m = 2$ objective problems, although the effectiveness of the g-tournament selection can be observed, the effectiveness of the proposed search amount control cannot be observed enough.

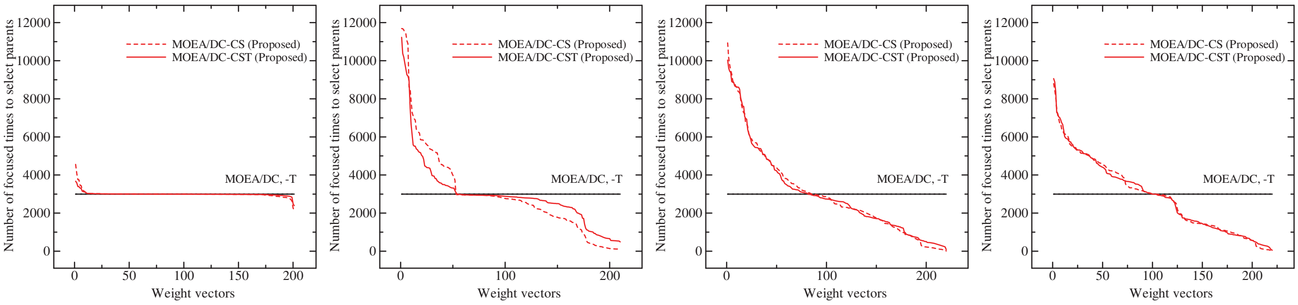
From the results on $m = 3$ objective problems, differences of HV among the four algorithms become significant. In the cases with the random selection, we can see that the proposed MOEA/DC-CS achieves higher HV than MOEA/DC. This result reveals that the proposed search amount control contributes to improving the search performance by transferring rights to generate offspring for mismatched weight vectors. Also, we can see that MOEA/DC-T and MOEA/DC-CST with the g-tournament selection show better HV than the MOEA/DC and MOEA/DC-CS with the random selection. Furthermore, we can see that the proposed MOEA/DC-CST shows higher HV than the conventional MOEA/DC-T. This result reveals that the effectiveness of the proposed search amount control giving more chance to generate offspring for mismatched weights is further enhanced by selecting appropriate parents for mismatched weight vectors based on the g-tournament selection. As the general tendency, the contribution of the g-tournament selection (T) for HV is higher than the one of the proposed controlling search amount control (CS). However, note that the highest HV is achieved by MOEA/DC-CST using both of them. That is, the combination of the proposed search amount control and the g-tournament selection contributes to improving the search performance.

Finally, from the results on problems with $\{4, 5\}$ objectives, we can see that the proposed MOEA/DC-CST achieves the highest HV among the four MOEA/DC based algorithms. This is the similar tendency observed on problems with $m = 3$ objectives.



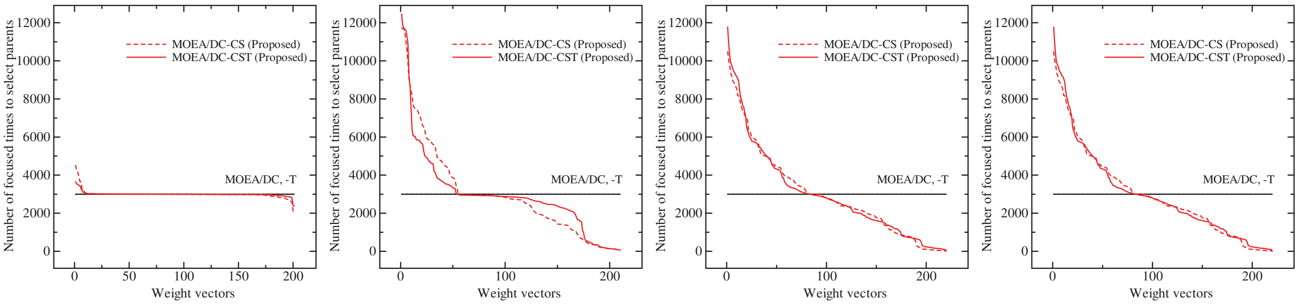
(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 8: The total number of focuses on each weight vector for offspring generations on WFG4 with $k = 2$



(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 9: The total number of focuses on each weight vector for offspring generations on WFG4 with $k = 6$



(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives (d) $m = 5$ objectives

Figure 10: The total number of focuses on each weight vector for offspring generations on WFG4 with $k = 10$

In the problem with $m = 4$ objectives and $k = 6$, the proposed MOEA/DC-CS shows better HV than MOEA/DC-T. In this case, the contribution of the proposed search amount control becomes higher than the one of the g-tournament selection. These results reveal that the proposed search amount control combined with the g-tournament selection improves the search performance.

5.2 Well-Matched Pair Ratio

Fig. 5-**7** show transitions of well-matched pair ratio of weight vector and solution. MOEA/D has N pairs of weight vector and solution. If a solution is paired with its nearest weight vector, the pair is a well-matched one. If a solution is not paired with its nearest weight vector, the pair is a mismatched one. Therefore, the ideal well-matched pair ratio is 1.0. That means all solutions in the population are paired with their nearest weight vectors, and there is no mismatched pair.

From the results on $m = 2$ objectives, the well-matched pair ratios of the four algorithms close to 1.0 on problems with any k after about 500 generations, and almost all solutions are paired with their nearest weight vectors. Next, from the results on problems with $k = 2$ shown in **Fig. 5 (a)**-**(d)**, we can see a tendency that the well-matched pair ratio is decreased with increasing the number of objectives m. The lowest well-matched pair ratio is observed by the conventional MOEA/DC, and the highest well-matched pair ratio is observed by the proposed MOEA/DC-CST. Also, we can see that each of the proposed search amount control and the g-tournament selection contributes to improving the well-matched pair ratio among all pairs of weight vector and solution. The well-matched pair ratios of the proposed MOEA/DC-CST close to 1.0 at the final generation, and almost all solutions are paired with their nearest weight vectors.

Next, from the **Fig. 6** and **7**, we can see that the well-matched pair ratios are decreased with increasing the problem parameter k. Also, in these cases, we can see that the proposed MOEA/DC-CST achieves the highest well-matched pair ratio among the four MOEA/DC based algorithms.

5.3 Number of Offspring Generation on Each Weight Vector

To observe the number of offspring generations on each weight vector, **Fig. 8**-**10** show the total number of focused times on each weight vector in offspring generation process. In these figure, the horizontal axis indicates weight vector numbers, and the vertical axis indicates the total number of focused times on each weight vector. To facilitate visualization, weight vectors on the horizontal axis are sorted in descending order of the total number of focused times.

From the results, first we can see that the total number of focused times on all weight vectors of the conventional MOEA/DC and MOEA/DC-T without the proposed controlling search amount method are the same, and it is 3,000 times which is equivalent to the total number of generations. On the other hand, the total number of focused times on each weight vector of the proposed MOEA/DC-CS and MOEA/DC-CST with the proposed controlling search amount method are different. In the case of $m = 2$ objective problems, the bias of the total number of focuses on each weight vector is relatively small. However, in the cases of $m = \{3, 4, 5\}$ objective problems, the bias becomes large. It will contribute to improving both the pair matchings and the search performance. Also, we can see the tendency that MOEA/DC-CST shows lower bias of focused times on each weight vector than MOEA/DC-CS. That is, the g-tournament selection has effects to generate good solutions for mismatched weight vectors and decrease the number of mismatched pairs among all pairs of weight vector and solution.

6. CONCLUSIONS

To improve the search performance of MOEA/D based algorithms, this work focused on MOEA/DC and proposed the controlling search amount method finding out mismatched pairs of weight vector and solution and transferring rights to generate offspring from well-matched weight vectors to mismatched ones. To verify the effectiveness of the proposed method, we used WFG4 problems. As the results, we showed that the proposed controlling search amount improved the search performance of MOEA/DC especially on problems with more than two objectives. Also, we showed that the proposed method combined with the g-tournament selecting appropriate parents for mismatched weight vectors further improved the well-matched pair ratio and the search performance.

As future works, we will verify the effectiveness of the proposed method on problems with many-objectives and discrete variable space. Also, although this work combined the four MOEA/DC based algorithms with the reciprocal weighted Tchebycheff scalarizing function, we had other options in scalarizing functions. Therefore, we will verify the search performance of the proposed method with other scalarizing functions.

7. REFERENCES

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

[2] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Boston, Kluwer Academic Publishers, 2002.

[3] Q. Zhang, H. Li, “MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. on Evolutionary Computation*, Vol. 11, No. 6, pp. 712–731, 2007.

[4] H. Sato, “Chain-Reaction Solution Update in MOEA/D and Its Effects on Multi and Many-Objective Optimization," Soft Computing, Springer, Volume 20, Issue 10, pp. 3803–3820, 2016.

[5] E. Zitzler, L. Thiele, “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Trans. on Evolutionary Computation*, Vol. 3 (4), pp. 257–271, 1999.

[6] S. Huband, P. Hingston, L. Barone, and L. While, “A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No 5, pp. 477–506, 2006.

[7] K. Deb, H. Jain, “An Evolutionary Many-objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints," *IEEE Trans. on Evolutionary Computation*, Volume:PP, Issue:99, pp.1–23, 2013.

[8] Q. Zhang, W. Liu, and H. Li, , “The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances," *Proc. of 2009 IEEE Congress on Evolutionary Computation (CEC’2009)*, pp. 203–208, 2009.

[9] H. Sato, “MOEA/D Using Constant-Distance Based Neighbors Designed for Many-Objective Optimization," *Proc. of 2015 IEEE Congress on Evolutionary Computation (CEC2015)*, pp. 2867-2874, 2015.

[10] K. Li, Q. Zhang, S. Kwong, M. Li, R. Wang, “Stable Matching Based Selection in Evolutionary Multiobjective Optimization," *IEEE Trans. on Evolutionary Computation*, Vol. PP, Issue 99, pp.1-15, 2013.

[11] K. Deb and M. Goyal, “A Combined Genetic Adaptive Search (GeneAS) for Engineering Design," *Computer Science and Informatics*, 26 (4), pp. 30–45, 1996.