A Decomposition-based Evolutionary Algorithm for Multi-modal Multi-objective Optimization

Ryoji Tanabe and Hisao Ishibuchi

Shenzhen Key Laboratory of Computational Intelligence, Department of Computer Science and Engineering, Southern University of Science and Technology, China
rt.ryoji.tanabe@gmail.com, hisao@sustc.edu.cn

Abstract. This paper proposes a novel decomposition-based evolutionary algorithm for multi-modal multi-objective optimization, which is the problem of locating as many as possible (almost) equivalent Pareto optimal solutions. In the proposed method, two or more individuals can be assigned to each decomposed subproblem to maintain the diversity of the population in the solution space. More precisely, a child is assigned to a subproblem whose weight vector is closest to its objective vector, in terms of perpendicular distance. If the child is close to one of individuals that have already been assigned to the subproblem in the solution space, the replacement selection is performed based on their scalarizing function values. Otherwise, the child is newly assigned to the subproblem, regardless of its quality. The effectiveness of the proposed method is evaluated on seven problems. Results show that the proposed algorithm is capable of finding multiple equivalent Pareto optimal solutions.

1 Introduction

A multi-objective optimization problem (MOP) is the problem of finding a solution $\boldsymbol{x} = (x_1, ..., x_D)^{\mathrm{T}} \in \mathbb{S}$ that minimizes an objective function vector $\boldsymbol{f} : \mathbb{S} \rightarrow \mathbb{R}^M$. Here, $\mathbb{S}$ is the D-dimensional solution space, and $\mathbb{R}^M$ is the M-dimensional objective space. Usually, $\boldsymbol{f}$ consists of M conflicting objective functions. A solution $\boldsymbol{x}^1$ is said to dominate $\boldsymbol{x}^2$ iff $f_i(\boldsymbol{x}^1) \leq f_i(\boldsymbol{x}^2)$ for all $i \in \{1, ..., M\}$ and $f_i(\boldsymbol{x}^1) < f_i(\boldsymbol{x}^2)$ for at least one index i. If there exists no $\boldsymbol{x}$ in $\mathbb{S}$ such that $\boldsymbol{x}$ dominates $\boldsymbol{x}^*$, $\boldsymbol{x}^*$ is called a Pareto optimal solution. The set of all $\boldsymbol{x}^*$ is the Pareto optimal solution set, and the set of all $\boldsymbol{f}(\boldsymbol{x}^*)$ is the Pareto front. The goal of MOPs is usually to find a set of nondominated solutions that approximates the Pareto front well in the objective space.

An evolutionary multi-objective optimization algorithm (EMOA) is an efficient population-based optimization method to approximate the Pareto front of a given MOP in a single run [1]. Although several paradigms of EMOAs (e.g., dominance-based EMOAs) have been proposed, decomposition-based EMOAs are recently popular in the EMO community. In particular, MOEA/D [18] is one of the most representative decomposition-based EMOAs [14].

There are multiple equivalent Pareto optimal solutions in some real-world problems (e.g., space mission design problems [12], rocket engine design problems [7], and path-planning problems [17]). Fig. 1 explains such a situation.

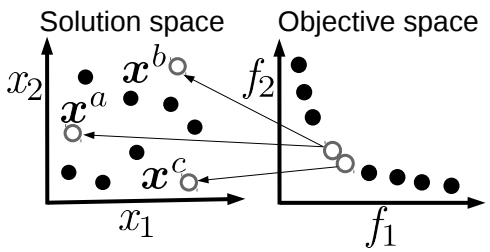

Fig. 1: Illustration of a situation where three solutions are identical or close to each other in the objective space but are far from each other in the solution space. This figure was made using [8,13] as reference.

Diverse solutions are helpful for decision-making [3,11,12,13]. If two or more solutions having (almost) the same objective vector are found, users can make a final decision according to their preference which cannot be represented by the objective functions. For example, in Fig. 1, if $\boldsymbol{x}^a$ becomes unavailable for some reasons (e.g., materials shortages and traffic accidents), $\boldsymbol{x}^b$ and $\boldsymbol{x}^c$ can be candidates for the final solution instead of $\boldsymbol{x}^a$.

A multi-modal MOP (MMOP) [3,8,17] is the problem of locating as many as possible (almost) equivalent Pareto optimal solutions. Unlike the general MOPs, the goal of MMOPs is to find a good approximation of the Pareto-optimal solution set. For example, in Fig. 1, it is sufficient to find one of $\boldsymbol{x}^a$, $\boldsymbol{x}^b$, and $\boldsymbol{x}^c$ for MOPs, because their objective vectors are almost the same. In contrast, EMOAs need to find all of $\boldsymbol{x}^a$, $\boldsymbol{x}^b$, and $\boldsymbol{x}^c$ for MMOPs. Some EMOAs for MMOPs have been proposed in the literature (e.g., [3,6,12,13,17]).

While MMOPs can be found in real-world problems [7,12,17], it is likely that most MOEA/D-type algorithms [14] are not capable of locating multiple equivalent Pareto optimal solutions. This is because they do not have any explicit mechanism to maintain the diversity of the population in the solution space. If the ability to keep solution space diversity in the population is incorporated into MOEA/D, an efficient multi-modal multi-objective optimizer may be realized.

This paper proposes a novel MOEA/D algorithm with addition and deletion operators (MOEA/D-AD) for multi-modal multi-objective optimization. In MOEA/D-AD, the population size μ is dynamically changed during the search process. Multiple individuals that are far from each other in the solution space can be assigned to the same decomposed single-objective subproblem. Only similar individuals in the solution space are compared based on their scalarizing function values. Thus, MOEA/D-AD maintains the diversity in the population by performing environmental selection for each subproblem among individuals that are close to each other in the solution space.

This paper is organized as follows. Section 2 introduces MOEA/D-AD. Section 3 describes experimental setup. Section 4 presents experimental results of MOEA/D-AD, including performance comparison and its analysis. Section 5 concludes this paper with discussions on directions for future work.

Algorithm 1: The procedure of MOEA/D-AD

1 $t \leftarrow 1$, initialize the population $\boldsymbol{P} = \{\boldsymbol{x}^1, ..., \boldsymbol{x}^N\}$;
2 **for** $i \in \{1, ..., N\}$ **do** Assign $\boldsymbol{x}^i$ to the i-th subproblem;
3 **while** The termination criteria are not met **do**
4 $\mu \leftarrow |\boldsymbol{P}|$;
5 Randomly select r_1 and r_2 from $\{1, ..., \mu\}$ such that $r_1 \neq r_2$;
6 Generate the child $\boldsymbol{u}$ by recombining $\boldsymbol{x}^{r_1}$ and $\boldsymbol{x}^{r_2}$;
7 Apply the mutation operator to $\boldsymbol{u}$;
8 **for** $i \in \{1, ..., N\}$ **do** $d_i \leftarrow \mathrm{PD}(\boldsymbol{f}'(\boldsymbol{u}), \boldsymbol{w}^i)$;
9 $j \leftarrow \underset{i \in \{1,...,N\}}{\arg\min} \{d_i\}$;
10 $b^{\mathrm{winner}} \leftarrow$ FALSE and $b^{\mathrm{explorer}} \leftarrow$ TRUE;
11 **for** $\boldsymbol{x} \in \boldsymbol{P} \mid \boldsymbol{x}$ has been assigned to the j-th subproblem **do**
12 **if** isNeighborhood$(\boldsymbol{u}, \boldsymbol{x})$ = TRUE **then**
13 $b^{\mathrm{explorer}} \leftarrow$ FALSE;
14 **if** $g(\boldsymbol{u}|\boldsymbol{w}^j) \leq g(\boldsymbol{x}|\boldsymbol{w}^j)$ **then**
15 $\boldsymbol{P} \leftarrow \boldsymbol{P} \backslash \{\boldsymbol{x}\}$ and $b^{\mathrm{winner}} \leftarrow$ TRUE;
16 **if** b^{winner} = TRUE or b^{explorer} = TRUE **then**
17 $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup \{\boldsymbol{u}\}$ and assign $\boldsymbol{u}$ to the j-th subproblem;
18 $t \leftarrow t + 1$;
19 $\boldsymbol{A} \leftarrow$ selectSparseSolutions$(\boldsymbol{P})$;
20 **return** $\boldsymbol{A}$;

2 Proposed MOEA/D-AD

MOEA/D decomposes a given M-objective MOP into N single-objective subproblems using a scalarizing function $g : \mathbb{R}^M \rightarrow \mathbb{R}$ and a set of uniformly distributed weight vectors $\boldsymbol{W} = \{\boldsymbol{w}^1, ..., \boldsymbol{w}^N\}$, where $\boldsymbol{w}^i = (w_1^i, ..., w_M^i)^{\mathrm{T}}$ for each $i \in \{1, ..., N\}$, and $\sum_{j=1}^{M} w_j^i = 1$. One individual in the population $\boldsymbol{P}$ is assigned to each decomposed subproblem. Thus, the population size μ of MOEA/D always equals N (i.e., $\mu = N$ and $|\boldsymbol{P}| = |\boldsymbol{W}|$).

Algorithm 1 shows the procedure of the proposed MOEA/D-AD for multimodal multi-objective optimization. While the number of subproblems N is still constant in MOEA/D-AD, μ is nonconstant. Although it is ensured that $\mu \geq N$, μ is dynamically changed during the search process (i.e., $\mu \neq N$ and $|\boldsymbol{P}| \neq |\boldsymbol{W}|$) unlike MOEA/D. After the initialization of the population (lines 1–2), the following steps are repeatedly applied until a termination condition is satisfied.

At the beginning of each iteration, parent individuals are selected from the whole population $\boldsymbol{P}$ (line 5). Unlike the original MOEA/D, the mating selection is not restricted to neighborhood individuals to generate diverse new solutions. Then, a child $\boldsymbol{u}$ is generated by applying the variation operators (lines 6–7).

After $\boldsymbol{u}$ has been generated, the environmental selection is performed (lines 8–17). Note that our subproblem selection method (described below) was derived

Algorithm 2: The isNeighborhood($\boldsymbol{u}, \boldsymbol{x}$) function

```
   /* The population P = {y^1, ..., y^μ}                                  */
1  for i ∈ {1, ..., μ} do d_i^E ← NED(y^i, u);
2  Sort individuals based on their distance values such that d_1^E ≤ d_2^E ≤ ... ≤ d_μ^E;
3  for i ∈ {1, ..., L} do
4      if y^i = x then return TRUE;
5  return FALSE;
```

from MOEA/D-DU [16]. However, unlike MOEA/D-DU, only one subproblem is updated for each iteration in MOEA/D-AD to preserve the diversity. First, the perpendicular distance d_i between the normalized objective vector $\boldsymbol{f}'(\boldsymbol{u})$ of $\boldsymbol{u}$ and $\boldsymbol{w}^i$ is calculated for each $i \in \{1, ..., N\}$ (line 8), where PD represents the perpendicular distance between two input vectors. Here, $\boldsymbol{f}'(\boldsymbol{u})$ is obtained as follows: $f'_k(\boldsymbol{u}) = (f_k(\boldsymbol{u}) - f_k^{\min})/(f_k^{\max} - f_k^{\min})$, where $f_k^{\min} = \min_{\boldsymbol{y} \in \boldsymbol{P}}\{f_k(\boldsymbol{y})\}$, and $f_k^{\max} = \max_{\boldsymbol{y} \in \boldsymbol{P}}\{f_k(\boldsymbol{y})\}$ for each $k \in \{1, ..., M\}$. Then, the j-th subproblem having the minimum d value is selected (line 9). The environmental selection is performed only on the j-th subproblem.

The child $\boldsymbol{u}$ is compared to all the individuals that have been assigned to the j-th subproblem (line 11–15). Two Boolean variables b^{winner} and $b^{\text{explorer}} \in \{\text{TRUE}, \text{FALSE}\}$ (line 10) are used for the addition operation of MOEA/D-AD. More precisely, b^{winner} represents whether $\boldsymbol{u}$ outperforms at least one individual belonging to the j-th subproblem regarding the scalarizing function value, and b^{explorer} indicates whether $\boldsymbol{u}$ is far from all the individuals assigned to the j-th subproblem in the solution space. If at least one of b^{winner} and b^{explorer} is TRUE, $\boldsymbol{u}$ enters the population $\boldsymbol{P}$ (lines 16–17).

In line 12 of Algorithm 1, the isNeighborhood($\boldsymbol{u}, \boldsymbol{x}$) function returns TRUE if $\boldsymbol{u}$ is close to $\boldsymbol{x}$ in the solution space (otherwise, it returns FALSE). Algorithm 2 shows details of the function, where the NED function returns the normalized Euclidean distance between two input vectors using the upper and lower bounds for each variable of a given problem. In Algorithm 2, L ($1 \leq L \leq \mu$) is a control parameter of MOEA/D-AD. First, the normalized Euclidean distance between each individual in $\boldsymbol{P}$ and $\boldsymbol{u}$ is calculated. Then, all the μ individuals are sorted based on their distance values in descending order. Finally, if $\boldsymbol{x}$ is within the L nearest individuals from $\boldsymbol{u}$ among the μ individuals, the function returns TRUE.

If $\boldsymbol{x}$ is in the neighborhood of $\boldsymbol{u}$ in the solution space (line 12), they are compared based on their scalarizing function values (lines 14–15). The environmental selection is performed only among similar individuals in the solution space in order to maintain the diversity of the population. If $\boldsymbol{x}$ is worse than $\boldsymbol{u}$, $\boldsymbol{x}$ is removed from $\boldsymbol{P}$ (line 15). This is the deletion operation of MOEA/D-AD.

Since μ is not bounded, $\boldsymbol{P}$ may include a large number of solutions at the end of the search. This is undesirable in practice because decision-makers are likely to want to examine only a small number of nondominated solutions that approximate the Pareto front and the Pareto solution set [18]. To address this

Algorithm 3: The selectSparseSolutions($\boldsymbol{P}$) function

```
   /* The population P = {x^1, ..., x^μ}                                  */
 1 P ← selectNondominatedSolutions(P), μ ← |P|, A ← ∅;
 2 for i ∈ {1, ..., μ} do b_i^selected ← FALSE, D_i ← ∞;
 3 Randomly select j from {1, ..., μ};
 4 A ← A ∪ {x^j}, b_j^selected ← TRUE;
 5 for i ∈ {1, ..., μ} do
 6     if b_i^selected = FALSE then D_i ← min(NED(x^i, x^j), D_i);
 7 while |A| < N do
 8     j ← arg max_{i∈{1,...,μ} | b_i^selected = FALSE} D_i, A ← A ∪ {x^j}, b_j^selected ← TRUE;
 9     for i ∈ {1, ..., μ} do
10         if b_i^selected = FALSE then D_i ← min(NED(x^i, x^j), D_i);
11 return A, which is the N or fewer nondominated solutions selected from P;
```

issue, a method of selecting N nondominated solutions is applied to the final population (line 19). Recall that N denotes the number of subproblems.

Algorithm 3 shows details of the selectSparseSolutions function, which returns N or less nondominated solutions $\boldsymbol{A}$. First, nondominated solutions are selected from $\boldsymbol{P}$. Then, one individual is randomly selected from $\boldsymbol{P}$ and inserted into $\boldsymbol{A}$. Then, a solution having the maximum distance to solutions in $\boldsymbol{A}$ is repeatedly stored into $\boldsymbol{A}$. It is expected that a set of nondominated solutions being far from each other in the solution space are obtained by this procedure.

3 Experimental settings

Test problems. We used the following seven two-objective MMOPs: the Two-On-One problem [10], the Omni-test problem [3], the three SYM-PART problems [11], and the two SSUF problems [9]. The number of variables D is five for the Omni-test problem and two for the other problems. In the Two-On-One and SSUF1 problems, there are two symmetrical Pareto optimal solutions that are mapped to the same objective vector. In the other problems, Pareto optimal solutions are regularly distributed. The number of equivalent Pareto optimal solutions is two for the SSUF3 problem, nine for the three SYM-PART problems, and 45 for the Omni-test problem.

Performance indicators. We used the inverted generational distance (IGD) [20] and IGDX [19] for performance assessment of EMOAs. Below, $\boldsymbol{A}$ denotes a set of nondominated solutions of the final population of an EMOA. The IGD and IGDX metrics require a set of reference points $\boldsymbol{A}^*$. For $\boldsymbol{A}^*$ for each problem, we used 5 000 solutions which were selected from randomly generated 10 000 Pareto-optimal solutions by using the selectSparseSolutions function (Algorithm 3).

The IGD value is the average distance from each reference solution in $\boldsymbol{A}^*$ to its nearest solution in $\boldsymbol{A}$ in the objective space as follows:

$$\mathrm{IGD}(\boldsymbol{A}) = \frac{1}{|\boldsymbol{A}^*|}\left(\sum_{\boldsymbol{z}\in\boldsymbol{A}^*}\min_{\boldsymbol{x}\in\boldsymbol{A}}\left\{\mathrm{ED}\big(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{f}(\boldsymbol{z})\big)\right\}\right),$$

where $\mathrm{ED}(\boldsymbol{x}^1,\boldsymbol{x}^2)$ represents the Euclidean distance between $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$.

Similarly, the IGDX value of $\boldsymbol{A}$ is given as follows:

$$\mathrm{IGDX}(\boldsymbol{A}) = \frac{1}{|\boldsymbol{A}^*|}\left(\sum_{\boldsymbol{z}\in\boldsymbol{A}^*}\min_{\boldsymbol{x}\in\boldsymbol{A}}\left\{\mathrm{ED}\big(\boldsymbol{x},\boldsymbol{z}\big)\right\}\right).$$

While IGD measures the quality of $\boldsymbol{A}$ in terms of both convergence to the Pareto front and diversity in the objective space, IGDX evaluates how well $\boldsymbol{A}$ approximates the Pareto-optimal solution set in the solution space. Thus, EMOAs that can find $\boldsymbol{A}$ with small IGD and IGDX values are efficient multi-objective optimizers and multi-modal multi-objective optimizers, respectively. It should be noted that small IGD values do not always mean small IGDX values.

Setup for EMOAs. We compared MOEA/D-AD with the following five methods: MO_Ring_PSO_SCD [17], Omni-optimizer [3], NSGA-II [2], MOEA/D [18], and MOEA/D-DU [16]. Omni-optimizer is a representative EMOA for MMOPs. MO_Ring_PSO_SCD is a recently proposed PSO algorithm for MMOPs. NSGA-II and MOEA/D are widely used EMOAs for MOPs. Since the selection method of the subproblem to be updated in MOEA/D-AD was derived from MOEA/D-DU, we also included it in our experiments.

Available source code through the Internet were used for algorithm implementation. For the implementation of MOEA/D-AD, we used the jMetal framework [4]. Source code of MOEA/D-AD can be downloaded from the first author's website (`https://ryojitanabe.github.io/`). The population size μ and the number of weight vectors N were set to 100 for all the methods. In MOEA/D-AD, μ is dynamically changed as shown in Fig. 4(a). For a fair comparison, we used a set of nondominated solutions of the size $N = 100$ selected from the final population by using the selectSparseSolutions function (Algorithm 3). Thus, the EMOAs were compared using the obtained solution sets of the same size (100). For all the six EMOAs, the number of maximum function evaluations was set to 30 000, and 31 runs were performed. The SBX crossover and the polynomial mutation were used in all the EMOAs (except for MO_Ring_PSO_SCD). Their control parameters were set as follows: $p_c = 1$, $\eta_c = 20$, $p_m = 1/D$, and $\eta_m = 20$.

We used the Tchebycheff function [18] for MOEA/D and MOEA/D-AD as the scalarizing function. The control parameter L of MOEA/D-AD was set to $L = \lfloor 0.1\mu \rfloor$ (e.g., $L = 201$ when $\mu = 2\,018$). According to [16,18], the neighborhood size T of MOEA/D and MOEA/D-DU was set to $T = 20$. All other parameters of MOEA/D-DU and MO_Ring_PSO_SCD were set according to [16,17].

Table 1: Results of the six EMOAs on the seven MMOPs. The tables (a) and (b) show the mean IGD and IGDX values, respectively. The best and second best data are represented by the bold and italic font. The numbers in parenthesis indicate the ranks of the EMOAs. The symbols +, −, and ≈ indicate that a given EMOA performs significantly better (+), significantly worse (−), and not significantly better or worse (≈) compared to MOEA/D-AD according to the Wilcoxon rank-sum test with $p < 0.05$.

(a) IGD

	MOEA/D-AD	MO_Ring_PSO_SCD	Omni-optimizer	NSGA-II	MOEA/D	MOEA/D-DU
Two-On-One	0.0637 (5)	0.0606≈ (4)	*0.0489*+ (2)	0.0490+ (3)	**0.0450**+ (1)	0.0709− (6)
Omni-test	0.0755 (5)	0.1814− (6)	*0.0303*+ (2)	**0.0297**+ (1)	0.0517+ (4)	0.0458+ (3)
SYM-PART1	0.0302 (4)	0.0283+ (3)	*0.0236*+ (2)	**0.0210**+ (1)	0.0467− (5)	0.0478− (6)
SYM-PART2	0.0305 (3)	0.0312≈ (4)	*0.0284*+ (2)	**0.0229**+ (1)	0.0466− (5)	0.0474− (6)
SYM-PART3	*0.0307* (2)	0.0323− (3)	0.0343− (4)	**0.0228**+ (1)	0.0455− (5)	0.0470− (6)
SSUF1	0.0075 (6)	0.0065+ (5)	0.0060+ (4)	*0.0055*+ (2)	0.0055+ (3)	**0.0042**+ (1)
SSUF3	0.0190 (5)	0.0106+ (3)	0.0170+ (4)	**0.0073**+ (1)	0.0629− (6)	*0.0082*+ (2)

(b) IGDX

	MOEA/D-AD	MO_Ring_PSO_SCD	Omni-optimizer	NSGA-II	MOEA/D	MOEA/D-DU
Two-On-One	**0.0353** (1)	*0.0369*− (2)	0.0383− (3)	0.1480− (4)	0.2805− (6)	0.2067− (5)
Omni-test	**1.3894** (1)	2.2227− (3)	*2.0337*− (2)	2.5664− (4)	4.3950− (6)	2.9251− (5)
SYM-PART1	**0.0686** (1)	*0.1482*− (2)	3.8027− (3)	7.9287− (5)	9.1551− (6)	5.0426− (4)
SYM-PART2	**0.0783** (1)	*0.1610*− (2)	1.0863− (3)	5.3711− (5)	9.4834− (6)	5.1610− (4)
SYM-PART3	**0.1480** (1)	*0.4909*− (2)	1.3620− (3)	5.8410− (5)	7.3969− (6)	4.6767− (4)
SSUF1	**0.0761** (1)	*0.0860*− (2)	0.0899− (3)	0.1323− (5)	0.2443− (6)	0.1143− (4)
SSUF3	*0.0302* (2)	**0.0198**+ (1)	0.0541− (3)	0.0710− (5)	0.3083− (6)	0.0599− (4)

4 Experimental results

4.1 Performance comparison

IGD metric. Table 1 shows the comparison of the EMOAs on the seven problems. The IGD and IGDX values are reported in Tables 1(a) and (b), respectively.

Table 1(a) shows that the performance of NSGA-II regarding the IGD metric is the best on five problems. MOEA/D and MOEA/D-DU also perform best on the Two-On-One and SSUF1 problems, respectively. In contrast, MOEA/D-AD and MO_Ring_PSO_SCD perform poorly on most problems. Note that such a poor performance of multi-modal multi-objective optimizers for multi-objective optimization has already been reported in [13,17]. Since multi-modal multi-objective optimizers try to locate all equivalent Pareto optimal solutions, their ability to find a good approximation of the Pareto front is usually worse than that of multi-objective optimizers, which directly approximate the Pareto front.

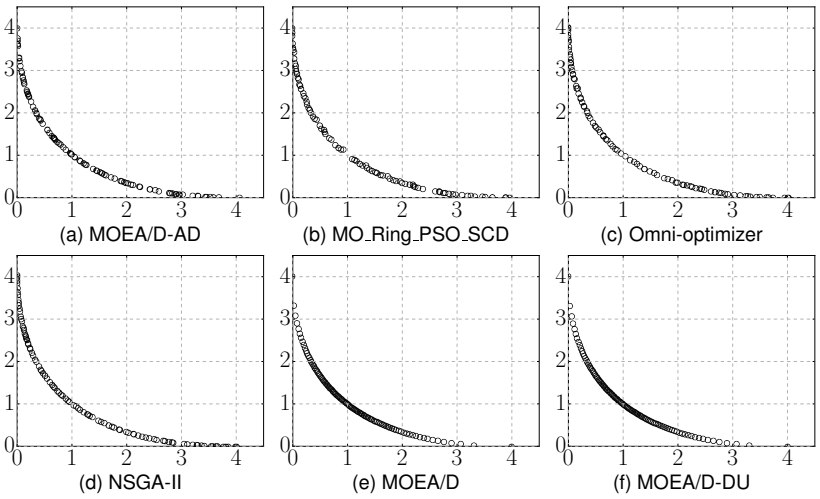

Fig. 2: Distribution of nondominated solutions in the final population of each EMOA in the objective space on the SYM-PART1 problem. The horizontal and vertical axis represent f_1 and f_2, respectively.

However, the IGD values achieved by MOEA/D-AD are only 1.3–2.6 times worse than the best IGD values on all the problems.

IGDX metric. Table 1(b) indicates that the three multi-modal multi-objective optimizers (MOEA/D-AD, MO_Ring_PSO_SCD, and Omni-optimizer) have good performance, regarding the IGDX indicator. In particular, MOEA/D-AD performs the best on the six MMOPs. MOEA/D-AD shows the second best performance only on the SSUF3 problem. In contrast, the performance of MOEA/D and MOEA/D-DU regarding the IGDX metric is quite poor. The IGDX values obtained by MOEA/D are 3.2–121.1 times worse than those by MOEA/D-AD. Thus, the new mechanism that maintains the solution space diversity in the population mainly contributes to the effectiveness of MOEA/D-AD.

Distribution of solutions found. Figs. 2 and 3 show the distribution of nondominated solutions in the final population of each EMOA in the objective and solution spaces on the SYM-PART1 problem. Again, we emphasize that only $N = 100$ nondominated solutions selected from the final population by using the selectSparseSolutions function (Algorithm 3) are shown for MOEA/D-AD in Figs. 2 and 3. Results of a single run with median IGD and IGDX values among 31 runs are shown in Figs. 2 and 3, respectively.

As shown in Fig. 2, the Pareto front of the SYM-PART1 problem is convex. While the distribution of nondominated solutions found by MOEA/D and MOEA/D-DU in the objective space is biased to the center of the Pareto front,

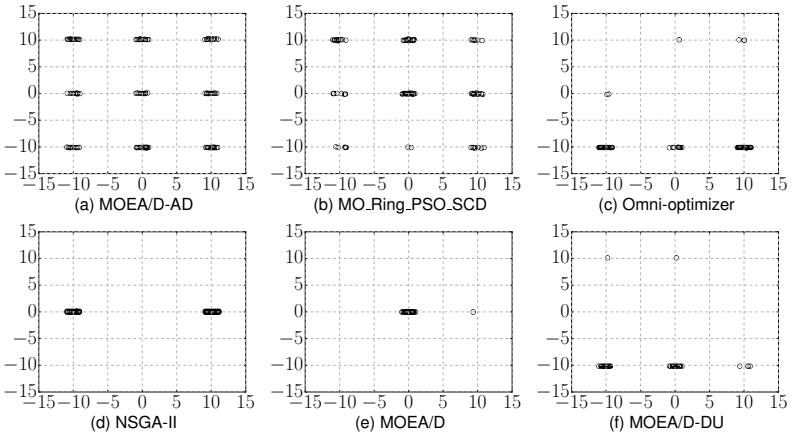

Fig. 3: Distribution of nondominated solutions in the final population of each EMOA in the solution space on the SYM-PART1 problem. The horizontal and vertical axis represent x_1 and x_2, respectively.

that by NSGA-II is uniform. Compared to the result of NSGA-II, nondominated solutions obtained by MOEA/D-AD and MO_Ring_PSO_SCD are not uniformly distributed in the objective space. This is because they also take into account the diversity of the population in the solution space.

The Pareto optimal solutions are on the nine lines in the SYM-PART1 problem. Fig. 3 shows that Omni-optimizer, NSGA-II, MOEA/D, and MOEA/D-DU fail to locate all the nine equivalent Pareto optimal solution sets. Solutions obtained by the four methods are only on a few lines. In contrast, MOEA/D-AD and MO_Ring_PSO_SCD successfully find nondominated solutions on all the nine lines. In particular, solutions obtained by MOEA/D-AD are more evenly distributed on the nine lines. Similar results to Figs. 2 and 3 are observed in other test problems. In summary, our results indicate that MOEA/D-AD is an efficient method for multi-modal multi-objective optimization.

4.2 Analysis of MOEA/D-AD

Influence of L on the performance of MOEA/D-AD. MOEA/D-AD has the control parameter L, which determines the neighborhood size of the child in the solution space. Generally speaking, it is important to understand the influence of control parameters on the performance of a novel evolutionary algorithm. Here, we investigate how L affects the effectiveness of MOEA/D-AD.

Table 2 shows results of MOEA/D-AD with six L values on the seven test problems. Due to space constraint, only aggregations of statistical testing results to MOEA/D-AD with $L = \lfloor 0.1\mu \rfloor$ are shown here. Intuitively, MOEA/D-AD with a large L value should perform well regarding the IGD metric because large L values relax the restriction for the environmental selection and may

Table 2: Results of MOEA/D-AD with various L values on the seven MMOPs. The tables (a) and (b) show aggregations of statistical testing results (+, −, and ≈) of the IGD and IGDX metrics. Each entry in the table shows the number of problems where the performance of MOEA/D-AD with each value of L is significantly better (worse) or has no significant difference from that of MOEA/D-AD with $L = \lfloor 0.1\mu \rfloor$.

(a) IGD

0.1μ	0.05μ	0.2μ	0.3μ	0.4μ	0.5μ
+ (better)	1	0	0	0	0
− (worse)	0	3	5	5	5
≈ (no sig.)	6	4	2	2	2

(b) IGDX

0.1μ	0.05μ	0.2μ	0.3μ	0.4μ	0.5μ
+ (better)	2	1	0	0	0
− (worse)	3	2	5	6	7
≈ (no sig.)	2	4	2	1	0

improve its ability for multi-objective optimization. However, Table 2(a) shows that the performance of MOEA/D-AD with a large L value is poor, regarding the IGD indicator. As pointed out in [15], the solution space diversity may help MOEA/D-AD to approximate the Pareto front well.

Table 2(b) indicates that the best IGDX values are obtained by $L = \lfloor 0.05\mu \rfloor$ and $L = \lfloor 0.2\mu \rfloor$ on two problems and one problem, respectively. Thus, the performance of MOEA/D-AD depends on the L value. A control method of L is likely to be beneficial for MOEA/D-AD. However, MOEA/D-AD with $L = \lfloor 0.1\mu \rfloor$ performs well on most problems. Therefore, $L = \lfloor 0.1\mu \rfloor$ can be the first choice.

Adaptive behavior of MOEA/D-AD. Unlike other MOEA/D-type algorithms, the population size μ and the number of individuals belonging to each subproblem are adaptively adjusted in MOEA/D-AD.

Fig. 4(a) shows the evolution of μ of MOEA/D-AD on the seven problems. In Fig. 4(a), μ is increased as the search progresses. This is because the diverse individuals in the solution space are iteratively added in the population. Recall that the number of equivalent Pareto optimal solutions n^{same} is 45 for the Omni-test problem, nine for the three SYM-PART problems, and two for other problems. Fig. 4(a) indicates that the trajectory of μ is problem-dependent. Ideally, μ should equal $n^{\text{same}} \times N$ so that n^{same} Pareto optimal solutions are assigned to each of N subproblems. However, the actual μ values are significantly larger than the expected values. For example, while the ideal μ value is 200 (2×100) on the Two-On-One problem, the actual μ value at the end of the search is 2 480.

To analyze the reason, we show the number of individuals assigned to each subproblem at the end of the search on the Two-On-One, Omni-test, and SYM-PART1 problems in Fig. 4(b). Fig. 4(b) indicates that the distribution of individuals is not even. More extra individuals are allocated to subproblems whose indices are close to 50. That is, MOEA/D-AD allocates unnecessary individuals to most subproblems. If individuals can be evenly assigned to each subproblem, the performance of MOEA/D-AD may be improved. An in-depth analysis of the adaptive behavior of MOEA/D-AD is needed.

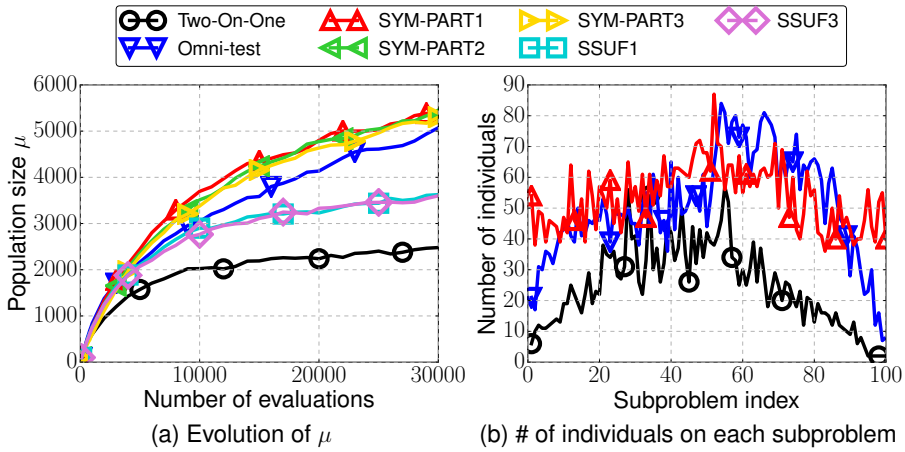

Fig. 4: (a) Evolution of the population size μ of MOEA/D-AD. (b) Number of individuals assigned to the j-th subproblem ($j \in \{1, ..., N\}$) at the end of the search, where $N = 100$. Results of a single run with a median IGDX value among 31 runs are shown.

5 Conclusion

We proposed MOEA/D-AD, which is a novel MOEA/D for multi-modal multi-objective optimization. In order to locate multiple equivalent Pareto optimal solutions, MOEA/D-AD assigns one or more individuals that are far from each other in the solution space to each subproblem. We examined the performance of MOEA/D-AD on the seven two-objective problems having equivalent Pareto optimal solutions. Our results indicate that MOEA/D-AD is capable of finding multiple equivalent Pareto optimal solutions. The results also show that MOEA/D-AD performs significantly better than Omni-optimizer and MO_Ring_PSO_SCD, which are representative multi-modal multi-objective optimizers.

Several interesting directions for future work remain. Any neighborhood criterion (e.g., sharing and clustering [8]) can be introduced in MOEA/D-AD. Although we used the relative distance-based neighborhood decision (Algorithm 2) in this study, investigating the performance of MOEA/D-AD with other neighborhood criteria is one future research topic. Also, the effectiveness of MOEA/D-AD could be improved by using an external archive that stores diverse solutions [12]. A more efficient search can be performed by utilizing a decision-maker's preference [5]. Incorporating the decision-maker's preference into the search process of MOEA/D-AD is an avenue for future work. Since the existing multi-modal multi-objective test problems are not scalable in the number of objectives and variables, this paper dealt with only two-objective problems with up to five variables. Designing scalable test problems is another research topic.

Acknowledgments. This work was supported by the Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. ZDSYS201703031748284).

References

1. K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.
2. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE TEVC*, 6(2):182–197, 2002.
3. K. Deb and S. Tiwari. Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *EJOR*, 185(3):1062–1087, 2008.
4. J. J. Durillo and A. J. Nebro. jMetal: A Java framework for multi-objective optimization. *Adv. Eng. Softw.*, 42(10):760–771, 2011.
5. M. Gong, F. Liu, W. Zhang, L. Jiao, and Q. Zhang. Interactive MOEA/D for multi-objective decision making. In *GECCO*, pages 721–728, 2011.
6. O. Kramer and H. Danielsiek. DBSCAN-based multi-objective niching to approximate equivalent pareto-subsets. In *GECCO*, pages 503–510, 2010.
7. F. Kudo, T. Yoshikawa, and T. Furuhashi. A study on analysis of design variables in Pareto solutions for conceptual design optimization problem of hybrid rocket engine. In *IEEE CEC*, pages 2558–2562, 2011.
8. X. Li, M. G. Epitropakis, K. Deb, and A. P. Engelbrecht. Seeking Multiple Solutions: An Updated Survey on Niching Methods and Their Applications. *IEEE TEVC*, 21(4):518–538, 2017.
9. J. J. Liang, C. T. Yue, and B. Y. Qu. Multimodal multi-objective optimization: A preliminary study. In *IEEE CEC*, pages 2454–2461, 2016.
10. M. Preuss, B. Naujoks, and G. Rudolph. Pareto Set and EMOA Behavior for Simple Multimodal Multiobjective Functions. In *PPSN*, pages 513–522, 2006.
11. G. Rudolph, B. Naujoks, and M. Preuss. Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets. In *EMO*, pages 36–50, 2007.
12. O. Schütze, M. Vasile, and C. A. C. Coello. Computing the Set of Epsilon-Efficient Solutions in Multiobjective Space Mission Design. *JACIC*, 8(3):53–70, 2011.
13. O. M. Shir, M. Preuss, B. Naujoks, and M. T. M. Emmerich. Enhancing Decision Space Diversity in Evolutionary Multiobjective Algorithms. In *EMO*, pages 95–109, 2009.
14. A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh. A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition. *IEEE TEVC*, 21(3):440–462, 2017.
15. T. Ulrich, J. Bader, and E. Zitzler. Integrating decision space diversity into hypervolume-based multiobjective search. In *GECCO*, pages 455–462, 2010.
16. Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao. Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers. *IEEE TEVC*, 20(2):180–198, 2016.
17. C. Yue, B. Qu, and J. Liang. A Multi-objective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multi-objective Problems. *IEEE TEVC*, 2017 (in press).
18. Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE TEVC*, 11(6):712–731, 2007.
19. A. Zhou, Q. Zhang, and Y. Jin. Approximating the Set of Pareto-Optimal Solutions in Both the Decision and Objective Spaces by an Estimation of Distribution Algorithm. *IEEE TEVC*, 13(5):1167–1189, 2009.
20. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE TEVC*, 7(2):117–132, 2003.