A Framework to Handle Multi-modal Multi-objective Optimization in Decomposition-based Evolutionary Algorithms

Ryoji Tanabe, *Member, IEEE,* and Hisao Ishibuchi, *Fellow, IEEE*

Abstract—Multi-modal multi-objective optimization is to locate (almost) equivalent Pareto optimal solutions as many as possible. While decomposition-based evolutionary algorithms have good performance for multi-objective optimization, they are likely to perform poorly for multi-modal multi-objective optimization due to the lack of mechanisms to maintain the solution space diversity. To address this issue, this paper proposes a framework to improve the performance of decomposition-based evolutionary algorithms for multi-modal multi-objective optimization. Our framework is based on three operations: assignment, deletion, and addition operations. One or more individuals can be assigned to the same subproblem to handle multiple equivalent solutions. In each iteration, a child is assigned to a subproblem based on its objective vector, i.e., its location in the objective space. The child is compared with its neighbors in the solution space assigned to the same subproblem. The performance of improved versions of six decomposition-based evolutionary algorithms by our framework is evaluated on various test problems regarding the number of objectives, decision variables, and equivalent Pareto optimal solution sets. Results show that the improved versions perform clearly better than their original algorithms.

Index Terms—Multi-modal multi-objective optimization, decomposition-based evolutionary algorithms, reference vector-based evolutionary algorithms, solution space diversity

I. INTRODUCTION

MULTI-OBJECTIVE optimization problems (MOPs) appear in real-world applications. Since no solution x can simultaneously minimize multiple objective functions in general, the goal of MOPs is usually to find a Pareto optimal solution preferred by a decision maker [1]. When the decision maker's preference information is unavailable a priori, an "a posteriori" decision making is conducted. The decision maker selects the final solution x^{final} from a set of solutions that approximates the Pareto front in the objective space.

An evolutionary multi-objective optimization algorithm (EMOA) is frequently used for the "a posteriori" decision making. Since EMOAs are population-based optimizers, they are likely to find a set of solutions in a single run. A number of EMOAs have been proposed in the literature. They can be classified into the following three categories: dominance-based EMOAs (e.g., NSGA-II [2]), indicator-based EMOAs (e.g.,

R. Tanabe and H. Ishibuchi are with Shenzhen Key Laboratory of Computational Intelligence, University Key Laboratory of Evolving Intelligent Systems of Guangdong Province, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China. e-mail: (rt.ryoji.tanabe@gmail.com, hisao@sustc.edu.cn). (Corresponding author: Hisao Ishibuchi)
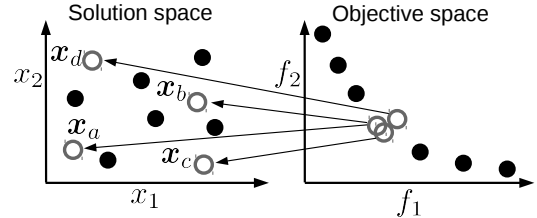
Fig. 1: Illustration of a situation where three solutions are almost the same in the objective space but dissimilar in the solution space.

IBEA [3]), and decomposition-based EMOAs (e.g., MOEA/D [4] and NSGA-III [5]). In particular, decomposition-based EMOAs have shown promising performance [6].

In the EMO community, it has been implicitly assumed that the decision maker is interested only in the distribution of solutions in the objective space. Thus, the distribution of solutions in the solution space has not received much attention. However, after the decision maker has selected the final solution x^{final} based on its objective vector $f(x^{\text{final}})$, she/he may want to examine other dissimilar solutions with equivalent quality or slightly inferior quality [7]–[9].

Fig. 1 shows a situation where the four solutions x_a, x_b, x_c, and x_d are far from each other in the solution space but close to each other in the objective space. Although x_d is a dominated solution, it may be acceptable for the decision maker. This is because the difference between $f(x_a)$, $f(x_b)$, $f(x_c)$, and $f(x_d)$ is small enough. If the decision maker has obtained multiple dissimilar solutions with similar quality, she/he can select x^{final} according to her/his preference in the solution space. For example, suppose that x_a in Fig. 1 becomes unavailable due to some accident (e.g., material shortages and mechanical failures) after the decision maker has selected x_a as x^{final}. In such a case, she/he can substitute one of x_b, x_c, and x_d for x_a. Schütze et al. give a more practical example on space mission design problems [9]. Sebag et al. also demonstrate the importance of multiple equivalent solutions for the decision maker on functional brain imaging problems [7]. Previous studies on other real-world problems with multiple equivalent solutions include diesel engine design problems [10], distillation plant layout problems [11], and rocket engine design problems [12].

A multi-modal multi-objective problem (MMOP) is to locate (almost) equivalent Pareto optimal solutions as many as possible [7], [13]. On the one hand, it is sufficient for

MOPs to find one of $\boldsymbol{x}_a$, $\boldsymbol{x}_b$, and $\boldsymbol{x}_c$ in Fig. 1 since their quality is almost the same. On the other hand, all of $\boldsymbol{x}_a$, $\boldsymbol{x}_b$, $\boldsymbol{x}_c$, and $\boldsymbol{x}_d$ should be found for MMOPs. Since multiple equivalent solutions play crucial roles in the reliable decision making process as explained above, MMOPs are important in practice. Evolutionary multi-modal multi-objective optimization algorithms (EMMAs) are specially designed optimizers for MMOPs. Unlike EMOAs, EMMAs have mechanisms to handle multiple equivalent solutions. Representative EMMAs include $P_{Q,\epsilon}$-MOEA [9], Omni-optimizer [13], DIOP [14], Niching-CMA [15], and MO_Ring_PSO_SCD [16].

In our earlier study [17], we proposed MOEA/D-AD, which is a specially designed MOEA/D for MMOPs. MOEA/D-AD can assign multiple individuals to each subproblem in order to handle equivalent solutions. For each iteration, a child $\boldsymbol{u}$ is assigned to a subproblem whose weight vector is closest to its objective vector $\boldsymbol{f}(\boldsymbol{u})$, in terms of the perpendicular distance. Then, $\boldsymbol{u}$ is compared to the individuals assigned to the same subproblem based on the weighted Tchebycheff function and the contribution to the solution space diversity.

In this paper, we extend our earlier study [17] and propose a general framework (called ADA) to handle multi-modal multi-objective optimization in a variety of decomposition-based EMOAs (including reference-based algorithms such NSGA-III [5] and θ-DEA [18]). The proposed ADA framework uses assignment, deletion, and addition operations. While decomposition-based EMOAs work well for multi-objective optimization, they are likely to perform poorly for multi-modal multi-objective optimization. This is because most of them do not have mechanisms to maintain the solution space diversity. In this paper, we show that efficient EMMAs could be realized by incorporating a solution space diversity maintenance mechanism into decomposition-based EMOAs. The proposed ADA framework facilitates the solution space diversity maintenance of existing EMMAs. We incorporate ADA into the following six representative decomposition-based EMOAs: MOEA/D-AGR [19], MOEA/D-DU [20], eMOEA/D [21], NSGA-III [5], θ-DEA [18], and RVEA [22]. We examine the ability of the ADA versions of those EMOAs (MOEA/D-AGR-ADA, MOEA/D-DU-ADA, eMOEA/D-ADA, NSGA-III-ADA, θ-DEA-ADA, and RVEA-ADA) to locate multiple equivalent Pareto optimal solutions on various test problems. We also analyze the behavior of those ADA versions.

The differences between this paper and our previous study [17] are as follows:

1) We propose ADA. While MOEA/D-AD [17] is a single algorithm for MMOPs, ADA is a framework to improve the performance of the existing decomposition-based EMOAs for MMOPs. In contrast to MOEA/D-AD, each configuration (e.g., the assignment operation) in ADA depends on an EMOA to be combined.

2) We demonstrate that ADA can be combined with the six decomposition-based EMOAs, including so-called reference vector-based EMOAs. MOEA/D-AD does not have such flexibility.

3) We introduce a practical two-phase decision making method with a solution set found by an ADA-based algorithm. We also introduce two post-processing methods

for benchmarking an ADA-based algorithm.

4) While we used only two-objective and two-variable problems in [17], we use problems with a various number of objectives M, design variables D, and equivalent Pareto optimal solution subsets O to investigate the scalability of EMMAs. Such a scale-up study has not been performed in the literature. We also use problems with distance-related variables [23], [24].

5) We compare the six ADA-based algorithms to three state-of-the-art EMMAs, including TriMOEA-TA&R [24] proposed after the publication of [17].

The rest of this paper is organized as follows. Section II provides some preliminaries of this paper. Section III introduces ADA and explains how to incorporate it into decomposition-based EMOAs. Section IV describes the experimental setup. Section V examines the performance of the six ADA variants. Section VI concludes this paper.

II. PRELIMINARIES

First, Subsections II-A and II-B give definitions of MOPs and MMOPs, respectively. Then, Subsections II-C and II-D describe decomposition-based EMOAs and reference vector-based EMOAs, respectively. Subsections II-C and II-D mainly explain components of the six existing EMOAs used in ADA. Algorithms S.1–S.6 in the supplementary file provide the overall procedures of the six EMOAs. Section III introduces how each component is used in ADA.

A. Definition of MOPs

A continuous MOP is to find a solution $\boldsymbol{x} \in \mathbb{S} \subseteq \mathbb{R}^D$ that minimizes a given objective function vector $\boldsymbol{f} : \mathbb{S} \to \mathbb{R}^M, \boldsymbol{x} \mapsto \boldsymbol{f}(\boldsymbol{x})$. $\mathbb{S} = \prod_{j=1}^{D}[x_j^{\min}, x_j^{\max}]$ is the D-dimensional solution space where $x_j^{\min} \leq x_j \leq x_j^{\max}$ for each index $j \in \{1, ..., D\}$. $\mathbb{R}^M$ is the M-dimensional objective space.

A solution $\boldsymbol{x}_1$ is said to dominate $\boldsymbol{x}_2$ if and only if $f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$ for all $i \in \{1, ..., M\}$ and $f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2)$ for at least one index i. If $\boldsymbol{x}^*$ is not dominated by any other solutions, it 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 for the "a posteriori" decision making is to find a non-dominated solution set that approximates the Pareto front in the objective space.

B. Definition of MMOPs

Although the term "multi-modal multi-objective optimization" was firstly coined in [7], [25] in 2005, the definition of MMOPs has not been explicitly given in most previous studies. We consider the following two types of MMOPs: (i) *Type1-MMOP* is to locate all Pareto optimal solutions. (ii) *Type2-MMOP* is to locate all Pareto optimal solutions and non-Pareto optimal solutions which have acceptable quality for the decision maker. Those non-Pareto optimal solutions should be far from the Pareto optimal solutions and the other non-Pareto optimal solutions in the solution space.

For example, $\boldsymbol{x}_a$, $\boldsymbol{x}_b$, and $\boldsymbol{x}_c$ in Fig. 1 should be found for Type1-MMOPs. In addition, the non-Pareto optimal solution

$\boldsymbol{x}_d$ should be found for Type2-MMOPs if its quality $\boldsymbol{f}(\boldsymbol{x}_d)$ is acceptable to the decision maker. While most existing studies (e.g., [13], [16]) assume Type1-MMOPs, only a few studies (e.g., [7], [9], [14]) address Type2-MMOPs. Although there is room for discussion, Type2-MMOPs may be more practical than Type1-MMOPs. Diverse solutions with similar quality to the final solution $\boldsymbol{x}^{\text{final}}$ are beneficial to the decision maker even if their quality is slightly worse than $\boldsymbol{x}^{\text{final}}$ [7], [9].

Nevertheless, we focus only on Type1-MMOPs in this paper. The main reason is due to the difficulty in benchmarking EMMAs on Type2-MMOPs. In contrast to Type1-MMOPs, Type2-MMOPs are loosely defined. This is because the terms "acceptable quality" and "far from" in Type2-MMOPs significantly depend on the decision maker. It is difficult to define the two key factors in Type2-MMOPs for benchmarking purposes in a fair manner. How to evaluate the performance of EMMAs for Type2-MMOPs itself is another research topic.

C. Decomposition-based EMOAs (MOEA/D-type algorithms)

Although some frameworks of decomposition-based EMOAs have been proposed in the literature, MOEA/D-type algorithms show the promising performance [6]. Here, we describe MOEA/D-type algorithms. First, we explain the most basic MOEA/D [4] and its modified version called MOEA/D-DE [26]. The framework of MOEA/D-DE is used in most MOEA/D-type algorithms. Then, we describe components of three MOEA/D-type algorithms: MOEA/D-AGR [19], MOEA/D-DU [20], and eMOEA/D [21].

1) MOEA/D: The most basic MOEA/D [4] decomposes an M-objective MOP into N single-objective subproblems using a scalarizing function $g : \mathbb{R}^M \to \mathbb{R}$ and a set of uniformly distributed weight vectors $\boldsymbol{W} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_N\}$. For each $i \in \{1, ..., N\}$, $\boldsymbol{w}_i = (w_{i,1}, ..., w_{i,M})^{\text{T}}$, and $\sum_{k=1}^{M} w_{i,k} = 1$. The j-th individual $\boldsymbol{x}_j$ in the population $\boldsymbol{P}$ is assigned to the j-th subproblem with $\boldsymbol{w}_j$. Thus, the population size μ is always equal to the number of weight vectors N. The j-th subproblem also has its neighborhood index list $\boldsymbol{B}_j = \{b_{j,1}, ..., b_{j,S}\}$, which consists of indices of the S closest weight vectors to $\boldsymbol{w}_j$ in the weight vector space.

After the initialization of $\boldsymbol{P}$ and $\boldsymbol{W}$, the following steps are repeatedly performed until a termination condition is satisfied. In each iteration t, for the j-th subproblem ($j \in \{1, ..., N\}$), an index list $\boldsymbol{T}$ is set to $\boldsymbol{B}_j$. Two indices a and b of the parent individuals $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ are randomly selected from $\boldsymbol{T}$. $\boldsymbol{u}$ is generated by applying variation operators to $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. The SBX crossover and the polynomial mutation [27] are used in the original MOEA/D. After $\boldsymbol{u}$ has been generated, the replacement selection is applied to each neighborhood subproblem $k \in \boldsymbol{T}$. The current solution $\boldsymbol{x}_k$ of the k-th subproblem is replaced with $\boldsymbol{u}$ if $g(\boldsymbol{u}|\boldsymbol{w}_k) \leq g(\boldsymbol{x}_k|\boldsymbol{w}_k)$.

2) MOEA/D-DE: The differences between MOEA/D and MOEA/D-DE are threefold. First, the differential evolution (DE) operator [28] is used instead of the SBX crossover [27]. Second, two types of index lists ($\boldsymbol{B}_j$ and $\{1, ..., N\}$) are used. The index list $\boldsymbol{T}$ is set to $\boldsymbol{B}_j$ with a probability of $\delta \in [0, 1]$ and $\{1, ..., N\}$ with a probability of $1 - \delta$. Thus, $\boldsymbol{T}$ can be set to all individual indices. Third, the maximum number of individuals replaced by the child $\boldsymbol{u}$ is restricted to n^{rep}.

3) MOEA/D-AGR: The difference between MOEA/D-AGR and MOEA/D-DE is caused by a replacement method. In MOEA/D-AGR, indices of subproblems to be updated by a newly generated solution $\boldsymbol{u}$ are set to K neighborhood indices of the j-th subproblem in the weight vector space where j is the subproblem index whose scalarizing function value $g(\boldsymbol{u}|\boldsymbol{w}_j)$ is the best among all N subproblems:

$$j = \underset{k \in \{1, ..., N\}}{\operatorname{argmin}} \big\{ g(\boldsymbol{u}|\boldsymbol{w}_k) \big\}. \tag{1}$$

The weighted Tchebycheff function g^{tch} is used in (1):

$$g^{\text{tch}}(\boldsymbol{x}|\boldsymbol{w}) = \max_{i \in \{1, ..., M\}} \big\{ w_i |f_i(\boldsymbol{x}) - z_i^*| \big\}, \tag{2}$$

where $\boldsymbol{z}^* = (z_1^*, ..., z_M^*)^{\text{T}}$ is the ideal point. Since finding the true ideal point is difficult in general, its approximation is used in (2). The i-th element of the approximation of $\boldsymbol{z}^*$ is the minimum objective value found during the search process.

The replacement neighborhood size K plays a crucial role in balancing exploration and exploitation in MOEA/D-AGR in a similar manner to S in MOEA/D. A large K value encourages exploitation. In MOEA/D-AGR, the K value deterministically increases with the number of iterations. The results presented in [19] show that a scheduling method based on the sigmoid function is suitable for MOEA/D-AGR.

4) MOEA/D-DU: In contrast to MOEA/D-AGR, the selection of subproblems that need to be updated is based on the distance in the objective and weight vector spaces in MOEA/D-DU. In the normalized objective space, the perpendicular distance between the normalized objective vector $\boldsymbol{f}'(\boldsymbol{u})$ and $\boldsymbol{w}_j$ is calculated for each $j \in \{1, ..., N\}$. The replacement is performed for K subproblems with the minimum perpendicular distance. Similar to MOEA/D-AGR, K controls the balance between exploration and exploitation.

MOEA/D-DU uses the division version of the weighted Tchebycheff function g^{dtch} [29]:

$$g^{\text{dtch}}(\boldsymbol{x}|\boldsymbol{w}) = \max_{i \in \{1, ..., M\}} \left\{ \frac{|f_i(\boldsymbol{x}) - z_i^*|}{w_i} \right\}, \tag{3}$$

if $w_i = 0$, it is set to 10^{-6} to avoid division by zero. It is reported in [29] that the distribution of the search directions of MOEA/D with g^{dtch} is more uniform than that with g^{tch}.

5) eMOEA/D: In eMOEA/D, the replacement method is applied to K subproblems with the minimum scalarizing function values. MOEA/D-AGR and eMOEA/D are the same regarding the use of scalarizing function values to select subproblems. The following multiplicative scalarizing function (MSF) [21] is used in eMOEA/D:

$$g^{\text{msf}}(\boldsymbol{x}|\boldsymbol{w}) = \frac{\left(\max_{i \in \{1, ..., M\}} \big\{ \frac{1}{w_i} |f_i(\boldsymbol{x} - z_i^*)| \big\} \right)^{1+\alpha}}{\left(\min_{i \in \{1, ..., M\}} \big\{ \frac{1}{w_i} |f_i(\boldsymbol{x} - z_i^*)| \big\} \right)^{\alpha}}, \tag{4}$$

where α controls the size of the so-called improvement region. α plays a similar role in the penalty value θ of the PBI function in (7), which will be explained later. Even if $\boldsymbol{f}(\boldsymbol{x})$ is far from $\boldsymbol{w}_j$ regarding the the perpendicular distance, $\boldsymbol{x}$ would be evaluated as being superior using a sufficiently large α value. In (4), g^{msf} with $\alpha = 0$ is identical to g^{dtch}.

According to a general rule of thumb "emphasize diversity and convergence at the early and later stages, respectively", α decreases linearly with the number of iterations t:

$$\alpha = \beta\left(1 - \frac{t}{t^{\max}}\right)\left(M\left(\min_{k\in\{1,...,M\}}\{w_k\}\right)\right), \quad (5)$$

where $t^{\max}$ is the maximum number of iterations. The recommended value of β is 1.

D. Reference vector-based EMOAs

Representative reference vector-based EMOAs include NSGA-III [5], θ-DEA [18], RVEA [22], VaEA [30], and SPEA/R [31]. While $w \in W$ is called the weight vector in decomposition-based EMOAs, it is referred to the reference vector in reference vector-based EMOAs. Below, we explain NSGA-III, θ-DEA, and RVEA.

1) NSGA-III: NSGA-III is an improved version of NSGA-II [2] for many-objective optimization. For each iteration t, children Q are generated by applying variation operators to randomly selected pairs of individuals from P. In the environmental selection, μ individuals for the next iteration $t+1$ are selected from the union of P and Q. The primary and secondary criteria are based on the non-domination levels and the reference vector-based niching method, respectively.

Individuals in the union $P \cup Q$ are grouped as $F_1, F_2, ...$ according to their non-domination levels. First, P and the front index i are initialized as $P = \emptyset$ and $i = 1$, respectively. Then, individuals in F_i are added to P and i is incremented until $|P| + |F_i| \geq \mu$. After this operation, $P = F_1 \cup ... \cup F_{l-1}$, where l is the index of the last front. If $|P| < \mu$, other $\mu - |P|$ individuals are selected from the last front F_l using the niching method described below. At the beginning of the niching procedure, an individual x in P and F_l is assigned to the j-th subproblem with the minimum perpendicular distance between its normalized objective vector $f'(x)$ and w_j:

$$j = \operatorname*{argmin}_{k\in\{1,...,N\}}\left\{\mathrm{PD}\big(f'(x), w_k\big)\right\}, \quad (6)$$

where the function PD returns the perpendicular distance between two input vectors. After the assignments of all individuals, other individuals in the next iteration are selected from F_l based on the number of individuals assigned to each subproblem and their perpendicular distance.

2) θ-DEA: The environmental selection in θ-DEA is similar to that of NSGA-III. However, the secondary criterion in θ-DEA is based on the so-called θ-dominance. First, each individual in the union of P and F_l is assigned to the j-th subproblem using (6). Then, individuals assigned to each subproblem are ranked based on their θ-dominance levels. Let us assume that two individuals x and y are assigned to the same j-th subproblem. x is said to θ-dominate y if $g^{\mathrm{pbi}}(x|w_j) < g^{\mathrm{pbi}}(y|w_j)$. The PBI function g^{pbi} is given as:

$$g^{\mathrm{pbi}}(x|w) = d_1 + \theta\, d_2, \quad (7)$$

$$d_1 = \frac{\|(f(x) - z^*)^{\mathrm{T}}\, w\|}{\|w\|}, \quad (8)$$

$$d_2 = \left\|f(x) - \left(z^* + d_1\frac{w}{\|w\|}\right)\right\|, \quad (9)$$

where $\|a\|$ indicates the Euclidean norm of a. The distance d_1 represents how close the objective vector $f(x)$ is to the Pareto front, and d_2 is the perpendicular distance between $f(x)$ and w. The penalty parameter θ balances the convergence (d_1) and the diversity (d_2). The recommended θ value is 10^6 for M vectors w with the axis directions (e.g., $(1, 0, ..., 0)^{\mathrm{T}}$) and 5 for all the other vectors w.

3) RVEA: RVEA uses a set of unit reference vectors $V = \{v_1, ..., v_N\}$, instead of a set of reference vectors $W = \{w_1, ..., w_N\}$, where $v_i = w_i/\|w_i\|$ for each $i \in \{1, ..., N\}$. RVEA adaptively adjusts V based on the current P.

After children Q have been generated, the environmental selection is applied to the union of P and Q. First, for each individual in $P \cup Q$, $f(x)$ is transformed as $f'(x) = f(x) - z^*$. Then, x in $P \cup Q$ is assigned to the j-th subproblem with the minimum angle between $f'(x)$ and v_j:

$$j = \operatorname*{argmin}_{k\in\{1,...,N\}}\left\{\mathrm{angle}\big(f'(x), v_k\big)\right\}, \quad (10)$$

where the function $\mathrm{angle}(a, b)$ in (10) returns the angle between the two input vectors a and b.

Then, individuals assigned to the same subproblem are compared based on their angle-penalized distance (APD) values. For each $j \in \{1, ..., N\}$, the best individual with the minimum APD value can survive to the next iteration. The APD value of x with the unit reference vector v is given as:

$$\mathrm{APD}(x) = \big(1 + P(x, v)\big)\,\|f'(x)\|, \quad (11)$$

$$P(x, v) = M\left(\frac{t}{t^{\max}}\right)^\alpha\left(\frac{\mathrm{angle}\big(f'(x), v\big)}{\gamma(v)}\right), \quad (12)$$

$$\gamma(v) = \min_{s\in V\setminus\{v\}}\left\{\mathrm{angle}(v, s)\right\}, \quad (13)$$

where $P(x, v)$ is a penalty value for x. The larger the angle between $f'(x)$ and v is, the larger the penalty value is given to x. $\gamma(v)$ is used to normalize the angle. t is the current number of iterations, and $t^{\max}$ is the maximum number of iterations. The influence of the angle-based penalty scheme increases as the search progresses. The recommended setting of α is 2.

III. PROPOSED ADA FRAMEWORK

This section explains the proposed ADA framework and the six ADA versions. Unlike EMOAs for MOPs, EMMAs for MMOPs need to maintain the diversity of the population in both the objective and solution spaces. For example, Omni-optimizer [13] uses an aggregate crowding distance metric in the objective and solution spaces. While most EMMAs (e.g., [14]–[16]) aggregate the objective and solution space diversity metrics similar to Omni-optimizer, ADA uses them in a two-phase manner similar to TriMOEA-TA&R [24]. More specifically, ADA handles the objective space diversity by an assignment method in an original EMOA and the solution space diversity by a simple niching criterion.

On the one hand, a single individual is assigned to each subproblem in most decomposition-based EMOAs. Thus, the population size μ is equal to the number of weight/reference vectors N (i.e., $\mu = N$). On the other hand, one or more individuals can be assigned to each subproblem in the ADA

framework (i.e., $\mu \geq N$). This mechanism is to maintain multiple equivalent individuals in each subproblem. The μ value is adaptively adjusted in ADA.

Algorithm 1 shows the ADA framework. Algorithms S.7–S.12 in the supplementary file show the six ADA-based algorithms. Algorithms S.7–S.12 are almost the same to Algorithm 1. Lines 3, 11, and 16 are different between Algorithms S.7–S.12 and Algorithm 1. At the beginning of the search, μ is set to N (line 1). The population $\boldsymbol{P}$ and the weight/reference vector set $\boldsymbol{W}$ are also initialized. The i-th individual $\boldsymbol{x}_i$ in $\boldsymbol{P}$ is assigned to the i-th subproblem (lines 2–3). This operation is unnecessary for those EMOAs with a reassignment procedure of individuals to subproblems such as RVEA. After the child $\boldsymbol{u}$ has been generated by the reproduction operations (line 6–8), the population $\boldsymbol{P}$ is updated using the assignment (line 10), deletion (lines 16–17), and addition (lines 18–19) operations. The assignment and deletion operations differ depending on an EMOA to be combined with the ADA framework. After the normalization of the $\mu + 1$ objective vectors (line 9), $\boldsymbol{u}$ is assigned to the j-th subproblem (line 10). $\boldsymbol{X}$ denotes a set of individuals that have been assigned to the j-th subproblem and are in the neighborhood of $\boldsymbol{u}$ in the solution space (line 11). Thus, ADA requires a neighborhood criterion in the solution space. $\boldsymbol{X}$ is explained later in detail using Fig. 2. Two Boolean variables b^{winner} and b^{explorer} (line 12) are used in the addition operation (lines 18–19). $\boldsymbol{u}$ enters $\boldsymbol{P}$ if it satisfies either of the following two conditions. One is that there is no individual in $\boldsymbol{X}$ (lines 13–14). The other is that $\boldsymbol{u}$ is better than at least one individual in $\boldsymbol{X}$ in the deletion operation (lines 15–17).

The following Subsections (from III-A to III-F) explain each step of ADA in detail. Subsection III-G presents an effective decision making based on a solution set found by the proposed approach. Apart from the decision making, Subsection III-H introduces two post-processing methods for benchmarking. Subsection III-I discusses the applicability of ADA. Subsection III-J discusses the originality of ADA.

A. Reproduction operation

ADA uses the basic GA operators (i.e., the SBX crossover and the polynomial mutation [27]) and the simplest method of selecting parents. First, two parents $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ are randomly selected from $\boldsymbol{P}$ such that $a \neq b$, regardless of the subproblem to which each individual has been assigned. Then, $\boldsymbol{u}$ is reproduced by applying SBX to $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. Finally, the polynomial mutation is applied to $\boldsymbol{u}$. All ADA-based algorithms use the same mating selection scheme, regardless of the mating selection schemes in their original EMOAs.

In principle, any variation operators can be incorporated into ADA, including DE operators [26], PSO operators [16], and model-based methods [15], [32]. Any parent selection methods can also be used in ADA, including the distance-based selection methods in the solution space [33]. The performance of ADA can be improved by using these more sophisticated methods. However, if such effective methods are used in ADA, it is unclear which algorithmic component mainly contributes to the overall performance of ADA-based algorithms. Since we want to investigate the effectiveness of the ADA framework

---

Algorithm 1: The ADA framework

1. $t \leftarrow 1$, $\mu \leftarrow N$, initialize the population $\boldsymbol{P} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_\mu\}$ and the weight/reference vector set $\boldsymbol{W} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_N\}$;
2. **for** $i \in \{1, ..., N\}$ **do**
3.   Assign $\boldsymbol{x}_i$ to the i-th subproblem;
4. **while** The termination criteria are not met **do**
5. $\mu \leftarrow |\boldsymbol{P}|$;
6.   Randomly select a and b from $\{1, ..., \mu\}$ such that $a \neq b$;
7.   Generate the child $\boldsymbol{u}$ by applying the crossover operation to $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$;
8.   Apply the mutation operation to $\boldsymbol{u}$;
9.   Normalize the objective vectors $\boldsymbol{f}(\boldsymbol{x}_1), ..., \boldsymbol{f}(\boldsymbol{x}_\mu), \boldsymbol{f}(\boldsymbol{u})$;
10.   Assign $\boldsymbol{u}$ to the j-th subproblem;
11. $\boldsymbol{X} \leftarrow \{\boldsymbol{x} \in \boldsymbol{P} | \boldsymbol{x}$ has been assigned to the j-th subproblem and is in the neighborhood of $\boldsymbol{u}$ in the solution space$\}$;
12. $b^{\text{explorer}} \leftarrow$ FALSE and $b^{\text{winner}} \leftarrow$ FALSE;
13.   **if** $\boldsymbol{X} = \emptyset$ **then**
14. $b^{\text{explorer}} \leftarrow$ TRUE;
15.   **for** $\boldsymbol{x} \in \boldsymbol{X}$ **do**
16.   **if** $\boldsymbol{x}$ is worse than $\boldsymbol{u}$ **then**
17. $\boldsymbol{P} \leftarrow \boldsymbol{P} \backslash \{\boldsymbol{x}\}$ and $b^{\text{winner}} \leftarrow$ TRUE
18.   **if** $b^{\text{winner}} =$ TRUE or $b^{\text{explorer}} =$ TRUE **then**
19. $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup \{\boldsymbol{u}\}$;
20. $t \leftarrow t + 1$;
21. **return** $\boldsymbol{P}$ for the decision making (Subsection III-G) or benchmarking (Subsection III-H);

---

in an isolated manner, we use the simplest variation operators and parent selection method in this paper.

B. Normalization

Since the objective functions of most real-world problems are differently scaled, the normalization method is mandatory. In ADA, the normalization method depends on an EMOA to be combined. For example, NSGA-III-ADA uses the intercept-based normalization method in NSGA-III.

If the original EMOA does not have a normalization method (e.g., MOEA/D-AGR), we use the following simple normalization method. The objective vector $\boldsymbol{f}(\boldsymbol{x})$ is normalized using the approximated ideal point $\boldsymbol{z}^* = (z_1^*, ..., z_M^*)^{\text{T}}$ and the worst point in the population $\boldsymbol{z}^{\text{worst}} = (z_1^{\text{worst}}, ..., z_M^{\text{worst}})^{\text{T}}$. The i-th element of the normalized objective vector $\boldsymbol{f}'(\boldsymbol{x})$ is given as: $f_i'(\boldsymbol{x}) = (f_i(\boldsymbol{x}) - z_i^*)/(z_i^{\text{worst}} - z_i^*)$.

C. Assignment operation

The child $\boldsymbol{u}$ is assigned to the j-th subproblem based on its normalized objective vector (line 10 in Algorithm 1). The assignment operation plays a crucial role in ADA to maintain the diversity of the population in the objective space.

Each EMOA has a different method to select the index j of the subproblem to which $\boldsymbol{u}$ is assigned. MOEA/D-AGR-ADA and eMOEA/D-ADA select the j-th subproblem with the minimum scalarizing function value $g(\boldsymbol{u} | \boldsymbol{w}_j)$ as in (1). MOEA/D-DU-ADA, NSGA-III-ADA, and θ-DEA-ADA assign $\boldsymbol{u}$ to the j-th subproblem with the minimum perpendicular distance between $\boldsymbol{f}'(\boldsymbol{u})$ and $\boldsymbol{w}_j$ as in (6). RVEA selects the subproblem

with the minimum angle between $\boldsymbol{f}'(\boldsymbol{u})$ and the unit reference vector $\boldsymbol{v}_j$ as in (10). ADA requires only a single index j for $\boldsymbol{u}$ whereas MOEA/D-AGR, eMOEA/D, and MOEA/D-DU select K subproblem indices from $\{1, ..., N\}$.

D. Neighborhood criterion in the solution space

ADA requires the neighborhood criterion in the solution space (line 11 in Algorithm 1). While the objective space diversity is maintained by the assignment operation, the solution space diversity is controlled by the neighborhood criterion.

We use a simple relative distance-based neighborhood criterion presented in [17]. First, the normalized Euclidean distance between each individual $\boldsymbol{x}$ in $\boldsymbol{P}$ and $\boldsymbol{u}$ is calculated in the solution space. The upper and lower bounds for each decision variable of a problem are used for the normalization. Then, all μ individuals in $\boldsymbol{P}$ are sorted based on their distance values in descending order. If $\boldsymbol{x}$ is within the L nearest individuals from $\boldsymbol{u}$, $\boldsymbol{x}$ is said to be a neighbor of $\boldsymbol{u}$ in the solution space. L is a control parameter in this neighborhood criterion.

In addition to the relative distance-based neighborhood criterion, any neighborhood criterion can be incorporated into ADA. A number of niching methods have been proposed in the multi-modal single-objective optimization community [34]. Modern niching methods include the adaptive radius-based method [35] and the nearest-better clustering [36]. However, our preliminary results show that such sophisticated niching methods do not work well in ADA. The main reason for the failure is that an appropriate parameter specification for such a modern niching method is difficult due to the difficulty in understanding the fitness landscape of an MOP [37]. For this reason, we use the simple neighborhood criterion.

E. Deletion operation

Let $\boldsymbol{X}$ be a set of individuals that are in the neighborhood of $\boldsymbol{u}$ in the solution space among the individuals assigned to the same j-th subproblem as $\boldsymbol{u}$ (line 11 in Algorithm 1). Fig. 2 shows an example of $\boldsymbol{X}$. In Fig. 2, $\mu = 10$, $N = 4$, $L = 3$, and $\boldsymbol{u}$ has been assigned to the third-subproblem ($\boldsymbol{w}_3$). A set of neighborhood individuals of $\boldsymbol{u}$ in the solution space is $\boldsymbol{Y} = \{\boldsymbol{x}_b, \boldsymbol{x}_c, \boldsymbol{x}_d\}$. A set of individuals assigned to the third subproblem in the objective space is $\boldsymbol{Z} = \{\boldsymbol{x}_a, \boldsymbol{x}_b, \boldsymbol{x}_c\}$. In this case, $\boldsymbol{X} = \boldsymbol{Y} \cap \boldsymbol{Z} = \{\boldsymbol{x}_b, \boldsymbol{x}_c\}$. Whereas $\boldsymbol{x}_d$ is the neighborhood individual of $\boldsymbol{u}$, $\boldsymbol{x}_d$ has been assigned to the second subproblem ($\boldsymbol{w}_2$). $\boldsymbol{x}_a$ has been assigned to the third subproblem with $\boldsymbol{u}$, but $\boldsymbol{x}_a$ is not in the neighborhood of $\boldsymbol{u}$. Since $\boldsymbol{x}_a$ is dominated by $\boldsymbol{u}$ with respect to the two objectives in Fig. 2, $\boldsymbol{x}_a$ is deleted in most EMOAs. In contrast, $\boldsymbol{x}_a$ can survive in $\boldsymbol{P}$ in ADA. This is because $\boldsymbol{x}_a$ is not a neighbor of $\boldsymbol{u}$ (and any other individual assigned to the third subproblem). Thus, $\boldsymbol{x}_a$ is not compared to $\boldsymbol{u}$. Although the quality of $\boldsymbol{x}_a$ is poor, $\boldsymbol{x}_a$ contributes to the solution space diversity of $\boldsymbol{P}$.

A paired comparison between $\boldsymbol{u}$ and each $\boldsymbol{x}$ in $\boldsymbol{X}$ is performed (lines 15–17 in Algorithm 1). If $\boldsymbol{x}$ is evaluated as being worse than $\boldsymbol{u}$ (by the evaluation criterion explained in the next paragraph), $\boldsymbol{x}$ is deleted from $\boldsymbol{P}$. The deletion of such $\boldsymbol{x}$ is reasonable since it is based on both the quality in the objective space and the diversity in the solution space.
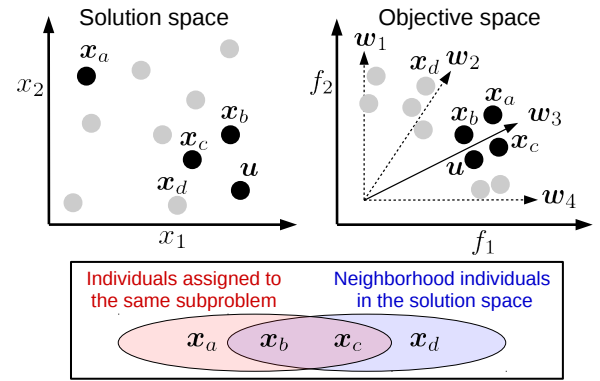

Fig. 2: Example of $\boldsymbol{X}$. In this example, $\boldsymbol{X} = \{\boldsymbol{x}_b, \boldsymbol{x}_c\}$. Since $\boldsymbol{x}_a$ and $\boldsymbol{x}_d$ are not compared to the child $\boldsymbol{u}$ in ADA, $\boldsymbol{x}_a$ and $\boldsymbol{x}_d$ survive to the next iteration independent of their quality.

The comparison criterion depends on the environmental selection in each original EMOA. In MOEA/D-type algorithms, the comparison is based on the scalarizing function g. MOEA/D-AGR-ADA, MOEA/D-DU-ADA, and eMOEA/D-ADA use g^{tch} in (2), g^{dtch} in (3), and g^{msf} in (4), respectively. If $g(\boldsymbol{x}|\boldsymbol{w}_j) \geq g(\boldsymbol{u}|\boldsymbol{w}_j)$ on the j-th subproblem, $\boldsymbol{x}$ is removed from $\boldsymbol{P}$. In NSGA-III-ADA and θ-DEA-ADA, the primary comparison between $\boldsymbol{x}$ and $\boldsymbol{u}$ is based on the Pareto-dominance relation. In NSGA-III-ADA, ties are broken by the perpendicular distance between the normalized objective vector and $\boldsymbol{w}_j$. In θ-DEA-ADA, the tie-breaker is the θ-dominance. RVEA-ADA uses the APD value given in (11).

F. Addition operation

The child $\boldsymbol{u}$ is added to the population $\boldsymbol{P}$ if either of the following two conditions is met (lines 18–19 in Algorithm 1). One is that no individual exists in $\boldsymbol{X}$. An empty $\boldsymbol{X}$ means that there is no neighborhood individual of $\boldsymbol{u}$ in $\boldsymbol{P}$ in the solution space (or the objective space). If the first condition is met, $\boldsymbol{u}$ enters $\boldsymbol{P}$ without any comparison. Although $\boldsymbol{u}$ with inferior quality is likely to enter $\boldsymbol{P}$, it helps $\boldsymbol{P}$ to maintain the solution space diversity (or the objective space diversity). While a dominated individual is unlikely to survive to the next iteration in most EMOAs, it can remain in $\boldsymbol{P}$ due to the first criterion in the addition operation.

The other is that $\boldsymbol{u}$ performs better than at least one individual in $\boldsymbol{X}$ in the deletion operation (lines 16–17 in Algorithm 1). Since $\boldsymbol{u}$ has good quality in its neighborhood in the solution space, it should be added to $\boldsymbol{P}$.

G. Decision making

We introduce an effective decision making with a solution set found by an ADA-based algorithm. Fig. 3 shows an example of the decision making. Fig. 3 (a) exhibits the distribution of all objective vectors in the final population $\boldsymbol{P}$ of MOEA/D-AGR-ADA on the two-objective and two-variable SYM-PART1 problem [38]. As shown in Fig. 3 (d), nine equivalent Pareto optimal solution sets are on the nine lines in SYM-PART1. The experimental setting is described in Section IV later. The decision making in ADA is the following two-phase method based on $\boldsymbol{A}^{\text{primary}}$ and $\boldsymbol{A}^{\text{secondary}}$.

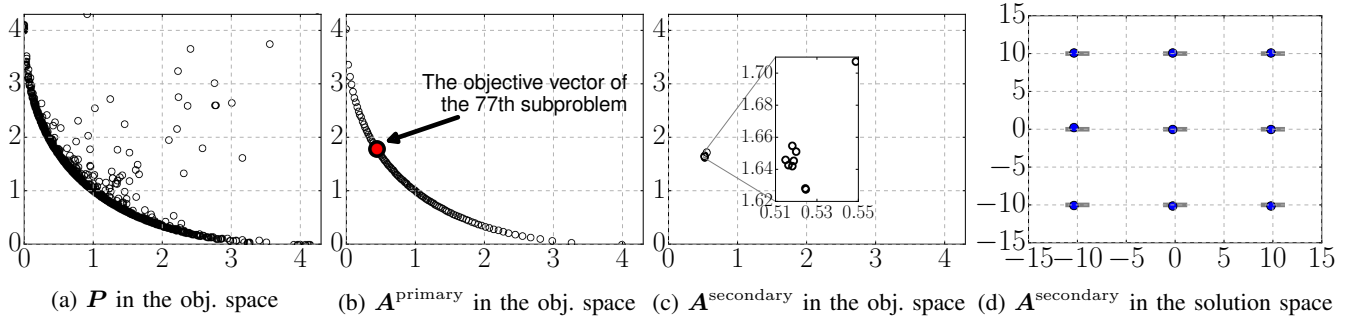| (a) $\boldsymbol{P}$ in the obj. space | (b) $\boldsymbol{A}^{\mathrm{primary}}$ in the obj. space | (c) $\boldsymbol{A}^{\mathrm{secondary}}$ in the obj. space | (d) $\boldsymbol{A}^{\mathrm{secondary}}$ in the solution space |

Fig. 3: Example of the decision making based on multiple equivalent solutions found by the proposed approach. Solutions obtained by MOEA/D-AGR-ADA on SYM-PART1 are shown. These figures show results of a single run with a median IGD$^+$ value among 31 runs. In Fig. 3 (a), (b), and (c), the x and y axes represent f_1 and f_2, respectively. In Fig. 3 (d), the x and y axes represent x_1 and x_2, respectively. The nine gray lines in Fig. 3 (d) are the equivalent Pareto optimal solution sets.

1) $\boldsymbol{A}^{\mathrm{primary}}$-based decision making: Fig. 3 (a) shows that $\boldsymbol{P}$ contains a large number of solutions. This is undesirable for the decision maker, because she/he usually wants to examine a small number of well-distributed solutions in the objective space [4]. To address this issue, the best solution for the j-th subproblem is selected from $\boldsymbol{P}$ based on the same criterion in the deletion operation ($j \in \{1, ..., N\}$). Then, only non-dominated solutions are selected from the N best solutions. Let $\boldsymbol{A}^{\mathrm{primary}}$ be a set of the non-dominated solutions obtained by this procedure. Fig. 3 (b) shows non-dominated solutions in $\boldsymbol{A}^{\mathrm{primary}}$ in the objective space. The decision maker examines objective vectors in $\boldsymbol{A}^{\mathrm{primary}}$, rather than $\boldsymbol{P}$.

When the decision maker wants to examine N or less non-dominated solutions (e.g., only 10 solutions), other solution selection methods can be used to select $\boldsymbol{A}^{\mathrm{primary}}$ from $\boldsymbol{P}$. If the number of objectives M is less than 4, efficient hypervolume-based selection approaches (e.g., [39]) are available. If $M \geq 4$, computationally cheap distance-based selection methods (e.g., [40]) can be used.

2) $\boldsymbol{A}^{\mathrm{secondary}}$-based decision making: After the decision maker has examined $\boldsymbol{A}^{\mathrm{primary}}$ in the objective space, she/he selects the final solution $\boldsymbol{x}^{\mathrm{final}}$ from $\boldsymbol{A}^{\mathrm{primary}}$. Suppose that the decision maker has selected a solution on the 77-th subproblem as $\boldsymbol{x}^{\mathrm{final}}$ in Fig. 3 (b). In ADA, she/he can examine other dissimilar solutions with similar quality to $\boldsymbol{x}^{\mathrm{final}}$. Let $\boldsymbol{A}^{\mathrm{secondary}}$ be a set of all solutions assigned to the same 77-th subproblem with $\boldsymbol{x}^{\mathrm{final}}$. Figs. 3 (c) and (d) show the distribution of all nine solutions in $\boldsymbol{A}^{\mathrm{secondary}}$ in the objective and solution spaces, respectively. Although the nine solutions in $\boldsymbol{A}^{\mathrm{secondary}}$ have almost the same quality in the objective space, they are dissimilar in the solution space. In addition to $\boldsymbol{x}^{\mathrm{final}}$, the decision maker can examine other candidates in $\boldsymbol{A}^{\mathrm{secondary}}$ based on her/his preference in the solution space.

H. Two Post-processing methods for benchmarking

Apart from the practical two-phase decision making method described in Subsection III-G, we here consider benchmarking of ADA-based EMMAs. The performance of EMOAs and EMMAs is evaluated using performance indicators. However, a fair performance comparison is difficult between ADA-based EMMAs and other optimizers. This is because the current population $\boldsymbol{P}$ in ADA can contain an unbounded number of solutions. Most performance indicators cannot assess solution sets with different sizes in a fair manner [41]. IGD$^+$ [42] and IGDX [32] described in Section IV later are not exceptions. Thus, ADA requires a method of selecting a constant number of solutions from $\boldsymbol{P}$ for benchmarking studies.

We introduce two post-processing methods that are used to evaluate the performance of ADA-based algorithms for MOPs and MMOPs, respectively. In general, uniformly distributed solutions are unlikely to be uniformly distributed objective vectors due to a non-uniform mapping from the solution space to the objective space. To address this issue, we use the two post-processing methods for the objective and solution spaces, respectively. On the one hand, the same method of selecting $\boldsymbol{A}^{\mathrm{primary}}$ from $\boldsymbol{P}$ described in Subsection III-G is used for performance indicators of MOPs (e.g., IGD [43]). Since indicators of MOPs assess the distribution of a solution set in the objective space, the choice of $\boldsymbol{A}^{\mathrm{primary}}$ is reasonable.

On the other hand, the solution distance-based selection method presented in [17] is used for performance indicators for MMOPs (e.g., IGDX [32]). Let us consider the task of selecting N sparsely distributed solutions from all non-dominated solutions in $\boldsymbol{P}$. Below, $D(\boldsymbol{x}, \boldsymbol{A})$ denotes the distance between a solution $\boldsymbol{x}$ and its nearest solution in a solution set $\boldsymbol{A}$ in the normalized solution space. First, $\boldsymbol{A}^{\mathrm{tertiary}}$ is set to empty. A solution is randomly selected from the non-dominated solution set and stored into $\boldsymbol{A}^{\mathrm{tertiary}}$. Then, a solution with the maximum $D(\boldsymbol{x}, \boldsymbol{A}^{\mathrm{tertiary}})$ value is repeatedly added to $\boldsymbol{A}^{\mathrm{tertiary}}$ until $|\boldsymbol{A}^{\mathrm{tertiary}}| = N$. Unlike $\boldsymbol{A}^{\mathrm{primary}}$ and $\boldsymbol{A}^{\mathrm{secondary}}$, $\boldsymbol{A}^{\mathrm{tertiary}}$ is used only for benchmarking of ADA-based algorithms.

I. Applicability of ADA

ADA is a framework to improve the performance of decomposition-based EMOAs for MMOPs. We do not claim that ADA can be combined into any decomposition-based EMOAs. Since ADA requires a method of assigning a child $\boldsymbol{u}$ to a subproblem, ADA is inapplicable to EMOAs with no assignment mechanism. Such EMOAs include MOEA/D

[4] and MOEA/D-DRA [44]. Also, the deletion operation performs the pairwise comparison independently from other individuals. Thus, ADA is not applicable to EMOAs whose environmental selection is performed for all individuals in $\boldsymbol{P}$, such as MOEA/D-STM [45] and VaEA [30].

In summary, ADA can be combined into EMOAs with an assignment mechanism and a pairwise comparison-based environmental selection. The six EMOAs explained in Subsections II-C and II-D satisfy these conditions. In addition, ADA is applicable to I-DBEA [46] and SPEA/R [31].

Fortunately, even if a decomposition-based EMOA does not satisfy the above-mentioned two conditions, ADA can be applied to the EMOA after some modifications. For example, since MOEA/D does not have the assignment method, we cannot directly combine ADA in MOEA/D. However, MOEA/D can be easily modified by using any of the three assignment methods described in Subsection III-C. In fact, MOEA/D-AD proposed in [17] is an ADA-based MOEA/D with the assignment operation in (6). Thus, the applicability of ADA is not limited to only a few decomposition-based EMOAs.

J. Originality of ADA

In addition to MOEA/D-AD [17], a variant of MOEA/D for MMOPs is proposed in [47]. The MOEA/D variant assigns K individuals to each subproblem. The fitness value is based on the PBI function value and two distance values in the solution space. The main disadvantage of the MOEA/D variant in [47] is the difficulty in finding a proper K value. Since the number of equivalent Pareto optimal solution subsets is unknown a priori, fine-tuning of K is necessary for a given problem. Although a multi-start decomposition-based approach is proposed in [38], it has a similar disadvantage. In contrast, ADA adaptively adjusts the number of individuals assigned to each subproblem. Thus, ADA does not require a problem-dependent parameter such as K.

TriMOEA-TA&R [24] consists of various advanced components, including the convergence and diversity archives-based strategy as in Two_Arch2 [48], the decision variable-decomposition method in MOEA/DVA [49], and the angle-based individual assignment in RVEA [22]. ADA and TriMOEA-TA&R are similar in that they handle diversity in the objective and solution spaces in a two-phase manner. While TriMOEA-TA&R uses an *absolute* distance-based neighborhood criterion with σ_{niche}, ADA uses the *relative* distance-based neighborhood criterion with L. ADA uses a much simpler mating scheme, and it does not use any decomposition method of decision variables. ADA also uses the adaptive population sizing strategy to handle equivalent solutions, as mentioned above. Whereas TriMOEA-TA&R is an algorithm for MMOPs with distance-related variables, ADA is a general framework for various MMOPs.

IV. EXPERIMENTAL SETTINGS

A. Test problems

We use the following eight multi-modal multi-objective test problems: the Two-On-One problem [50], the three SYM-PART problems [38], the two SSUF problems [51], the Polygon problem [52], and the Omni-test problem [13]. Table I

TABLE I: Properties of multi-modal multi-objective test problems, where M, D, and O denote the number of objectives, design variables, and equivalent Pareto optimal solution subsets, respectively.

Test problems	M	D	O
Two-On-One [50] and SSUF1,3 [51]	2	2	2
SYM-PART1–3 [38]	2	2	9
Polygon [52]	Any	2	Any
Omni-test [13]	2	Any	3^D

shows their properties, including the number of objectives M, the number of decision variables D, and the number of equivalent Pareto optimal solution sets O.

Two-On-One has two equivalent Pareto optimal solution sets that are symmetrical with respect to the origin. Equivalent Pareto optimal solution sets are on the nine lines in SYM-PART1, as shown in Fig. 3. The nine lines are rotated in SYM-PART2. In addition, the nine lines are distorted in SYM-PART3. SSUF1 and SSUF3 have two symmetrical Pareto optimal solution sets. We evaluate the scalability of EMMAs to M using Polygon. We set M of Polygon as follows: $M \in \{3, 5, 8, 10\}$. Although O can be any number in Polygon, it was set to be nine. We investigate the scalability of EMMAs to D and O using Omni-test. We set D as follows: $D \in \{2, 3, 5, 8, 10\}$. O increases exponentially with increased D in Omni-test. In summary, we use 15 test problem instances.

HPS [23] and MMMOP [24] have been recently proposed. However, HPS and MMMOP have so-called "distance-related" variables that affect only the distance between the objective vector and the Pareto front. For this reason, we do not mainly use HPS and MMMOP for our benchmarking study. We use HPS and MMMOP only in Subsections V-E and V-F.

B. Performance indicators

Below, $\boldsymbol{A}$ is a set of solutions obtained by an EMMA. $\boldsymbol{A}^*$ is also a set of reference solutions in the Pareto optimal solution set. The size of $\boldsymbol{A}^*$ was set to $5\,000$. For $\boldsymbol{A}^*$ of each problem, $5\,000$ solutions were selected from randomly generated $10\,000$ Pareto-optimal solutions using the distance-based solution selection method [17] (Subsection III-H).

We use IGD$^+$ [42] to evaluate $\boldsymbol{A}$ in terms of both convergence to the Pareto front and diversity in the objective space:

$$\text{IGD}^+(\boldsymbol{A}) = \frac{1}{|\boldsymbol{A}^*|}\left(\sum_{\boldsymbol{z}\in\boldsymbol{A}^*}\min_{\boldsymbol{x}\in\boldsymbol{A}}\left\{d\big(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{z})\big)\right\}\right), \quad (14)$$

where $d(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\sum_{i=1}^{M}\big(\max\{a_i - b_i, 0\}\big)^2}$. IGD$^+$ is a modified version of IGD [43]. While the original IGD is Pareto non-compliant, IGD$^+$ is weakly Pareto compliant.

We evaluate how well $\boldsymbol{A}$ approximates the Pareto-optimal solution set in the solution space using IGDX [32]:

$$\text{IGDX}(\boldsymbol{A}) = \frac{1}{|\boldsymbol{A}^*|}\left(\sum_{\boldsymbol{z}\in\boldsymbol{A}^*}\min_{\boldsymbol{x}\in\boldsymbol{A}}\left\{\text{ED}(\boldsymbol{x}, \boldsymbol{z})\right\}\right), \quad (15)$$

where $\text{ED}(\boldsymbol{a}, \boldsymbol{b})$ is the Euclidean distance between $\boldsymbol{a}$ and $\boldsymbol{b}$.

EMOAs that can find $\boldsymbol{A}$ with small IGD$^+$ and IGDX values are efficient multi-objective optimizers and multi-modal multi-objective optimizers, respectively. In the ADA-based algorithms, $\boldsymbol{A}^{\mathrm{primary}}$ and $\boldsymbol{A}^{\mathrm{tertiary}}$ explained in Subsection III-H are used for the IGD$^+$ and IGDX calculations, respectively.

C. Average performance score

We use the average performance score (APS) [53] in order to aggregate results on various problems. Suppose that n algorithms $A_1, ..., A_n$ are compared for a problem instance based on the indicator values obtained in multiple runs. For each $i \in \{1, ..., n\}$ and $j \in \{1, ..., n\} \setminus \{i\}$, if A_j significantly outperforms A_i using the Wilcoxon rank-sum test with $p < 0.001$, then $\delta_{i,j} = 1$; otherwise, $\delta_{i,j} = 0$. The score $P(A_i)$ is defined as follows: $P(A_i) = \sum_{j \in \{1,...,n\} \setminus \{i\}}^{n} \delta_{i,j}$. The score $P(A_i)$ represents the number of algorithms that outperform A_i. The APS value of A_i is the average of the $P(A_i)$ values for all problem instances. A small APS value of A_i indicates that A_i performs well among n algorithms.

D. EMMAs and EMOAs

We examine the performance of the six ADA-based EMMAs: MOEA/D-AGR-ADA, MOEA/D-DU-ADA, eMOEA/D-ADA, NSGA-III-ADA, θ-DEA-ADA, and RVEA-ADA. We implemented the ADA-based algorithms using jMetal [54]. Their source codes can be downloaded from the supplementary website (https://sites.google.com/view/mmoada). We compare the ADA-based EMMAs to their original EMOAs: MOEA/D-AGR [19], MOEA/D-DU [20], eMOEA/D [21], NSGA-III [5], θ-DEA [18], and RVEA [22]. Our implementations of the six original EMOAs were based on their corresponding articles, except for MOEA/D-AGR. We replaced the DE operator with SBX in MOEA/D-AGR to remove the effect of variation operators. For details, refer to the corresponding articles.

The number of maximum function evaluations was $30\,000$. 31 runs were performed for each test problem. A set of weight/reference vectors were generated using the simplex-lattice design method for $M < 6$ and its two-layered version [5] for $M \geq 6$. The number of the weight/reference vectors N was 100, 105, 210, 156, and 230, for $M = 2, 3, 5, 8$, and 10, respectively. The SBX crossover and the polynomial mutation were used in all methods, including MOEA/D-AGR. Their control parameters were set as follows: $p_c = 1$, $\eta_c = 20$, $p_m = 1/D$, and $\eta_m = 20$. According to the analysis presented in [17], L of the neighborhood criterion in ADA was set to $L = \lfloor 0.1\mu \rfloor$. For example, $L = 201$ when $\mu = 2\,019$. Other parameters were set according to the corresponding references.

V. Experimental Results

This section shows performance analysis of the six ADA-based EMMAs. Subsection V-A describes the effect of ADA on the six EMOAs. Subsection V-B compares the six ADA-based EMMAs to state-of-the-art EMMAs. Subsection V-C discusses the adaptive population sizing in ADA. Subsection V-D investigates the influence of the L value on the performance of the six ADA-based EMMAs. Subsection V-E

TABLE II: Results of the six EMOAs and their ADA versions on the 15 test problem instances. Tables (a) and (b) show the APS values of the algorithms for IGD$^+$ and IGDX, respectively. AGR and DU stand for MOEA/D-AGR and MOEA/D-DU, respectively.

(a) IGD$^+$

	Orig.	ADA
AGR-ADA	0.0 (1)	1.0 (2)
DU-ADA	0.0 (1)	0.8 (2)
eMOEA/D-ADA	0.2 (1)	0.7 (2)
NSGA-III-ADA	0.0 (1)	0.8 (2)
θ-DEA-ADA	0.1 (1)	0.6 (2)
RVEA-ADA	0.5 (2)	0.3 (1)

(b) IGDX

	Orig.	ADA
AGR-ADA	1.0 (2)	0.0 (1)
DU-ADA	0.9 (2)	0.0 (1)
eMOEA/D-ADA	1.0 (2)	0.0 (1)
NSGA-III-ADA	1.0 (2)	0.0 (1)
θ-DEA-ADA	1.0 (2)	0.0 (1)
RVEA-ADA	1.0 (2)	0.0 (1)

examines the performance of the six ADA-based EMMAs on test problems with distance-related variables. Subsection V-F presents a comparison with the state-of-the-art EMMAs using an unbounded external archive. Subsection V-G discusses the runtime of the ADA-based algorithms.

A. Effect of ADA

Tables II (a) and (b) show the paired comparison of each EMOA and its ADA version on the 15 test problem instances in terms of IGD$^+$ and IGDX, respectively. Table II shows only the APS value of each algorithm at the final iteration. Table S.1 in the supplementary file shows detailed results. As described in Subsection III-H, only N solutions in $\boldsymbol{A}^{\mathrm{primary}}$ and $\boldsymbol{A}^{\mathrm{tertiary}}$ were used for the IGD$^+$ and IGDX calculations. Thus, all algorithms are compared under the same number of solutions. We set the p value to 0.001 for the Wilcoxon rank-sum test. M-Polygon is the M-objective Polygon problem, and D-Omni-test is the D-variable Omni-test problem. Below, we describe the results of IGD$^+$ and IGDX.

1) IGD$^+$: Table II (a) shows that the original EMOAs outperform their ADA-based EMMAs regarding IGD$^+$ (except for RVEA-ADA). In general, EMMAs perform slightly worse than EMOAs for multi-objective optimization [15], [16]. While EMOAs aim to find a good approximation of the Pareto front, EMMAs attempt to locate all Pareto optimal solutions. For this reason, the performance of the ADA versions regarding IGD$^+$ is worse than that of their original EMOAs.

However, as shown in Table S.1 in the supplementary file, the IGD$^+$ value of the ADA versions is only 3.57 times worse than that of their original EMOAs even in the worst case. Also, the ADA versions perform better than their original EMOAs on some problems. For example, eMOEA/D-ADA obtains better IGD$^+$ values than eMOEA/D on SYM-PART2, SSUF3, and 10-Polygon. As discussed in [55], a mechanism for maintaining the solution space diversity can help the ADA versions to find high-quality solutions.

2) IGDX: Table II (b) shows that the ADA-based EMMAs significantly outperform their original EMOAs. As shown in Table S.1, the IGDX value of the ADA versions is 141.58 times better than that of their original EMOAs in the best case. These results indicate that ADA improves the performance of the six decomposition-based EMOAs for MMOPs.

Table S.1 shows that the ADA-based EMMAs work well on Polygon with $M \in \{3, 5, 8, 10\}$ and Omni-test with $D \in \{2, 3, 5, 8, 10\}$. Since the number of equivalent Pareto
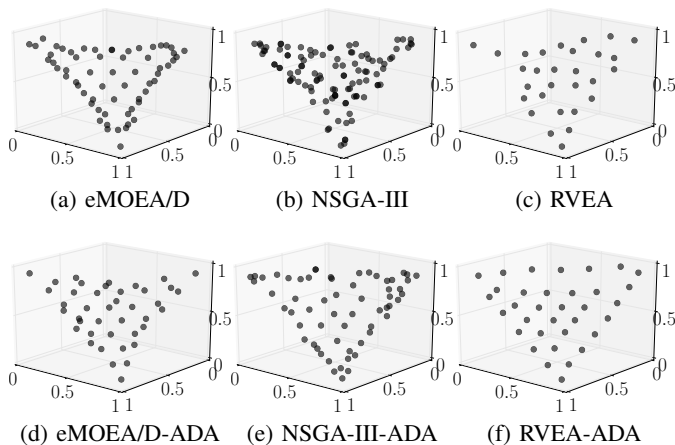
Fig. 4: Distribution of non-dominated solutions found by each method in the objective space on the three-objective Polygon problem. The x, y, and z axes represent f_1, f_2, and f_3, respectively.
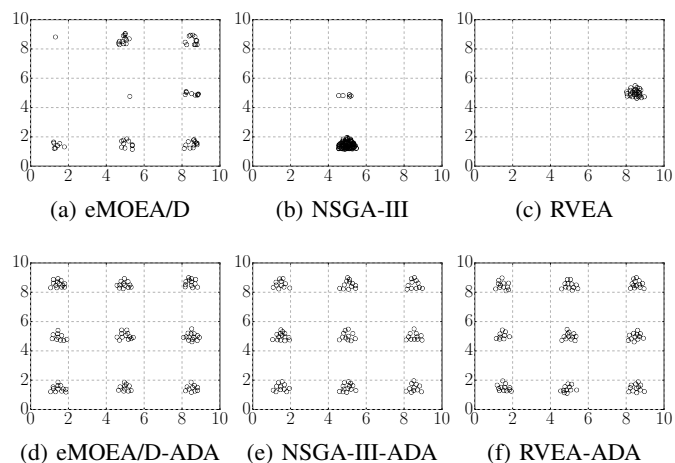

Fig. 5: Distribution of non-dominated solutions found by each method in the solution space on the three-objective Polygon problem. The x and y axes represent x_1 and x_2, respectively.

optimal solution sets O increases exponentially with increased D in Omni-test, the results show that ADA can also handle problems with a large O value. In summary, ADA has a sufficient scalability to M, D, and O.

3) Distribution of solutions: Figs. 4 and 5 show the distribution of non-dominated solutions found by eMOEA/D-ADA, NSGA-III-ADA, RVEA-ADA, and their original versions on Polygon with $M = 3$ in the objective and solution spaces, respectively. These figures show results of a single run with a median IGD$^+$ and IGDX values, respectively. For the ADA-based algorithms, non-dominated solutions in $\boldsymbol{A}^{\mathrm{primary}}$ and $\boldsymbol{A}^{\mathrm{tertiary}}$ are shown in Figs. 4 and 5, respectively. Figs. S.1 and S.2 in the supplementary file show the results of the other ADA-based algorithms, but the results are similar to Figs. 4 and 5. It was shown in [5] that some decomposition-based EMOAs did not work well on problems with convex Pareto fronts. It was also presented in [56] that some decomposition-based EMOAs performed poorly on problems with inverted-triangular Pareto fronts. Since Polygon has a convex and an in-

TABLE III: Results of the nine EMMAs on the 15 test problem instances. Tables (a) and (b) show the APS values of the algorithms for the IGD$^+$ and IGDX indicators, respectively. The numbers in parentheses are the ranks of the algorithms based on their APS values.

(a) IGD$^+$

MOEA/D-AGR-ADA	3.7 (7)
MOEA/D-DU-ADA	2.8 (5)
eMOEA/D-ADA	2.9 (6)
NSGA-III-ADA	1.7 (2)
θ-DEA-ADA	2.7 (4)
RVEA-ADA	5.9 (9)
TriMOEA-TA&R	2.4 (3)
MO_Ring_PSO_SCD	4.2 (8)
Omni-optimizer	1.6 (1)

(b) IGDX

MOEA/D-AGR-ADA	2.1 (3)
MOEA/D-DU-ADA	2.1 (4)
eMOEA/D-ADA	2.3 (5)
NSGA-III-ADA	0.8 (1)
θ-DEA-ADA	1.0 (2)
RVEA-ADA	4.1 (6)
TriMOEA-TA&R	7.7 (9)
MO_Ring_PSO_SCD	4.3 (7)
Omni-optimizer	5.1 (8)

verted triangular Pareto front, it is difficult for decomposition-based EMOAs to find good solutions.

For the above-mentioned reasons, no method in Fig. 4 can approximate the Pareto front of Polygon well. Nevertheless, some ADA-based algorithms (e.g., eMOEA/D-ADA and RVEA-ADA) find better distributed objective vectors than their original EMOAs. This unintended effect of ADA may be caused by the diversity of the population in the solution space [55]. Although we do not claim that ADA can improve the performance of EMOAs for MOPs in addition to MMOPs, the solution space diversity maintenance might help the original EMOAs to handle problems with irregular Pareto fronts.

In the Polygon problem with $M = 3$ and $O = 9$, equivalent Pareto solution sets are inside of the nine regular triangles in the solution space. Figs. 5 (a)–(c) show that the original EMOAs cannot locate all equivalent Pareto solution sets well. In contrast, Figs. 5 (d)–(f) demonstrate that their ADA versions can locate all nine equivalent Pareto solution sets. Results on other test problems are similar to Fig. 5.

B. Comparison with other EMMAs

In Subsection V-A, we demonstrated that ADA can improve the performance of the six EMOAs for MMOPs. Here, we compare the six ADA-based algorithms to the following three EMMAs: Omni-optimizer [13], MO_Ring_PSO_SCD [16], and TriMOEA-TA&R [24]. Omni-optimizer is the most representative EMMA. MO_Ring_PSO_SCD and TriMOEA-TA&R are recently proposed methods. Default parameter settings were used for the three methods.

Table III shows the APS values of the nine EMMAs on the 15 test problem instances. Table S.2 in the supplementary file shows detailed results. TriMOEA-TA&R incorrectly recognizes that some problems have distance-related variables (e.g., SYM-PART1). In such a case, TriMOEA-TA&R generates a set of additional solutions $\boldsymbol{A}^{\mathrm{add}}$ by recombining solutions in the diversity archive and the distance-related variables at the end of the search. Since $|\boldsymbol{A}^{\mathrm{add}}| > N$, methods of selecting N solutions from $\boldsymbol{A}^{\mathrm{add}}$ are needed. We use the two post-processing methods in ADA. In the same manner as in ADA, $\boldsymbol{A}^{\mathrm{primary}}$ and $\boldsymbol{A}^{\mathrm{tertiary}}$ are used for the IGD$^+$ and IGDX calculations, respectively. We use the selection method in NSGA-III-ADA to obtain $\boldsymbol{A}^{\mathrm{primary}}$ from $\boldsymbol{A}^{\mathrm{add}}$.

Table III (a) shows that Omni-optimizer performs the best regarding IGD$^+$. The good performance of Omni-optimizer
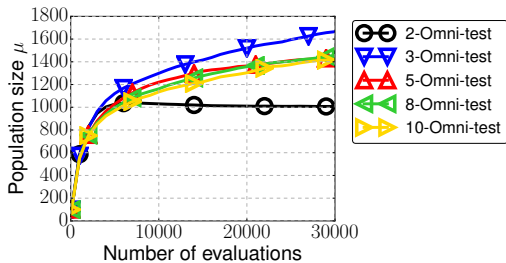
Fig. 6: Evolution of μ of NSGA-III-ADA on Omni-test with $D \in \{2, 3, 5, 8, 10\}$. The mean μ values over 31 runs are reported.
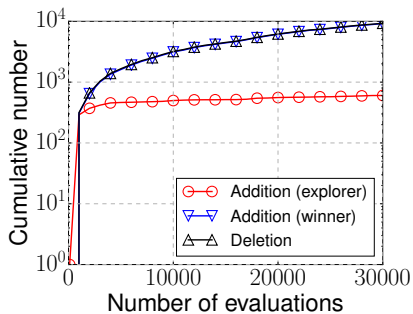

Fig. 7: Cumulative number of the activations of the addition operation based on the two criteria (b^{explorer} and b^{winner}) and the deletion operation in NSGA-III-ADA on 3-Polygon ($M = 3$). Results of a single run with a median IGDX value among 31 runs are shown.

for MOPs is consistent with the results presented in [16], [17]. NSGA-III-ADA shows the best performance regarding IGD$^+$ among the six ADA-based algorithms. Table III (b) shows that the six ADA-based algorithms perform better than the three other algorithms regarding IGDX. TriMOEA-TA&R performs the worst regarding IGDX. This is because the 15 problem instances have no distance-related variables. In summary, the results indicate that the six ADA-based algorithms have better performance than the state-of-the-art algorithms for MMOPs.

C. Adaptive population sizing in ADA

In ADA, the number of individuals assigned to each subproblem is adaptively adjusted by the deletion and addition operations. Thus, the population size μ is not constant. Here, we discuss the adaptive population sizing in ADA.

Fig. 6 shows the change of μ of NSGA-III-ADA on Omni-test with $D \in \{2, 3, 5, 8, 10\}$. Results of the other ADA versions are similar to Fig. 6. The μ value for $D = 3$ is always larger than that for $D = 2$. In contrast, the μ value decreases as the D value increases from $D = 3$. This may be because problems with a large D value are generally difficult for EMMAs. Since the search does not proceed well for $D \in \{5, 8, 10\}$, the μ value does not significantly increase. The difficulty of finding multiple equivalent Pareto sets has been reported for the case of a large D value [57].

In addition, the trajectory of μ is problem-dependent. Fig. S.3 in the supplementary file shows the change of μ on other problems. While the μ value is approximately 640 on Two-On-One and 3-Polygon at the end of the search, it is approximately 960–1 090 on other problems. Fig. S.3 shows that the μ value

sharply increases in early iterations. After some evaluations, the μ value is stable.

Fig. 7 shows the cumulative numbers of the activations of the addition operation based on the two criteria (b^{explorer} and b^{winner} in Algorithm 1) and the deletion operation in NSGA-III-ADA on 3-Polygon. On the one hand, the number of activations of the addition operation based on b^{winner} is almost the same as that of the deletion operation throughout the evolution in Fig. 7. This competitive behavior of the two operations leads to the stable μ value in Fig. S.3. On the other hand, the addition operation based on b^{explorer} is frequently performed only in an early stage of evolution in Fig. 7. Thus, the sharp increase of the μ value in Fig. S.3 is due to the b^{explorer}-based addition operation. Since only one individual is assigned to each subproblem at the beginning of the search, the b^{explorer}-based addition operation is often activated.

In summary, the μ value is adaptively adjusted by the cooperative behavior of the deletion and addition operations in ADA. In the addition operation, the roles of the two criteria b^{explorer} and b^{winner} differ from each other. Thus, it is important for ADA to use both b^{explorer} and b^{winner}.

D. Impact of L on the six ADA-based algorithms

We investigate the influence of the L value on the performance of the ADA-based algorithms. Table IV shows the APS values of the six ADA-based algorithms with six L values on the 15 test problem instances. Tables S.3–S.8 in the supplementary file show detailed results.

Table IV (a) shows that the six ADA-based algorithms with $L = \lfloor 0.05\mu \rfloor$ have the worst performance regarding IGD$^+$. Too small L values degrade the performance of the ADA-based algorithms regarding IGD$^+$. Table IV (b) shows that $L = \lfloor 0.1\mu \rfloor$ is most suitable for all ADA-based algorithms (except for RVEA-ADA) in terms of IGDX. Too large L values degrade the performance of the ADA-based algorithms regarding IGDX. In summary, $L = \lfloor 0.1\mu \rfloor$ are suitable for most ADA-based algorithms for MMOPs.

We also investigate the influence of L on the evolution of μ. Figs. S.4 and S.5 in the supplementary file show the change of μ of NSGA-III-ADA with various L values on Omni-test with various D values and other problems, respectively. Results of other ADA versions are similar to the results of NSGA-III-ADA. Figs. S.4 and S.5 indicate that the μ value decreases as the L value increases. The larger the L value is, the more individuals can be neighbors of the child $\boldsymbol{u}$. As a result, the niching mechanism in ADA deteriorates. These observations are consistent with the above-mentioned results of IGDX.

E. Results on test problems with distance-related variables

This subsection shows results on the HPS and MMMOP test problems. Although the six HPS problem instances (HPS1, ..., HPS6) are proposed in [23], we use only HPS2. This is because all the other five problem instances are variants of HPS2 and also because the details of only HPS2 are provided in [23]. M, D, and O in HPS2 are as follows: $M = 2$, $D = 7$, and $O = 4$. The 20 MMMOP problem instances (MMMOP1A, ..., MMMOP6D) are proposed in [24]. We also use those 20

TABLE IV: Results of the six ADA-based algorithms with $L \in \{\lfloor 0.05\mu \rfloor, \lfloor 0.1\mu \rfloor, \lfloor 0.2\mu \rfloor, \lfloor 0.3\mu \rfloor, \lfloor 0.4\mu \rfloor, \lfloor 0.5\mu \rfloor\}$ on the 15 test problem instances. Tables (a) and (b) show the APS values of the algorithms for the IGD$^+$ and IGDX indicators, respectively. AGR and DU stand for MOEA/D-AGR and MOEA/D-DU, respectively.

(a) IGD$^+$

	$\lfloor 0.05\mu \rfloor$	$\lfloor 0.1\mu \rfloor$	$\lfloor 0.2\mu \rfloor$	$\lfloor 0.3\mu \rfloor$	$\lfloor 0.4\mu \rfloor$	$\lfloor 0.5\mu \rfloor$
AGR-ADA	1.6 (6)	0.9 (3)	0.6 (1)	0.7 (2)	1.0 (4)	1.3 (5)
DU-ADA	1.6 (6)	0.7 (2)	0.5 (1)	0.7 (3)	1.0 (4)	1.3 (5)
eMOEA/D-ADA	1.7 (6)	0.9 (3)	0.5 (1)	0.5 (2)	0.9 (4)	1.1 (5)
NSGA-III-ADA	2.6 (6)	1.5 (5)	0.7 (4)	0.1 (1)	0.1 (1)	0.2 (3)
θ-DEA-ADA	2.2 (6)	1.3 (5)	0.4 (4)	0.2 (3)	0.0 (1)	0.0 (1)
RVEA-ADA	1.9 (6)	0.5 (1)	1.1 (3)	0.9 (2)	1.3 (4)	1.9 (5)

(b) IGDX

	$\lfloor 0.05\mu \rfloor$	$\lfloor 0.1\mu \rfloor$	$\lfloor 0.2\mu \rfloor$	$\lfloor 0.3\mu \rfloor$	$\lfloor 0.4\mu \rfloor$	$\lfloor 0.5\mu \rfloor$
AGR-ADA	0.8 (2)	0.1 (1)	0.8 (3)	2.6 (4)	2.9 (5)	3.2 (6)
DU-ADA	0.8 (2)	0.3 (1)	1.0 (3)	2.5 (4)	2.9 (5)	3.5 (6)
eMOEA/D-ADA	0.9 (2)	0.3 (1)	1.1 (3)	2.2 (4)	2.5 (5)	3.2 (6)
NSGA-III-ADA	1.0 (3)	0.1 (1)	0.5 (2)	2.2 (4)	2.7 (5)	3.5 (6)
θ-DEA-ADA	1.1 (3)	0.2 (1)	0.9 (2)	2.1 (4)	2.7 (5)	3.2 (6)
RVEA-ADA	0.5 (1)	0.6 (2)	1.6 (3)	2.7 (4)	2.9 (5)	3.1 (6)

TABLE V: Results of the nine EMMAs on the 21 test problem instances with distance-related variables: APS value of each algorithm for IGD+ in (a) and for IGDX in (b).

(a) IGD$^+$

MOEA/D-AGR-ADA	3.0 (5)
MOEA/D-DU-ADA	2.2 (3)
eMOEA/D-ADA	1.5 (2)
NSGA-III-ADA	4.2 (7)
θ-DEA-ADA	3.0 (6)
RVEA-ADA	4.4 (8)
TriMOEA-TA&R	1.0 (1)
MO_Ring_PSO_SCD	6.8 (9)
Omni-optimizer	2.3 (4)

(b) IGDX

MOEA/D-AGR-ADA	2.0 (3)
MOEA/D-DU-ADA	1.5 (2)
eMOEA/D-ADA	0.9 (1)
NSGA-III-ADA	3.0 (6)
θ-DEA-ADA	2.2 (4)
RVEA-ADA	3.7 (7)
TriMOEA-TA&R	2.7 (5)
MO_Ring_PSO_SCD	6.9 (9)
Omni-optimizer	4.1 (8)

instances. M, D, and O in MMMOP are as follows: $M \in \{2, 3\}$, $D \in \{2, ..., 7\}$, and $O \in \{2, ..., 9\}$.

Table S.9 in the supplementary file shows results of the six EMOAs and their ADA versions on the 21 problem instances. The results in Table S.9 are similar to the results in Table II. Since the solution space diversity maintenance mechanism in ADA is not useful to handle distance-related variables, the ADA-based algorithms are likely to perform poorly on the problems with distance-related variables. However, the results show that the ADA-based algorithms perform better than their original EMOAs (except for NSGA-III-ADA).

Table V shows the APS values of the nine EMMAs on the 21 problem instances. Table S.10 in the supplementary file shows detailed results. Since TriMOEA-TA&R can explicitly exploit distance-related variables for the search similar to MOEA/DVA [49], its performance is improved on the 21 problem instances. Table V (a) shows that TriMOEA-TA&R performs the best regarding IGD$^+$. Table V (b) shows that TriMOEA-TA&R is the fifth-ranked algorithm regarding IGDX. MOEA/D-AGR-ADA, MOEA/D-DU-ADA, eMOEA/D-ADA, and θ-DEA-ADA perform better than the three competitors regarding IGDX. In summary, the ADA-based algorithms have high performance for MMOPs even on the problems with distance-related variables.

F. Comparison using an unbounded external archive

While the three EMMAs (TriMOEA-TA&R, MO_Ring_PSO_SCD, and Omni-optimizer) keep the μ value constant, the ADA-based algorithms can adaptively adjust the μ value. One may think that the comparisons in Subsections V-B and V-E are unfair for this reason. Here, we compare the ADA-based algorithms to the three EMMAs using an unbounded external archive (UEA) [58], which stores all non-dominated solutions found during the search process. The UEA can be incorporated into any algorithms with no changes in their algorithmic behavior. The three EMMAs with the UEA can maintain all non-dominated solutions found so far. Note that only the three EMMAs use the UEA in this section. Since an algorithm with the UEA significantly outperforms its original version (see [58]), such a comparison is unfair for the ADA-based algorithms.

Tables S.11 and S.12 in the supplementary file show the comparisons with the three EMMAs using the UEA on the 15 test problem instances with no distance-related variables and the 21 test problem instances with distance-related variables, respectively. The performance of the three EMMAs is improved by using the UEA. However, Table S.11 indicates that all ADA-based algorithms (except for RVEA-ADA) outperform the three EMMAs with the UEA on the 15 test problems in terms of IGDX. Table S.12 also shows that eMOEA/D-ADA performs the best on the 21 test problems in terms of IGDX. Thus, the results show that some ADA-based algorithms perform significantly better than the three EMMAs even with the UEA. Note that the performance of the ADA-based algorithms can be further improved by using the UEA as in the three EMMAs demonstrated here.

G. On the runtime of the ADA-based algorithms

The space complexity of ADA itself is $\mathcal{O}(\mu D)$ if $D > M$. Otherwise, it is $\mathcal{O}(\mu M)$. Although the μ value equals to the maximum number of function evaluations n^{max} in ADA in the worst case, the μ value is naturally bounded as shown in Subsection V-C. Thus, the empirical space complexity is much smaller than $\mathcal{O}(n^{\mathrm{max}}D)$ and $\mathcal{O}(n^{\mathrm{max}}M)$.

Below, we explain the worst-case time complexity of the ADA-based algorithms. ADA itself in one iteration requires $\mathcal{O}(\mu D)$, which is due to the procedure of selecting the neighbors of the child $\boldsymbol{u}$ in the solution space (line 11 in Algorithm 1). The time complexity of an ADA-based algorithm depends on its original version. Since the time complexity of NSGA-III is the larger value between $\mathcal{O}(\mu^2 \log^{M-2}\mu)$ and $\mathcal{O}(\mu^2 M)$ [5], that of NSGA-III-ADA is the largest value among $\mathcal{O}(\mu^2 \log^{M-2}\mu)$, $\mathcal{O}(\mu^2 M)$, and $\mathcal{O}(\mu D)$. Similarly, the time complexity of θ-DEA-ADA and RVEA-ADA is the larger value between $\mathcal{O}(\mu^2 M)$ [18], [22] and $\mathcal{O}(\mu D)$. The assignment operations in MOEA/D-AGR, MOEA/D-DU, and eMOEA/D in one function evaluation require $\mathcal{O}(\mu M)$ [19]–[21]. Thus, the time complexity of MOEA/D-AGR-ADA, MOEA/D-DU-ADA, and eMOEA/D-ADA in one iteration is the larger value between $\mathcal{O}(\mu M)$ and $\mathcal{O}(\mu D)$.

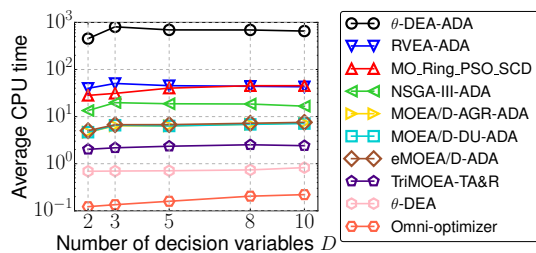Fig. 8 shows the average CPU time of each EMMA over the five Omni-test problems with $D = 2, 3, 5, 8, 10$. We

Fig. 8: Average CPU time of the EMMAs over the five Omni-test problems with $D = 2, 3, 5, 8, 10$.

obtained all results using a workstation with Xeon E5-2620 v4/2.10GHz and 8GB RAM. Fig. 8 also shows results of θ-DEA. The six ADA-based algorithms and θ-DEA are implemented in Java, Omni-optimizer is implemented in C, and MO_Ring_PSO_SCD and TriMOEA-TA&R are implemented in Matlab. These implementations are based on the available codes of these algorithms. The use of C and Matlab is due to the original implementations of these algorithms [13], [16], [24]. Since different programming languages require different CPU time, Fig. 8 provides rough comparison.

As shown in Fig. 8, Omni-optimizer is fastest, followed by θ-DEA and TriMOEA-TA&R. The six ADA-based algorithms are much slower than these three algorithms. This is because the μ value of the ADA-based algorithms is always larger than that of the other algorithms (see Subsection V-C). Since some operations require $\mathcal{O}(\mu^2 M)$ and $\mathcal{O}(\mu D)$ as discussed above, a larger μ value makes an algorithm slower. Although MO_Ring_PSO_SCD and RVEA-ADA perform similarly in Fig. 8, we believe that MO_Ring_PSO_SCD with an optimized source code can perform much faster than the ADA-based algorithms. Clearly, θ-DEA-ADA is slowest. This is because θ-DEA-ADA needs to normalize objective vectors for every iteration in a computationally expensive manner. Since the original θ-DEA is a generational EMOA, it can reduce the number of normalizations. In fact, θ-DEA is much faster than θ-DEA-ADA. Note that θ-DEA-ADA can be speeded up by using a simple normalization method (e.g., Subsection III-B).

As demonstrated here, the slow speed of the ADA-algorithms is their disadvantage. Speeding up the ADA-algorithms is an avenue for future work. However, this does not mean that ADA is impractical. Some real-world problems require a long computation time to evaluate a solution, e.g., by the execution of an expensive computer simulation [59]. In this case, the time of function evaluations dominates that of the other parts of optimization algorithms. For this reason, in the EMO community, the comparison is generally based on the number of function evaluations, rather than the CPU time.

VI. Conclusion

We proposed the ADA framework to improve the performance of decomposition-based EMOAs for MMOPs. ADA adaptively assigns one or more individuals to each subproblem to locate multiple equivalent Pareto optimal solutions. The population size μ is automatically adjusted during the search process. As presented in Subsection III-G, the effective

decision making can be performed using multiple equivalent solutions found by ADA. We incorporated ADA into the six decomposition-based EMOAs. We examined the performance of those six ADA-based algorithms on the 15 test problem instances and the 21 test problem instances with distance-related variables. The results show that ADA can improve the performance of the original EMOAs for MMOPs. We also analyzed the adaptive behavior of ADA.

Computational overhead in the ADA-based algorithms is their disadvantage (see Subsection V-G). Thus, the unbound population size of ADA is a "double-edged sword". However, as discussed in Subsection V-G, the computational overhead in ADA is practically acceptable in real-world applications where solution evaluations need dominant computation time in the whole optimization process.

For a fair comparison, we compared the ADA-based algorithms to the three EMMAs using the UEA in Subsection V-F. However, the UEA is inherently a post-processing method since an algorithm cannot take advantage of non-dominated solutions in the UEA during the search process as in the ADA-based algorithms. For this reason, the comparison with the three EMMAs using the UEA is not totally fair. A general adaptive population sizing framework which can be combined with any EMMA is needed for a totally fair comparison.

Although we focused on Type1-MMOPs, we believe that ADA is applicable to Type2-MMOPs using a relaxed dominance relation or a relaxed equivalent relation. An analysis of ADA on Type2-MMOPs is an avenue for future work. Other post-processing methods that handle the diversity in both the objective and solution spaces may be better than the two post-processing methods in ADA. Further analysis of post-processing methods is a future research topic.

References

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer, 1998.
[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE TEVC*, vol. 6, no. 2, pp. 182–197, 2002.
[3] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *PPSN*, 2004, pp. 832–842.
[4] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE TEVC*, vol. 11, no. 6, pp. 712–731, 2007.
[5] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE TEVC*, vol. 18, no. 4, pp. 577–601, 2014.
[6] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition," *IEEE TEVC*, vol. 21, no. 3, pp. 440–462, 2017.
[7] M. Sebag, N. Tarrisson, O. Teytaud, J. Lefèvre, and S. Baillet, "A

Multi-Objective Multi-Modal Optimization Approach for Mining Stable Spatio-Temporal Patterns," in *IJCAI*, 2005, pp. 859–864.

[8] G. Rudolph and M. Preuss, "A multiobjective approach for finding equivalent inverse images of pareto-optimal objective vectors," in *MCDM*, 2009, pp. 74–79.

[9] O. Schütze, M. Vasile, and C. A. C. Coello, "Computing the Set of Epsilon-Efficient Solutions in Multiobjective Space Mission Design," *JACIC*, vol. 8, no. 3, pp. 53–70, 2011.

[10] T. Hiroyasu, S. Nakayama, and M. Miki, "Comparison study of SPEA2+, SPEA2, and NSGA-II in diesel engine emissions and fuel economy problem," in *IEEE CEC*, 2005, pp. 236–242.

[11] M. Preuss, C. Kausch, C. Bouvy, and F. Henrich, "Decision Space Diversity Can Be Essential for Solving Multiobjective Real-World Problems," in *MCDM*, 2008, pp. 367–377.

[12] 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*, 2011, pp. 2558–2562.

[13] K. Deb and S. Tiwari, "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization," *EJOR*, vol. 185, no. 3, pp. 1062–1087, 2008.

[14] T. Ulrich, J. Bader, and L. Thiele, "Defining and Optimizing Indicator-Based Diversity Measures in Multiobjective Search," in *PPSN*, 2010, pp. 707–717.

[15] O. M. Shir, M. Preuss, B. Naujoks, and M. T. M. Emmerich, "Enhancing Decision Space Diversity in Evolutionary Multiobjective Algorithms," in *EMO*, 2009, pp. 95–109.

[16] C. Yue, B. Qu, and J. Liang, "A Multiobjective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multiobjective Problems," *IEEE TEVC*, vol. 22, no. 5, pp. 805–817, 2018.

[17] R. Tanabe and H. Ishibuchi, "A Decomposition-Based Evolutionary Algorithm for Multi-modal Multi-objective Optimization," in *PPSN*, 2018, pp. 249–261.

[18] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization," *IEEE TEVC*, vol. 20, no. 1, pp. 16–37, 2016.

[19] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive Replacement Strategies for MOEA/D," *IEEE Trans. Cyber.*, vol. 46, no. 2, pp. 474–486, 2016.

[20] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers," *IEEE TEVC*, vol. 20, no. 2, pp. 180–198, 2016.

[21] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing Functions in Decomposition-Based Multiobjective Evolutionary Algorithms," *IEEE TEVC*, vol. 22, no. 2, pp. 296–313, 2018.

[22] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization," *IEEE TEVC*, vol. 20, no. 5, pp. 773–791, 2016.

[23] B. Zhang, K. Shafi, and H. A. Abbass, "On Benchmark Problems and Metrics for Decision Space Performance Analysis in Multi-Objective Optimization," *IJCIA*, vol. 16, no. 1, pp. 1–18, 2017.

[24] Y. Liu, G. G. Yen, and D. Gong, "A Multi-Modal Multi-Objective Evolutionary Algorithm Using Two-Archive and Recombination Strategies," *IEEE TEVC*, 2018 (in press).

[25] K. Deb and S. Tiwari, "Omni-optimizer: A Procedure for Single and Multi-objective Optimization," in *EMO*, 2005, pp. 47–61.

[26] H. Li and Q. Zhang, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE TEVC*, vol. 13, no. 2, pp. 284–302, 2009.

[27] K. Deb and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex Systems*, vol. 9, no. 2, 1995.

[28] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *J. Global Optimiz.*, vol. 11, no. 4, pp. 341–359, 1997.

[29] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with Adaptive Weight Adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[30] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization," *IEEE TEVC*, vol. 21, no. 1, pp. 131–152, 2017.

[31] S. Jiang and S. Yang, "A Strength Pareto Evolutionary Algorithm Based on Reference Direction for Multiobjective and Many-Objective Optimization," *IEEE TEVC*, vol. 21, no. 3, pp. 329–346, 2017.

[32] 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*, vol. 13, no. 5, pp. 1167–1189, 2009.

[33] J. C. Castillo, C. Segura, A. H. Aguirre, G. Miranda, and C. León, "A multi-objective decomposition-based evolutionary algorithm with enhanced variable space diversity control," in *GECCO (Companion)*, 2017, pp. 1565–1571.

[34] 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*, vol. 21, no. 4, pp. 518–538, 2017.

[35] S. Bird and X. Li, "Adaptively choosing niching parameters in a PSO," in *GECCO*, 2006, pp. 3–10.

[36] M. Preuss, "Improved Topological Niching for Real-Valued Global Optimization," in *EvoApplications*, 2012, pp. 386–395.

[37] P. Kerschke, H. Wang, M. Preuss, C. Grimme, A. H. Deutz, H. Trautmann, and M. Emmerich, "Towards analyzing multimodality of continuous multiobjective landscapes," in *PPSN*, 2016, pp. 962–972.

[38] G. Rudolph, B. Naujoks, and M. Preuss, "Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets," in *EMO*, 2007, pp. 36–50.

[39] K. Bringmann, T. Friedrich, and P. Klitzke, "Two-dimensional subset selection for hypervolume and epsilon-indicator," in *GECCO*, 2014, pp. 589–596.

[40] H. K. Singh, K. S. Bhattacharjee, and T. Ray, "Distance based subset selection for benchmarking in evolutionary multi/many-objective optimization," *IEEE TEVC*, 2018 (in press).

[41] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "How to compare many-objective algorithms under different settings of population and archive sizes," in *IEEE CEC*, 2016, pp. 1149–1156.

[42] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified Distance Calculation in Generational Distance and Inverted Generational Distance," in *EMO*, 2015, pp. 110–125.

[43] C. A. C. Coello and M. R. Sierra, "A Study of the Parallelization of a Coevolutionary Multi-objective Evolutionary Algorithm," in *MICAI*, 2004, pp. 688–697.

[44] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *IEEE CEC*, 2009, pp. 203–208.

[45] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable Matching-Based Selection in Evolutionary Multiobjective Optimization," *IEEE TEVC*, vol. 18, no. 6, pp. 909–923, 2014.

[46] M. Asafuddoula, T. Ray, and R. A. Sarker, "A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization," *IEEE TEVC*, vol. 19, no. 3, pp. 445–460, 2015.

[47] C. Hu and H. Ishibuchi, "Incorporation of a decision space diversity maintenance mechanism into MOEA/D for multi-modal multi-objective optimization," in *GECCO (Companion)*, 2018, pp. 1898–1901.

[48] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization," *IEEE TEVC*, vol. 19, no. 4, pp. 524–541, 2015.

[49] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A Multiobjective Evolutionary Algorithm Based on Decision Variable Analyses for Multiobjective Optimization Problems With Large-Scale Variables," *IEEE TEVC*, vol. 20, no. 2, pp. 275–298, 2016.

[50] M. Preuss, B. Naujoks, and G. Rudolph, "Pareto Set and EMOA Behavior for Simple Multimodal Multiobjective Functions," in *PPSN*, 2006, pp. 513–522.

[51] J. J. Liang, C. T. Yue, and B. Y. Qu, "Multimodal multi-objective optimization: A preliminary study," in *IEEE CEC*, 2016, pp. 2454–2461.

[52] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Many-Objective Test Problems to Visually Examine the Behavior of Multiobjective Evolution in a Decision Space," in *PPSN*, 2010, pp. 91–100.

[53] J. Bader and E. Zitzler, "HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[54] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *AES*, vol. 42, no. 10, pp. 760–771, 2011.

[55] T. Ulrich, J. Bader, and E. Zitzler, "Integrating decision space diversity into hypervolume-based multiobjective search," in *GECCO*, 2010, pp. 455–462.

[56] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes," *IEEE TEVC*, vol. 21, no. 2, pp. 169–190, 2017.

[57] H. Ishibuchi, Y. Peng, and K. Shang, "A Scalable Multimodal Multiobjective Test Problem," in *IEEE CEC*, 2019, pp. 302–309.

[58] R. Tanabe, H. Ishibuchi, and A. Oyama, "Benchmarking Multi- and Many-objective Evolutionary Algorithms under Two Optimization Scenarios," *IEEE Access*, vol. 5, pp. 19 597–19 619, 2017.

[59] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Comput.*, vol. 23, no. 9, pp. 3137–3166, 2019.

Ryoji Tanabe is a Research Assistant Professor with Department of Computer Science and Engineering, Southern University of Science and Technology, China. He was a Post-Doctoral Researcher with ISAS/JAXA, Japan, from 2016 to 2017. He received his Ph.D. in Science from The University of Tokyo, Japan, in 2016. His research interests include stochastic single- and multi-objective optimization algorithms, parameter control in evolutionary algorithms, and automatic algorithm configuration.

Hisao Ishibuchi (M'93-SM'10-F'14) received the B.S. and M.S. degrees in precision mechanics from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from Osaka Prefecture University, Sakai, Osaka, Japan, in 1992. He was with Osaka Prefecture University in 1987-2017. Since 2017, he is a Chair Professor at Southern University of Science and Technology, China. His research interests include fuzzy rule-based classifiers, evolutionary multi-objective and many-objective optimization, memetic algorithms, and evolutionary games.

Dr. Ishibuchi was the IEEE Computational Intelligence Society (CIS) Vice-President for Technical Activities (2010-2013), an AdCom member of the IEEE CIS (2014-2019), and the Editor-in-Chief of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE (2014-2019).