Impact of Approximating Scalarization Functions on High-dimensional Multiobjective Optimization: A Fast and Scalable Approach

Masaya Nakata [1], Yuma Horaguchi [2], and Kei Nishihara [2]

[1]Yokohama National University
[2]Affiliation not available

October 30, 2023

Abstract

The approximation of objective functions is a major strategy in surrogate-assisted multi-objective evolutionary algorithms, but it tends to underperform on high-dimensional problems. We hypothesize that this is because the above strategy is vulnerable to unreliable approximations and even a single unreliable approximation model may mislead the entire search process. Therefore, an alternative strategy is to approximate each scalarization function, whereby candidate solutions for a decomposed problem can be evaluated using a single approximation model, which prevents the negative propagation of unreliable approximations to the entire search process. Accordingly, this study aims to confirm our hypothesis by introducing a basic surrogate-assisted algorithm, in which each approximated scalarization function is independently optimized by a differential evolution algorithm. Despite its methodological simplicity, the significant impact of approximating scalarization functions on high-dimensional problems is revealed for the first time. The presented algorithm is competitive with state-of-the-art algorithms that are adapted for high-dimensional problems, while exhibiting a reduced computational time. This computational efficiency is theoretically confirmed by our complexity analysis.

Impact of Approximating Scalarization Functions on High-dimensional Multiobjective Optimization: A Fast and Scalable Approach

Yuma Horaguchi, Kei Nishihara, and Masaya Nakata, *Member, IEEE,*

***Abstract*—The approximation of objective functions is a major strategy in surrogate-assisted multi-objective evolutionary algorithms, but it tends to underperform on high-dimensional problems. We hypothesize that this is because the above strategy is vulnerable to unreliable approximations and even a single unreliable approximation model may mislead the entire search process. Therefore, an alternative strategy is to approximate each scalarization function, whereby candidate solutions for a decomposed problem can be evaluated using a single approximation model, which prevents the negative propagation of unreliable approximations to the entire search process. Accordingly, this study aims to confirm our hypothesis by introducing a basic surrogate-assisted algorithm, in which each approximated scalarization function is independently optimized by a differential evolution algorithm. Despite its methodological simplicity, the significant impact of approximating scalarization functions on high-dimensional problems is revealed for the first time. The presented algorithm is competitive with state-of-the-art algorithms that are adapted for high-dimensional problems, while exhibiting a reduced computational time. This computational efficiency is theoretically confirmed by our complexity analysis.**

***Index Terms*—Surrogate-assisted evolutionary algorithm, multiobjective optimization, scalability, scalarization function.**

I. Introduction

MULTI-objective evolutionary algorithms (MOEAs) [1] are powerful black-box optimizers and their potential can be achieved with a sufficient number of fitness evaluations (FEs) [2]. However, in reality, the number of FEs is frequently restricted owing to their computational and/or financial cost, which is often referred to as an expensive multi-objective optimization problem (EMOP) [3], [4]. Moreover, the optimization of elaborate, large-scale models is generally performed using expensive simulations; thus, the high dimensionality of problems is a typical difficulty concomitant with EMOPs [5], [6]. For example, in car structure optimization [7], [8], a 222-dimensional car model is employed for optimization, wherein the evaluation of a single solution requires twenty hours using a supercomputer.

Surrogate-assisted evolutionary algorithms (SAEAs) [9] are a representative approach for accelerating the evolutionary search of MOEAs with thousands or even hundreds of FEs [10]. Surrogates are typically designed to estimate the quality of the solutions, and SAEAs use these to prescreen unevaluated solutions before their exact evaluations with expensive objective functions. Although various surrogate models exist, the use of approximation models for metric functions, namely approximation-based SAEAs, has become a central concept in the field [11]. A methodological benefit of approximation-based SAEAs is the production of well-prescreened candidate solutions by optimizing the approximation models, compared to, for example, classification models [12], [13].

The approximation of objective functions (AOF) is the most common strategy among various possible alternatives, such as the approximation of scalarization functions (ASF) [10], [11]. This is probably owing to its reasonability; approximated objective values are a fundamental resource for computing various metric values, such as scalarization functions [14]–[16], infill criteria [17]–[19], and certain indicators [20], [21]. Furthermore, the number of required models can be restricted to that of the objectives. These benefits are crucial in tackling current issues in EMOPs, that is, improving the scalability to the number of objectives [22] and reducing the runtime of SAEAs [11]. According to the observations reported in [23], well-known AOF-based SAEAs, namely MOEA/D-EGO [24] and K-RVEA [25], outperformed a representative ASF-based SAEA, namely ParEGO [26], while improving their runtime to 34% and 12% faster than that of ParEGO, respectively.

In addition to the aforementioned issues, systematic attempts have recently been made to deal with high-dimensional EMOPs [27], [28]. Existing insights in this regard have been mainly obtained from AOF-based SAEAs. A universal strategy is to construct surrogate models in a reduced search space which is mapped with dimension-reduction techniques, as in HeE-MOEA [29], SA-RVEA-PCA [30], and ADSAPSO [31]. Furthermore, ADSAPSO aims to accelerate the evolutionary search by executing the search in the reduced search space. Another strategy is the use of computationally efficient alternatives to Gaussian process (GP) [32], with the aim of reproducing the powerful search capacity of the GP on high-dimensional EMOPs. This strategy has been implemented in EDN-ARMOEA [27] and HeE-MOEA. These SAEAs have been tested on problems with over 70 dimensions.

However, the significant impact of using classification models on high-dimensional EMOPs has recently been revealed. Specifically, three classification-based SAEAs, namely CPS-MOEA [33], CSEA [23], and MCEA/D [28], tend to outperform MOEA/D-EGO and K-RVEA on benchmark problems with up to 150 dimensions [28]. Furthermore, they are

Y. Horaguchi is with the College of Engineering Sciences, Yokohama National University, Japan, e-mail: horaguchi-yuma-hv@ynu.jp.

K. Nishihara is with the Faculty of Engineering, Yokohama National University, Japan, e-mail: nishihara-kei-jv@ynu.jp.

M. Nakata is with the Faculty of Engineering, Yokohama National University, Japan, e-mail: nakata-masaya-tb@ynu.ac.jp.

competitive with EDN-ARMOEA, while exhibiting a reduced runtime. This trend is also confirmed by our comparison in this study; MCEA/D is competitive with dimension-reduction-based SAEAs, that is, ADSAPSO and HeE-MOEA (see Section IV).

Accordingly, our research question consists of exploring how the potential of approximation-based SAEAs can be further derived on high-dimensional EMOPs. Towards this goal, this study revisits the benefits of ASF-based SAEAs in terms of risk management for unreliable approximations and computational costs.

Specifically, for brevity, we suppose basic SAEA implementations as follows; to estimate the quality of a single solution, AOF-based SAEAs use multiple approximated objective functions, whereas ASF-based SAEAs use a single approximated scalarization function defined for a decomposed sub-problem. Supposing that a deterioration in the approximation accuracy is unavoidable under high dimensionality, we hypothesize that AOF-based SAEAs are vulnerable to unreliable approximations and ASF-based SAEAs are a reasonable option for the following reasons.

- AOF-based SAEAs use multiple approximated objective values, and thus, the reliability of the estimated solution-quality may be deteriorated if even a single unreliable model exists. ASF-based SAEAs may relax this issue because they use a single approximated value to estimate the solution-quality.
- As approximated objective functions are commonly used in the search, the negative influence of unreliable models may propagate to, and thus mislead, the entire search process [34]. ASF-based SAEAs use each approximation model for the corresponding sub-problem, thereby preventing this negative propagation.

As aforementioned, a known negative aspect of ASF-based SAEAs is the increase in their runtime as the model construction frequency increases. Although there probably exists no solid reputation in this regard, we argue that ASF-based SAEAs as well as AOF-based SAEAs exhibit the potential to be computationally efficient if computationally inexpensive surrogates are used, owing to the following reason.

- Although the calculation cost of the model construction constitutes a major part of the runtime of SAEAs, that of the model prediction is also non-negligible [35]. ASF-based SAEAs can save the number of model predictions as they can estimate the solution-quality with a single approximation value. The use of computationally inexpensive approximation models can improve the computational efficiency of ASF-based SAEAs.

Note that modern AOF-based SAEAs aim to hedge the risk of unreliable approximations, such as infill criteria and ensemble modeling [36]. Apart from these advances, there may be the aforementioned benefits of ASF-based SAEAs to handle high dimensionality. To the best of our knowledge, few examples of ASF-based SAEAs exist, as pointed out in [28], [34], [37], [38], and several comparative studies have argued the potential of ASF-based approaches for low-dimensional EMOPs [34], [39] (see Section II). However, no such approaches have been tested for high dimensionality, as the existing SAEAs adapted for high-dimensional EMOPs, that is, HeE-MOEA, SA-RVEA-PCA, ADSAPSO, and EDN-ARMOEA, employ the AOF-based approaches. Thus, the effect of the approximation of scalarization functions on high-dimensional EMOPs has remained unclear.

This study aims to demonstrate the impact of the approximation of scalarization functions under high dimensionality. We introduce a simple algorithm based on the basic principle of ASF-based SAEAs. In particular, the proposed algorithm, ASF/DE, uses the decomposition framework of MOEA/D [40], wherein a differential evolution (DE) algorithm optimizes each approximated scalarization function. In accordance with our hypothesis, we employ a radial basis function (RBF) [41] technique for the computationally efficient surrogate modelling [42]. Note that we do not intend to use any efficient extensions, such as infill criteria, to reveal the pure impact of the ASF-based approach. For this aim, we also introduce an AOF-based SAEA, namely AOF/DE, which uses approximated objective functions under the ASF/DE framework, to demonstrate the effect of ASF/DE.

The contributions of this study are summarized as follows.

- To the best of our knowledge, this study is the first to reveal the impact of the approximation of scalarization functions on high-dimensional EMOPs. Despite its methodological simplicity, ASF/DE is competitive with state-of-the-art SAEAs adapted for high-dimensional EMOPs. This finding can be viewed as a specific example of the foresight provided in [34] with respect to the potential of ASF-based approaches.
- We show that ASF/DE performs faster than AOF/DE and recent AOF-based SAEAs compared in this study. Therefore, together with the first contribution, we suggest that ASF-based SAEAs may be a promising new approach for handling high dimensionality in terms of both the performance and runtime.
- The first intensive comparison of the representative strategies for high-dimensional EMOPs, that is, dimensional reduction techniques, computationally-efficient alternatives of GPs, and classification models, is conducted on benchmark problems with up to 200 dimensions and 11 objectives. The comprehensive results reveal effective strategies for approximation-based SAEAs.

Note that this paper does not argue the methodological novelty of ASF/DE, as it follows a basic implementation of ASF-based SAEAs.

The remainder of this paper is organized as follows. Section II summarizes related works and describes the MOEA/D and RBF frameworks. The algorithms of ASF/DE and AOF/DE are explained in Section III. Section IV presents the experimental results to demonstrate the effect of ASF/DE through a comparison with state-of-the-art SAEAs, including AOF/DE. Thereafter, in Section V, additional results are provided to confirm the effectiveness of ASF/DE. Section VI presents a complexity analysis to discuss the computational efficiency of ASF/DE. Finally, conclusions are summarized in Section VII.

II. Preliminaries

A. Related work

We summarize existing insights into ASF-based SAEAs. Known improvements for high-dimensional EMOPs have been summarized in Section I, including very recent insights, that is, the superiority of classification-based SAEAs.

1) Existing ASF-based SAEAs: Although the approximation of scalarization functions is a common strategy for approximation-based SAEAs, relatively few efforts have been made in this regard [28], [34], [37], [38].

ParEGO is a representative algorithm that optimizes an approximated scalarization function using an efficient global optimization (EGO) framework. Although ParEGO was originally tested on small-scale problems, it has been compared with modern SAEAs for problems with up to 50 dimensions [43]. An alternative approach is to conduct EGO with approximated objective functions. This AOF-based approach has been employed in MOEA/D-EGO, SMS-EGO [44], and EIM-EGO [45], and their superiority to ParEGO has been confirmed. GS-MOMA [46] employs an ensemble model of approximated scalarization functions by unifying the heterogeneous models of GP, RBF, and polynomial regression. Experiments have demonstrated that GS-MOEA outperforms NSGA-II on three-objective problems with up to 50 dimensions under 8000 to 30000 FEs. SURROGATE-ASF [47] is an interactive ASF-based SAEA which adaptively selects a scalarization function to be approximated to obtain preferred solutions for a user.

These ASF-based SAEAs are based on certain motivations; for example, demonstrating an extension of EGO on EMOPs and the effect of ensemble models, rather than understanding the methodological benefits of ASF-based approaches.

2) Comparative studies: In recent years, several efforts have been made to discuss the impact of AOF/ASF-based SAEAs, as summarized below.

In [39], a comparative study of surrogate-assisted MOEA/D has been presented for a set of bi-objective problems with up to 10 dimensions. Although the effect of different definitions of training samples was mainly discussed, AOF/ASF-based MOEA/Ds (MOEA/D-filter1, 2, $\cdots$, 4) were introduced and compared in addition to other approaches such as ParEGO, M-EGO [48], and MOEA/D-RBF [15]. The observation has suggested that an AOF-based approach, namely MOEA/D-RBF, performs effectively, whereas an ASF-based MOEA/D (filter4) tends to outperform an AOF-based MOEA/D (filter1) on certain problems.

In [34], an intensive study has been conducted on small-scale unconstrained/constrained EMOPs while focusing on the methodological differences between AOF and ASF-based approaches. The experimental results have demonstrated that the approximation of an aggregated function of objective values and constraints tends to improve the performance. This superiority holds even for unconstrained problems; thus, the positive impact of ASF-based approaches has been revealed on low-dimensional problems. The authors have suggested the need for further exploration of ASF-based approaches. Note that, in [38], [49], [50], different scalarization functions to be approximated have been tested to understand their dependency on the performance of ASF-based algorithms.

Inspired by these works, this study reveals the effect of ASF-based SAEAs on high-dimensional EMOPs under the assumption that their possible benefits can be further enhanced for high dimensionality. Furthermore, a comparative discussion of the runtime is presented in this paper.

Algorithm 1 MOEA/D

Initialize $\mathcal{P}$ and EP as an empty set
Generate $\mathcal{P}$ with N initial solutions $\{\boldsymbol{x}^1, \cdots, \boldsymbol{x}^N\}$
Evaluate all initial solutions in $\mathcal{P}$
Initialize $\boldsymbol{z}$ as $z_j = \min_{\boldsymbol{x}\in\mathcal{P}} f_j(\boldsymbol{x}) \,\forall j \in \{1, \cdots, M\}$
while Termination criteria are not met **do**
    for $i = 1$ **to** N **do**
        Generate $\boldsymbol{y}^i$ from $\boldsymbol{x}^k$ and $\boldsymbol{x}^l$, where $k, l \in \mathcal{B}(i)$
        Evaluate $\boldsymbol{y}^i$
        Update $\boldsymbol{z}$ as $z_j \leftarrow \min\{z_j, f_j(\boldsymbol{y}^i)\} \;\forall j \in \{1, \cdots, M\}$
        for each $j \in \mathcal{B}(i)$ **do**
            if $g(\boldsymbol{y}^i \mid \boldsymbol{\lambda}^j, \boldsymbol{z}) \leq g(\boldsymbol{x}^j \mid \boldsymbol{\lambda}^j, \boldsymbol{z})$ **then**
                Replace $\boldsymbol{x}^j$ with $\boldsymbol{y}^i$
            end if
        end for
        Remove all solutions dominated by $\boldsymbol{y}^i$ from EP
        Add $\boldsymbol{y}^i$ to EP if $\boldsymbol{y}^i$ is the Pareto solution
    end for
end while

B. MOEA/D

This study considers a D-dimensional multi-objective optimization problem (MOP) with M objectives, formalized as

$$\begin{array}{ll} \text{minimize} & \boldsymbol{F}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})], \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}, \end{array} \tag{1}$$

where $M \geq 2$, $f : \mathcal{S} \rightarrow \mathbb{R}$, $\mathcal{S}$ is a feasible space of $\mathbb{R}^D$, and thus, $\boldsymbol{x}$ is a D-dimensional real-valued solution.

MOEA/D decomposes an MOP instance of (1) into N sub-problems, wherein a scalarization function $g : \mathbb{R}^D \rightarrow \mathbb{R}$ is assigned to each sub-problem. The Tchebycheff function is used in this study. Specifically, g for the i-th sub-problem is defined as

$$g(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z}) = \max_{1\leq j\leq M} \{\lambda^i_j |f_j(\boldsymbol{x}) - z_j|\}, \tag{2}$$

where $\boldsymbol{z} = \{z_1, z_2, \ldots, z_M\}$ is a set of reference points and $\boldsymbol{\lambda}^i$ is a weight vector such that $\sum_{j=1}^{M} \lambda^i_j = 1$ and $\lambda^i_j \in \mathbb{R}^+$. Here, $g(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ can be considered as a single-objective function, and MOEA/D aims to approximate the Pareto optimal front of (1) by independently minimizing N scalarization functions.

The detailed procedure of MOEA/D is presented in Algorithm 1, where $\mathcal{P}$ is a population and EP is an archive that contains all discovered non-dominated solutions. Initially, T neighbor sub-problems are determined for each sub-problem; an index set $\mathcal{B}(i)$ for the i-th sub-problem is defined as $\mathcal{B}(i) = \{i_1, i_2, \ldots, i_T\}$, where $\boldsymbol{\lambda}^{i_1}, \boldsymbol{\lambda}^{i_2}, \ldots, \boldsymbol{\lambda}^{i_T}$ are the T closest weight vectors to $\boldsymbol{\lambda}^i$. Subsequently, $\mathcal{P}$ is built with N initial solutions. All initial solutions are evaluated and each reference point z_j is determined as the minimum value of f_j among the objective values of the initial solutions.

The main loop is implemented as follows. For the i-th sub-problem, two neighbor solutions $\boldsymbol{x}^k$ and $\boldsymbol{x}^l$ are set as parents, where $k, l \in \mathcal{B}(i)$. A new solution $\boldsymbol{y}^i$ is produced based on the parents by applying the simulated binary crossover and the polynomial mutation. Subsequently, $\boldsymbol{y}^i$ is evaluated and $\boldsymbol{z}$ is updated. For each neighbor sub-problem that is indexed with $j \in \mathcal{B}(i)$, a neighbor solution $\boldsymbol{x}^j$ may be replaced with $\boldsymbol{y}^i$ if $g(\boldsymbol{y}^i|\boldsymbol{\lambda}^j, \boldsymbol{z}) \leq g(\boldsymbol{x}^j|\boldsymbol{\lambda}^j, \boldsymbol{z})$. Finally, $\boldsymbol{y}^i$ may be added to EP while eliminating the solutions dominated by $\boldsymbol{y}^i$ if $\boldsymbol{y}^i$ is the Pareto solution. This procedure is repeated for each sub-problem.

C. RBF

Assume n training samples $\mathcal{X} = \{\boldsymbol{x}^i\}_{i=1}^n$ and their target values $\mathcal{Y} = \{y^i\}_{i=1}^n$, where $y^i = f(\boldsymbol{x}^i)$ is mapped with a function $f : \mathbb{R}^D \to \mathbb{R}$. An RBF model expresses the approximation function of f, which is denoted by $\hat{f}$, as

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i \Phi(||\boldsymbol{x} - \boldsymbol{x}^i||), \tag{3}$$

where w_i is a scalar weight and $\Phi(||\boldsymbol{x} - \boldsymbol{x}^i||)$ is a defined kernel function. A weight vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]^T$ adapted for $\mathcal{Y}$ is then obtained as

$$\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\boldsymbol{y}, \tag{4}$$

where $\boldsymbol{y} = [y^1, y^2, \ldots, y^n]^T$ and $\boldsymbol{\Phi}$ is an $n \times n$ kernel matrix, defined as $\boldsymbol{\Phi} = [\Phi(||\boldsymbol{x}^i - \boldsymbol{x}^j||)]_{n \times n}$. This study uses a Gaussian function for $\Phi(\cdot)$, which is expressed as

$$\Phi(r) = \exp\left(-\frac{r^2}{\sigma^2}\right), \tag{5}$$

where σ^2 is set to $\sigma^2 = d_{\max}(nD)^{-1/D}$ [51]–[53] and $d_{\max}$ is the maximum distance between the samples in $\mathcal{X}$.

The time complexity for constructing an RBF model is $O(n^3)$ and that for calculating an RBF prediction is $O(n)$ [54]–[56].

III. Two basic algorithms

We introduce AOF/DE and ASF/DE, which use approximation models for the objective and scalarization functions, respectively. The only difference between these algorithms is the target functions to be approximated.

First, we define the basic framework of AOF/DE. Thereafter, ASF/DE with minimum modifications to AOF/DE is introduced. Both algorithms require the same hyper parameters: the population size N, the neighbor size T, the maximum DE iterations $\omega_{\max}$, the crossover rate CR, and the scaling factor F. Moreover, the following mathematical notations are used to introduce the AOF/DE and ASF/DE frameworks.

- FE The number of FEs consumed thus far, including those for the initialization process.
- $\mathcal{A}$ An archive set consisting of all evaluated solutions, which is identified as $\mathcal{A} = \{\boldsymbol{x}^k\}_{k=1}^{FE}$.
- $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ An approximated value for $g(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$.
- $\hat{f}_j$ An approximation model for f_j.
- $\hat{g}^i$ An approximation model for the i-th scalarization function with $\boldsymbol{\lambda}^i$.
- $\mathcal{X}$ A set of training samples.
- $\mathcal{Y}_i$ A set of target values of training samples, which is used to construct $\hat{f}_i$ or $\hat{g}^i$.
- $\mathcal{P}_{DE}$ A population of a DE algorithm.

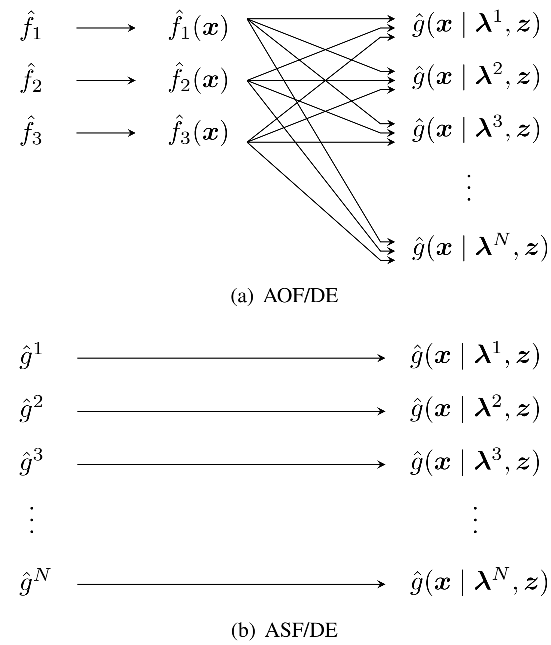


Fig. 1. Conceptual difference between AOF/DE and ASF/DE.

Hereinafter, we define that one generation of both algorithms is to complete the solution-update process for all N sub-problems, corresponding to the termination of the "for" loop in line 17 of Algorithm 1.

A. AOF/DE

1) Concept: AOF/DE follows the typical implementation of AOF-based approaches, as illustrated in Fig. 1(a). Each objective function is approximated and $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ is calculated from M approximated objective values, that is,

$$\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z}) = \max_{1 \leq j \leq M} \{\lambda_j^i |\hat{f}_j(\boldsymbol{x}) - z_j|\}. \tag{6}$$

Hence, the approximation accuracy of $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ depends on that of all M approximation models. The approximation models are constructed and fixed at the start of each generation and the same models are used for all N sub-problems during generation. Therefore, the number of approximation models required for each generation can be restricted to M.

2) Algorithm: Algorithm 2 describes the pseudo-procedure of AOF/DE, and the main modifications are highlighted with underline. Note that the termination criterion is expressly described to conform to our experimental design mentioned in Section IV; the algorithm is forcibly terminated when FE reaches the maximum FEs, $FE_{\max}$.

The initialization process is the same as that in MOEA/D, which requires N FEs. Subsequently, an approximation model

Algorithm 2 AOF/DE and ASF/DE

Initialize $\mathcal{P}, EP, \mathcal{A}$ as an empty set
Generate $\mathcal{P}$ with N initial solutions $\{\boldsymbol{x}^1, \cdots, \boldsymbol{x}^N\}$
Evaluate all initial solutions in $\mathcal{P}$, and set FE to $FE = N$
Initialize $\boldsymbol{z}$ as $z_j = \min_{\boldsymbol{x} \in \mathcal{P}} f_j(\boldsymbol{x}) \, \forall j \in \{1, \cdots, M\}$
Update $\mathcal{A}$ as $\mathcal{A} \cup \mathcal{P}$
while True **do**
  Build a set of training samples $\mathcal{X}$ from $\mathcal{A}$
  if AOF/DE **then**
    for $j = 1$ **to** M **do**
      Build the approx. model for f_j, $\hat{f}_j$, trained with $\mathcal{X}$
    end for
  else if ASF/DE **then**
    for $i = 1$ **to** N **do**
      Build the approx. model for g^i, $\hat{g}^i$, trained with $\mathcal{X}$
    end for
  end if
  for $i = 1$ **to** N **do**
    Generate $\boldsymbol{y}^i$ using DE with approximation models
    Evaluate $\boldsymbol{y}^i$ and set FE to $FE \leftarrow FE + 1$
    Update $\boldsymbol{z}$ as $z_j \leftarrow \min\{z_j, f_j(\boldsymbol{y}^i)\} \; \forall j \in \{1, \cdots, M\}$
    for each $j \in \mathcal{B}(i)$ **do**
      if $g(\boldsymbol{y}^i \mid \boldsymbol{\lambda}^j, \boldsymbol{z}) \leq g(\boldsymbol{x}^j \mid \boldsymbol{\lambda}^j, \boldsymbol{z})$ **then**
        Replace $\boldsymbol{x}^j$ with $\boldsymbol{y}^i$
      end if
    end for
    Remove all solutions dominated by $\boldsymbol{y}^i$ from EP
    Add $\boldsymbol{y}^i$ to EP if $\boldsymbol{y}^i$ is the Pareto solution
    Update $\mathcal{A}$ as $\mathcal{A} \cup \{\boldsymbol{y}^i\}$
    if $FE = FE_{\max}$ **then**
      return EP
    end if
  end for
end while

trained for $\mathcal{X}$ is constructed for each objective function at the beginning of each generation. Once the M approximation models are obtained, DE is executed to optimize $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ for each sub-problem, in which the best discovered solution is set to the offspring $\boldsymbol{y}^i$. Thereafter, the same solution-update process as in MOEA/D is performed.

The model construction and DE-based search processes are detailed in the following.

a) Model construction: We assume that well-distributed samples in the objective space tend to improve the reliability of the approximated objective functions; however, the use of many samples increases the computational cost of RBF models. To balance this tradeoff, we rank the solutions in $\mathcal{A}$ using non-dominated sorting [57] and employ the top N solutions as training samples. Although other efficient approaches may exist, the same training samples are used in AOF/DE and ASF/DE for a fair comparison.

Let $\mathcal{A}_s$ be a duplicate of $\mathcal{A}$, in which the solutions have been sorted using non-dominated sorting, and $\boldsymbol{x}^k \in \mathcal{A}_s$ is the top k-th ranked solution. The set of training samples $\mathcal{X}$ is defined as follows.

$$\mathcal{X} = \{\boldsymbol{x}^k \mid \boldsymbol{x}^k \in \mathcal{A}_s\}_{k=1}^N. \tag{7}$$

The target values are obtained as $\mathcal{Y}_j = \{f_j(\boldsymbol{x}^k) \mid \boldsymbol{x}^k \in \mathcal{X}\}_{k=1}^N$ to construct the approximation model for the j-th objective function. Subsequently, the approximation model $\hat{f}_j$ is modeled as in (3), with $\boldsymbol{w}$ adapted for $\mathcal{X}$ and $\mathcal{Y}_j$.

Algorithm 3 DE-based search

Initialize $\mathcal{P}_{DE}$ using (8)
Calculate $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ of all initial solutions in $\mathcal{P}_{DE}$
for $\omega = 1$ **to** $\omega_{\max}$ **do**
  Set $\mathcal{U}_{DE}$ to an empty set
  for $j = 1$ **to** $|\mathcal{P}_{DE}|$ **do**
    Set the best exemplar $\boldsymbol{x}^{best} \leftarrow \arg\min_{\boldsymbol{x} \in \mathcal{P}_{DE}} \hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$
    Set two random exemplars $\boldsymbol{x}^{r_1}, \boldsymbol{x}^{r_2} \in \mathcal{P}_{DE} \setminus \{\boldsymbol{x}^j\}$
    Set a random integer $k_{rand} \in \{1, \cdots, D\}$
    Initialize a new solution $\boldsymbol{u}^j = \boldsymbol{x}^j$
    for $k = 1$ **to** D **do**
      if $rand[0,1) \leq CR \; \vee \; k = k_{rand}$ **then**
        Set u_k^j as $u_k^j \leftarrow x_k^j + F(x_k^{best} - x_k^j) + F(x_k^{r_1} - x_k^{r_2})$
      end if
    end for
    Calculate $\hat{g}(\boldsymbol{u}^j|\boldsymbol{\lambda}^i, \boldsymbol{z})$ of $\boldsymbol{u}^j$
    if $\hat{g}(\boldsymbol{u}^j|\boldsymbol{\lambda}^i, \boldsymbol{z}) \leq \hat{g}(\boldsymbol{x}^j|\boldsymbol{\lambda}^i, \boldsymbol{z})$ **then**
      Add $\boldsymbol{u}^j$ to $\mathcal{U}_{DE}$
    else
      Add $\boldsymbol{x}^j$ to $\mathcal{U}_{DE}$
    end if
  end for
  Update $\mathcal{P}_{DE}$ as $\mathcal{P}_{DE} \leftarrow \mathcal{U}_{DE}$
end for
Determine $\boldsymbol{y}^i$ as $\boldsymbol{y}^i = \arg\min_{\boldsymbol{x} \in \mathcal{P}_{DE}} \hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$
return $\boldsymbol{y}^i$

b) DE-based search: Algorithm 3 describes the DE-based search process in detail. We employ DE with the current-to-best/1 mutation and the binomial crossover to accelerate the search process. Moreover, we aim to inherit the current search progress under the assumption of MOEA/D, that is, the optimum solutions of neighbor sub-problems exist in a similar region. Specifically, for the i-th sub-problem, its T neighbor solutions indexed with $\mathcal{B}(i)$ are used as the initial solutions. That is, $\mathcal{P}_{DE}$ is initialized as

$$\mathcal{P}_{DE} := \{\boldsymbol{x}^k \mid \boldsymbol{x}^k \in \mathcal{P}, \forall k \in \mathcal{B}(i)\}. \tag{8}$$

Thus, the population size of $\mathcal{P}_{DE}$ is set to T. Subsequently, the DE search process is repeated for $\omega_{\max}$ iterations to minimize $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ given by (6). Finally, the offspring solution for the i-th sub-problem is determined as the best discovered solution;

$$\boldsymbol{y}^i = \arg\min_{\boldsymbol{x} \in \mathcal{P}_{DE}} \hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z}). \tag{9}$$

B. ASF/DE

1) Concept: As illustrated in Fig. 1(b), ASF/DE directly approximates each scalarization function. Given an approximated scalarization function that is modeled by (3), $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ is formalized as

$$\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z}) = \hat{g}^i(\boldsymbol{x}) = \sum_{k=1}^{N} w_k^i \Phi(||\boldsymbol{x} - \boldsymbol{x}^k||), \tag{10}$$

where $\boldsymbol{x}^k \in \mathcal{X}$ and $\boldsymbol{w}^i = [w_1^i, \cdots, w_n^i]^T$ is a weight vector adapted for $\mathcal{Y}_i$. The approximation accuracy of $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ is dependent only on a single model $\hat{g}^i$ and N approximation models must be built during one generation.

TABLE I
PARAMETER SETTINGS AND CATEGORICAL FEATURES OF COMPARED ALGORITHMS.

Algorithm	Category (ML)	Parameter setting
K-RVEA (2016)	AOF (GP)	$N = \{91, 91, 77\}$ for $M = \{3, 7, 11\}$, $\alpha = 2$, $f_r = 0.1$, $\lvert u \rvert = 5$, $\delta = 0.05N$, $w_{\max} = 20$, $d_c = 20$, and $p_c = 1.0$ [25]
HeE-MOEA (2019)	AOF (SVM, RBF, PCA)	$N = 100$, $K_m = 5$, $l = 11D - 1 + 25$, the population size and the generation number of NSGA-II are 50 [29]
EDN-ARMOEA (2022)	AOF (ANN)	$N = 100$, $\delta = 0.08$, $iter = 20$, $k = 5$, $J = K = 40$, $wd = 10^{-5}$, $lr = 0.01$, $iter_{\mathrm{train}} = 80000$, $iter_{\mathrm{test}} = 100$, $iter_r = 8000$, $p_I = 0.8$, $p_R = 0.5$, $d_c = 20$, and $p_c = 1.0$ [27]
ADSAPSO (2022)	AOF (RBF)	$N = 100$, $K = 5$, $\beta = 0.5$, $N_\alpha = 200$, $N_s = 50$, the population size and the generation number of PSO are 100 [31]
CPS-MOEA (2015)	Classification (KNN)	$N = 100$, $F = 0.5$, $CR = 1.0$, and the number of offspring is 5 [33]
CSEA (2019)	Classification (ANN)	$N = 100$, $H = 10$, $K = 6$, and the number of iterations for each model construction is 800, $d_c = 20$, and $p_c = 1.0$ [23], [28]
MCEA/D (2022)	Classification (SVM)	$N = \{91, 91, 77\}$ for $M = \{3, 7, 11\}$, $R_{\max} = 10$, $\gamma = 1.0$, $C = 1.0$, $T = \lceil 0.1N \rceil$, $\delta = 0.9$, $n_r = 2$, $F = 0.5$, and $CR = 1.0$ [28]
AOF/DE	AOF (RBF)	$N = \{91, 91, 77\}$ for $M = \{3, 7, 11\}$, $T = \lceil 0.1N \rceil$, $F = 0.5$, $CR = 0.9$, and $\omega_{\max} = 20$
ASF/DE	ASF (RBF)	The same setting as in AOF/DE

2) Algorithm: As shown in Algorithm 2, the only difference from AOF/DE is to construct an approximation model of each scalarization function, as indicated in lines 13 and 14. This requires a modification of $\mathcal{Y}_i$, that is,

$$\mathcal{Y}_i = \{g(\boldsymbol{x}^k|\boldsymbol{\lambda}^i, \boldsymbol{z}) \mid \boldsymbol{x}^k \in \mathcal{X}\}_{k=1}^{N}. \tag{11}$$

Note that $\mathcal{X}$ is determined using the same procedure as that in AOF/DE.

Once $\hat{g}^i$ has been obtained by solving (4) with $\mathcal{X}$ and $\mathcal{Y}_i$, the DE-based search is executed, as in Algorithm 3, where $\hat{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ is calculated directly from (10) with $\hat{g}^i$.

IV. EXPERIMENTS

This section validates the effect of approximating the scalarization functions in terms of the performance and runtime. We compare ASF/DE with AOF/DE and state-of-the-art SAEAs adapted for high-dimensional EMOPs. All experiments reported in this paper are conducted on the evolutionary multi-objective optimization platform (PlatEMO) [58] with Intel Core i7-9700 (3.0 GHz) CPU and 16 GB RAM.

A. Experimental design

1) Test problems: Eleven MaF problems (MaF1 to 7, 10 to 13) [59] are used, wherein the number of objectives and problem dimension are set to $M = \{3, 7, 11\}$ and $D = \{50, 100, 150, 200\}$, respectively. These problem scales are based on existing studies on high-dimensional EMOPs [27], [28], [31]. Note that MaF8, 9, 14, and 15 have been eliminated from our experiments owing to the defined settings of the problem dimension; D must be set to 2 and $20 \times M$ for {MaF8, 9} and {MaF14, 15}, respectively.

2) Comparison: ASF/DE is compared with AOF/DE and the following seven SAEAs: K-RVEA, HeE-MOEA, EDN-ARMOEA, ADSAPSO, CPS-MOEA, CSEA, and MCEA/D. The parameter settings and categorical features of the algorithms are summarized in Table I. Note that $N = 100$ is used as a common setting in this study with reference to [22], [28]. However, for the decomposition-based SAEAs, $N = \{91, 91, 77\}$ is used for $M = \{3, 7, 11\}$ through the two-layered approach [60] with $(H_1, H_2) = \{(12, 0), (3, 1), (2, 1)\}$ to obtain uniformly distributed weight vectors of the scalarization functions. We use the Latin hypercube sampling method [61] to obtain the initial solutions. The maximum FEs, $FE_{\max}$ is set to 500, including those for the initialization process.

3) Evaluation criteria: The performance of the algorithms is evaluated using the Inverted Generational Distance (IGD) [62]. As shown in Algorithm 2, the algorithms are forcibly terminated when $FE_{\max}$ FEs have been consumed. Subsequently, the IGD values are calculated from the Pareto solutions that are identified from all evaluated solutions. The average IGD values of 21 trials with different random seeds are reported. Furthermore, the Wilcoxon rank-sum test is applied to find the significant difference with a significance level of $p < 0.05$.

B. Results

Tables II and III report the average IGD values for $D = \{50, 150\}$. The IGD values for $D = \{100, 200\}$ are reported in Table S-I in Section A of the Supplementary Material. The statistical results of the Wilcoxon rank-sum test are summarized at the bottom of each table; "+," "−," and "≈" indicate that the IGD value of the compared algorithm is significantly better than, worse than, and competitive with that of ASF/DE, respectively. The statistical results for all problem dimensions are summarized in Table IV. The mean rank that is calculated from the average IGD values is shown in Fig. 2, where the left and right figures show the mean ranks for the problem dimension and number of objectives, respectively. The runtime (wall-clock time) is shown in Fig. 3.

1) Impact of approximating scalarization functions: Table IV confirms the positive impact of ASF/DE compared to AOF/DE. The IGD values of ASF/DE are significantly better than those of AOF/DE on at least 24 experimental cases for all problem dimensions. Moreover, the number of "−" gradually increases with an increase in the problem dimension; for $D = 200$, ASF/DE does not significantly underperform AOF/DE on any experimental cases. This superiority of ASF/DE can also be observed in the mean rank, as illustrated in Fig. 2; the mean rank of ASF/DE is better than that of AOF/DE.

TABLE II

AVERAGE IGD VALUES FOR $D = 50$ (21 TRIALS).

	M	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO	CPS-MOEA	CSEA	MCEA/D	AOF/DE	ASF/DE
MaF1	3	1.935e+00 −	4.114e-01 −	2.867e+00 −	1.062e+00 −	2.077e+00 −	1.269e+00 −	4.235e-01 −	3.450e-01 −	2.154e-01
	7	4.084e+00 −	7.402e-01 −	4.413e+00 −	2.106e+00 −	3.587e+00 −	2.011e+00 −	6.593e-01 −	6.609e-01 −	4.268e-01
	11	3.909e+00 −	8.378e-01 −	4.878e+00 −	1.909e+00 −	4.061e+00 −	1.950e+00 −	6.852e-01 −	1.042e+00 −	5.018e-01
MaF2	3	2.033e-01 −	1.101e-01 −	2.517e-01 −	1.577e-01 −	2.215e-01 −	1.695e-01 −	9.842e-02 −	8.093e-02 ≈	7.551e-02
	7	2.477e-01 +	2.610e-01 +	2.647e-01 +	2.274e-01 +	2.364e-01 +	3.003e-01 ≈	2.810e-01 ≈	2.879e-01 ≈	2.870e-01
	11	3.311e-01 +	3.014e-01 +	3.455e-01 +	2.948e-01 +	3.182e-01 +	3.998e-01 ≈	3.882e-01 ≈	3.982e-01 ≈	4.016e-01
MaF3	3	3.337e+07 −	1.641e+07 −	1.864e+07 −	4.009e+06 −	9.647e+06 −	2.307e+07 −	1.591e+06 ≈	3.321e+07 −	2.362e+06
	7	2.510e+07 −	1.791e+07 −	1.518e+07 −	1.733e+07 −	9.019e+06 −	2.295e+07 −	1.429e+06 −	2.478e+07 −	1.162e+06
	11	2.275e+07 −	1.810e+07 −	1.178e+07 −	1.545e+07 −	7.862e+06 −	1.900e+07 −	1.211e+06 −	2.311e+07 −	9.726e+05
MaF4	3	1.047e+04 −	1.221e+04 −	1.139e+04 −	6.234e+03 ≈	9.544e+03 −	6.419e+03 −	4.703e+03 ≈	8.622e+03 −	5.815e+03
	7	1.418e+05 −	1.909e+05 −	1.789e+05 −	1.392e+05 −	1.569e+05 −	1.038e+05 −	7.864e+04 ≈	1.342e+05 −	8.574e+04
	11	2.295e+06 −	2.744e+06 −	2.579e+06 −	2.072e+06 −	2.174e+06 −	1.567e+06 −	9.952e+05 ≈	1.916e+06 −	1.253e+06
MaF5	3	1.785e+01 −	6.382e+00 −	8.736e+00 −	4.191e+00 +	8.640e+00 −	5.116e+00 ≈	5.243e+00 ≈	1.572e+01 −	4.814e+00
	7	9.219e+01 −	7.866e+01 −	3.047e+01 +	4.022e+01 ≈	4.033e+01 ≈	2.666e+01 +	3.765e+01 ≈	7.020e+01 −	4.013e+01
	11	1.423e+03 −	8.767e+02 −	2.499e+02 +	4.019e+02 −	3.984e+02 −	2.749e+02 ≈	3.989e+02 −	6.575e+02 −	2.917e+02
MaF6	3	1.992e+02 −	2.448e+01 −	2.532e+02 −	7.844e+01 −	1.556e+02 −	1.018e+02 −	1.813e+01 −	2.923e+01 −	1.027e+01
	7	2.262e+02 −	3.830e+01 −	2.255e+02 −	1.477e+02 −	1.591e+02 −	1.069e+02 −	1.863e+01 −	3.705e+01 −	1.069e+01
	11	1.777e+02 −	3.523e+01 −	2.004e+02 −	1.679e+02 −	1.451e+02 −	8.613e+01 −	2.563e+01 −	5.233e+01 −	1.918e+01
MaF7	3	5.527e+00 ≈	3.891e+00 ≈	4.111e+00 ≈	5.559e+00 −	8.863e+00 −	4.929e+00 −	8.330e+00 −	7.198e+00 −	3.626e+00
	7	1.196e+01 −	1.562e+01 −	8.878e+00 −	1.877e+01 −	2.064e+01 −	2.350e+01 −	1.885e+01 −	1.264e+01 −	3.244e+00
	11	1.675e+01 −	2.885e+01 −	1.690e+01 −	3.002e+01 −	3.303e+01 −	3.815e+01 −	2.806e+01 −	2.224e+01 −	5.434e+00
MaF10	3	2.257e+00 −	1.989e+00 −	2.063e+00 −	2.081e+00 −	2.270e+00 −	1.586e+00 +	2.168e+00 −	1.943e+00 −	1.893e+00
	7	2.945e+00 −	2.866e+00 ≈	2.782e+00 +	2.738e+00 +	2.948e+00 −	2.635e+00 +	2.982e+00 −	2.826e+00 +	2.904e+00
	11	3.581e+00 ≈	3.576e+00 +	3.471e+00 +	3.449e+00 +	3.624e+00 ≈	3.507e+00 +	3.647e+00 ≈	3.614e+00 ≈	3.620e+00
MaF11	3	8.817e-01 ≈	6.918e-01 +	7.836e-01 ≈	7.016e-01 +	8.246e-01 ≈	5.879e-01 +	7.354e-01 +	8.007e-01 ≈	8.077e-01
	7	2.805e+00 −	2.221e+00 ≈	1.616e+00 +	1.629e+00 +	1.796e+00 +	1.907e+00 +	2.207e+00 ≈	2.367e+00 ≈	2.191e+00
	11	4.396e+00 ≈	3.911e+00 ≈	2.825e+00 +	3.004e+00 +	3.026e+00 +	4.461e+00 ≈	3.387e+00 +	4.410e+00 ≈	4.384e+00
MaF12	3	9.223e-01 −	8.484e-01 −	9.375e-01 −	7.960e-01 −	9.067e-01 −	8.511e-01 −	7.692e-01 −	8.427e-01 −	6.716e-01
	7	3.088e+00 +	3.590e+00 ≈	3.493e+00 +	3.710e+00 ≈	3.775e+00 ≈	4.067e+00 −	3.749e+00 ≈	3.793e+00 ≈	3.739e+00
	11	7.448e+00 +	8.395e+00 +	8.615e+00 +	8.844e+00 ≈	8.931e+00 ≈	1.004e+01 −	8.660e+00 +	9.694e+00 −	9.162e+00
MaF13	3	2.224e+00 −	1.170e+00 −	2.607e+00 −	9.819e-01 ≈	2.393e+00 −	1.182e+00 −	1.117e+00 −	1.782e+00 −	8.058e-01
	7	2.062e+04 −	3.972e+01 ≈	1.259e+03 −	4.090e+00 +	2.706e+03 −	8.091e+00 +	2.852e+00 +	1.270e+04 −	2.202e+02
	11	4.303e+04 −	8.286e+01 ≈	1.429e+03 −	1.135e+01 ≈	1.985e+03 −	1.736e+01 ≈	3.002e+00 +	3.982e+04 −	9.295e+01
$+/-/\approx$		4/25/4	5/21/7	10/21/2	9/18/6	4/24/5	7/20/6	5/17/11	1/24/8	-

TABLE III

AVERAGE IGD VALUES FOR $D = 150$ (21 TRIALS).

	M	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO	CPS-MOEA	CSEA	MCEA/D	AOF/DE	ASF/DE
MaF1	3	1.074e+01 −	5.031e+00 −	1.152e+01 −	4.379e+00 −	6.581e+00 −	7.545e+00 −	9.427e-01 −	2.258e+00 −	6.491e-01
	7	2.192e+01 −	9.567e+00 −	2.207e+01 −	8.131e+00 −	1.223e+01 −	1.289e+01 −	1.187e+00 −	3.722e+00 −	9.926e-01
	11	2.665e+01 −	1.271e+01 −	2.833e+01 −	1.062e+01 −	1.510e+01 −	1.617e+01 −	1.194e+00 −	5.516e+00 −	9.261e-01
MaF2	3	8.425e-01 −	4.459e-01 −	8.455e-01 −	4.260e-01 −	5.533e-01 −	5.987e-01 −	2.001e-01 −	2.630e-01 −	1.169e-01
	7	5.219e-01 −	3.390e-01 +	4.848e-01 −	4.557e-01 −	4.696e-01 −	5.061e-01 −	3.967e-01 −	4.092e-01 −	3.714e-01
	11	5.070e-01 −	3.471e-01 +	4.439e-01 +	4.309e-01 +	4.547e-01 ≈	5.052e-01 −	4.655e-01 ≈	4.948e-01 ≈	4.749e-01
MaF3	3	1.198e+09 −	2.098e+08 −	2.109e+08 −	7.193e+07 −	1.157e+08 −	5.409e+08 −	1.756e+07 +	1.199e+09 −	2.176e+07
	7	7.702e+08 −	3.270e+08 −	2.019e+08 −	5.048e+08 −	1.298e+08 −	4.615e+08 −	1.856e+07 −	7.767e+08 −	1.341e+07
	11	1.407e+09 −	3.099e+08 −	1.885e+08 −	5.434e+08 −	1.213e+08 −	4.355e+08 −	1.768e+07 −	1.407e+09 −	1.222e+07
MaF4	3	4.207e+04 −	4.215e+04 −	3.990e+04 −	2.455e+04 −	3.262e+04 −	3.031e+04 −	1.879e+04 −	2.544e+04 −	1.571e+04
	7	7.044e+05 −	6.898e+05 −	6.742e+05 −	5.194e+05 −	5.848e+05 −	6.345e+05 −	3.200e+05 −	3.585e+05 −	2.576e+05
	11	1.079e+07 −	1.074e+07 −	1.053e+07 −	8.437e+06 −	9.091e+06 −	1.032e+07 −	4.827e+06 −	5.576e+06 −	4.087e+06
MaF5	3	6.372e+01 −	2.811e+01 −	2.676e+01 −	1.146e+01 −	2.577e+01 −	1.742e+01 −	9.771e+00 −	4.959e+01 −	6.485e+00
	7	4.289e+02 −	1.694e+02 −	4.999e+01 ≈	5.456e+01 ≈	5.187e+01 ≈	4.878e+01 ≈	4.923e+01 ≈	1.803e+02 −	6.074e+01
	11	5.926e+03 −	1.435e+03 −	3.444e+02 +	4.942e+02 −	5.032e+02 −	3.390e+02 +	4.910e+02 −	1.290e+03 −	4.152e+02
MaF6	3	1.001e+03 −	4.661e+02 −	9.847e+02 −	4.304e+02 −	5.360e+02 −	6.669e+02 −	6.535e+01 −	2.661e+02 −	5.038e+01
	7	9.743e+02 −	3.818e+02 −	9.544e+02 −	7.605e+02 −	5.859e+02 −	7.202e+02 −	1.091e+02 −	2.954e+02 −	7.337e+01
	11	9.138e+02 −	3.255e+02 −	9.180e+02 −	7.460e+02 −	5.658e+02 −	7.129e+02 −	1.452e+02 −	3.378e+02 −	9.482e+01
MaF7	3	1.069e+01 −	8.338e+00 +	6.965e+00 +	7.289e+00 +	1.023e+01 −	7.016e+00 +	1.021e+01 −	1.014e+01 −	8.952e+00
	7	2.648e+01 −	2.247e+01 −	1.959e+01 ≈	2.357e+01 −	2.543e+01 −	2.670e+01 −	2.387e+01 −	2.263e+01 −	1.841e+01
	11	4.252e+01 −	3.907e+01 −	3.255e+01 ≈	3.700e+01 −	4.049e+01 −	4.312e+01 −	3.820e+01 −	3.788e+01 −	3.164e+01
MaF10	3	2.397e+00 −	2.098e+00 −	2.087e+00 −	2.116e+00 −	2.261e+00 −	1.628e+00 +	2.221e+00 −	1.893e+00 +	1.963e+00
	7	3.029e+00 −	2.913e+00 ≈	2.832e+00 +	2.807e+00 +	2.957e+00 −	2.627e+00 +	2.983e+00 −	2.849e+00 +	2.915e+00
	11	3.693e+00 −	3.625e+00 ≈	3.525e+00 +	3.486e+00 +	3.632e+00 ≈	3.523e+00 +	3.643e+00 ≈	3.613e+00 ≈	3.627e+00
MaF11	3	1.088e+00 −	7.689e-01 ≈	8.308e-01 −	7.118e-01 +	8.493e-01 −	7.147e-01 +	6.946e-01 +	8.070e-01 ≈	7.818e-01
	7	3.074e+00 −	2.294e+00 ≈	1.693e+00 +	1.845e+00 +	1.799e+00 +	1.965e+00 +	1.768e+00 +	2.083e+00 ≈	2.321e+00
	11	5.556e+00 −	3.951e+00 ≈	3.022e+00 +	3.242e+00 +	3.076e+00 +	4.447e+00 ≈	3.393e+00 +	4.106e+00 ≈	4.541e+00
MaF12	3	1.151e+00 −	9.095e-01 −	1.030e+00 −	9.180e-01 −	9.970e-01 −	9.427e-01 −	7.916e-01 −	8.692e-01 −	6.634e-01
	7	4.606e+00 −	3.323e+00 ≈	3.752e+00 −	3.835e+00 −	3.708e+00 −	4.057e+00 −	3.559e+00 −	3.753e+00 −	3.301e+00
	11	1.013e+01 −	7.622e+00 +	9.151e+00 −	9.382e+00 −	8.870e+00 −	9.283e+00 −	8.904e+00 −	9.261e+00 −	8.479e+00
MaF13	3	4.040e+00 −	2.588e+00 −	3.769e+00 −	1.039e+00 ≈	2.386e+00 −	2.254e+00 −	1.121e+00 −	2.780e+00 −	9.885e-01
	7	3.910e+04 −	6.258e+02 −	1.417e+04 −	6.064e+00 ≈	2.760e+03 −	1.841e+02 −	3.497e+00 +	3.964e+04 −	1.768e+01
	11	9.607e+04 −	1.444e+03 −	2.321e+04 −	8.473e+00 +	4.868e+03 −	3.709e+02 −	4.270e+00 +	7.718e+04 −	5.238e+01
$+/-/\approx$		0/33/0	4/23/6	7/23/3	8/22/3	2/28/3	7/24/2	6/24/3	2/26/5	-

TABLE IV
STATISTICAL RESULTS OF COMPARISON WITH ASF/DE ($+/-/\approx$). THE SYMBOLS "$+$," "$-$," AND "$\approx$" INDICATE THAT THE IGD VALUE OF THE COMPARED ALGORITHM IS SIGNIFICANTLY BETTER THAN, WORSE THAN, AND COMPETITIVE WITH THAT OF ASF/DE, RESPECTIVELY.

D	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO
50	4/25/4	5/21/7	10/21/2	9/18/6
100	1/32/0	5/24/4	6/24/3	9/20/4
150	0/33/0	4/23/6	7/23/3	8/22/3
200	0/33/0	4/24/5	5/23/5	5/21/7
D	CPS-MOEA	CSEA	MCEA/D	AOF/DE
50	4/24/5	7/20/6	5/17/11	1/24/8
100	3/26/4	7/22/4	7/19/7	1/25/7
150	2/28/3	7/24/2	6/24/3	2/26/5
200	2/29/2	4/24/5	5/23/5	0/28/5

TABLE V
STATISTICAL RESULTS OF COMPARISON WITH AOF/DE ($+/-/\approx$). THE SYMBOLS "$+$," "$-$," AND "$\approx$" INDICATE THAT THE IGD VALUE OF THE COMPARED ALGORITHM IS SIGNIFICANTLY BETTER THAN, WORSE THAN, AND COMPETITIVE WITH THAT OF AOF/DE, RESPECTIVELY.

D	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO
50	5/13/15	14/8/11	19/13/1	18/10/5
100	1/28/4	11/13/9	14/14/5	12/10/11
150	0/27/6	12/13/8	15/14/4	12/12/9
200	0/28/5	10/12/11	16/14/3	16/12/5
D	CPS-MOEA	CSEA	MCEA/D	ASF/DE
50	12/14/7	16/10/7	19/7/7	24/1/8
100	10/14/9	13/13/7	21/5/7	25/1/7
150	12/16/5	12/15/6	21/4/8	26/2/5
200	11/16/6	11/15/7	22/4/7	28/0/5

As reported in Fig. 3, ASF/DE performs sufficiently faster than AOF/DE. The runtime of AOF/DE is clearly dependent on the number of objectives; however, ASF/DE relaxes this dependency. This is because that the number of required approximation models depends only on N in ASF/DE. This observation will be theoretically confirmed in Section VI.

2) Comparison with modern SAEAs: The IGD values of ASF/DE are sufficiently competitive with those of the compared algorithms while exhibiting a reduced runtime, as shown in Table IV and Fig. 3. Specifically, the number of "$-$" is more than 16 cases, but that of "$+$" is curbed to up to 10 for all problem dimensions. Furthermore, the number of "$-$" gradually increases with an increase in the problem dimension. Consequently, ASF/DE has been assigned to the first rank even when the problem dimension and the number of objectives increase (see Fig. 2). As illustrated in this figure, ASF/DE is scalable, especially for the problem dimension.

We further validate that the above impact of ASF/DE as achieved by approximating scalarization functions. Table V summarizes the statistical results ($+/-/\approx$) when the baseline algorithm is set to AOF/DE, where "$+$," "$-$," and "$\approx$" indicate that the IGD value of the compared algorithm is significantly better than, worse than, and competitive with that of AOF/DE, respectively. The superiority of AOF/DE cannot be observed except for K-RVEA, and MCEA/D sufficiently outperforms AOF/DE.

This observation confirms that approximating scalarization functions contributes to the superiority of ASF/DE, rather than its basic algorithm, that is, the use of RBF models with our defined training samples and the DE-based solution search.

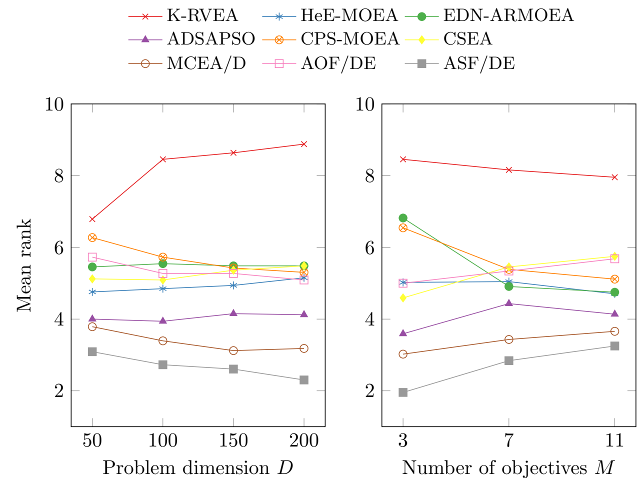


Fig. 2. Mean rank of compared algorithms. The left and right figures present the average values for the D-dimensional and M-objective experimental cases, respectively.

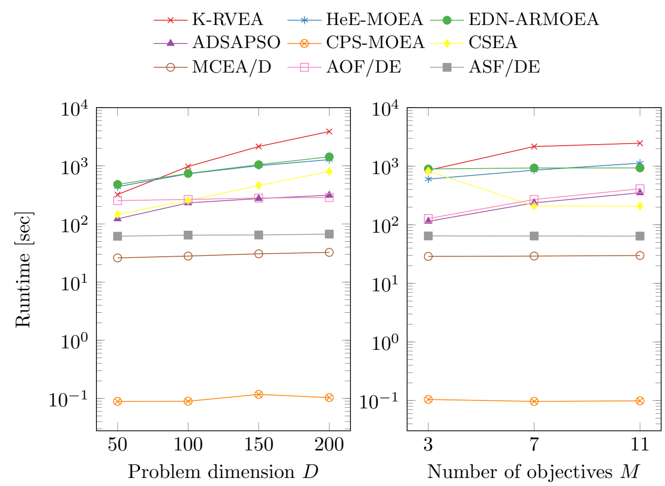


Fig. 3. Average runtime for computing one trial. The left and right figures further show the average runtime for the D-dimensional and M-objective experimental cases, respectively.

3) Other insights: Based on our comprehensive comparison, the following insights are presented, particularly for approximation-based SAEAs.

- The effect of the dimension-reduction technique may be enhanced further by conducting the evolutionary search in a reduced search space, as ADSAPSO performs effectively. This tendency is consistent with recent observations on autoencoder-embedded approaches for single-objective problems [63], [64].
- Among the approximation-based SAEAs, K-RVEA, HeE-MOEA, and EDN-ARMOEA stably improve their mean rank with the increase in the objectives. Although we have not identified a specific reason for this, the use of GP models and their alternatives tends to improve the scalability of SAEAs for the number of objectives.
- In addition to the above insight, the mean rank of EDN-ARMOEA clearly improves when the number of

TABLE VI
SUMMARY OF STATISTICAL RESULTS ON $D = \{10, 20, 30\}$ ($+/-/\approx$).

D	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO
10	16/10/3	7/19/3	12/14/3	16/9/4
20	18/12/3	9/21/3	11/19/3	13/13/7
30	13/16/4	5/21/7	11/21/1	11/16/6
D	CPS-MOEA	CSEA	MCEA/D	AOF/DE
10	7/19/3	14/8/7	9/9/11	5/15/9
20	6/24/3	11/13/9	6/17/10	3/19/11
30	5/24/4	10/16/7	8/16/9	2/21/10

TABLE VII
STATISTICAL RESULTS OF ASF/DE WITH RAND/1 MUTATION AND RANDOM INITIAL SOLUTIONS ($+/-/\approx$).

D	ASF/DE/rand/1	ASF/DE/LHS
10	8/9/12	8/19/2
20	11/5/17	7/24/2
30	14/2/17	7/26/0
50	12/3/18	7/25/1
100	8/11/14	8/25/0
150	8/13/12	6/25/2
200	9/14/10	6/26/1

objectives increases. EDN-ARMOEA constructs a single ANN model that approximates M objective values in a lump, and this approach is intuitively antithetical to that of ASF/DE. Thus, the potential of approximation-based SAEAs may be further expanded with this direction.

V. ADDITIONAL INSIGHTS

We provide additional insights into the ASF/DE framework. Specifically, this section is mainly composed of two parts: the impact on low-/middle-dimensional EMOPs and a demonstration of the affinity of ASF/DE with the DE settings. All experiments employ the same experimental settings with 21 trials as in the previous section, unless stated otherwise.

A. Results on low- and middle-dimensional EMOPs

Supposing that a deterioration in the approximation accuracy may occur on any problem dimension, we can expect that the effectiveness of ASF/DE is observed even in low- and middle-dimensional problems. We provide additional results in this regard.

The problem dimension is set to $D = \{10, 20, 30\}$ for the low-/middle-dimensional problem scales [65]. Note that MaF7, 10, 11, and 12 for $\{D = 10, M = 11\}$ have been eliminated because these problems define D as greater than M [59]. The statistical results are summarized in Table VI, and the specific IGD values are reported in Table S-II in Section A of the Supplementary Material.

As shown in the table, ASF/DE derives competitive performance with the compared algorithms except for K-RVEA. The effectiveness of ASF/DE gradually increases with the increase in the problem dimension, as the number of "$-$" increases. Furthermore, ASF/DE sufficiently outperforms AOF/DE for all problem dimensions.

Thus, the positive impact of ASF/DE can be observed even for $20, 30$-dimensional problems. This suggests that the approximation of scalarization functions can be a "fast-acting" strategy to handle an increase in the problem dimension.

B. Impact of DE settings

Existing single-objective SAEAs are often designed to search local regions around superior solutions with surrogate models specialized to those regions [65]–[67]. Inspired by this insight, AOF/DE and ASF/DE have employed the current-to-best/1 mutation and neighbor solutions as the initial solutions. We validate this effect on ASF/DE.

Two ASF/DE variants are compared: ASF/DE/rand/1, which uses the rand/1 mutation, and ASF/DE/LHS, which uses random initial solutions for DE, generated using the Latin hypercube sampling method. We set $D = \{10, 20, 30, 50, 100, 150, 200\}$ for a comprehensive comparison. The statistical results are summarized in Table VII and the IGD values are reported in Table S-III in Section A of the Supplementary Material. Note that "$+$," "$-$," and "$\approx$" indicate that the IGD value of the ASF/DE variant is significantly better than, worse than, and competitive with that of the default version of ASF/DE, respectively.

We can firstly confirm the effect of using neighbor solutions. The obtained IGD values of ASF/DE are significantly better than those of ASF/DE/LHS on at least 19 experimental cases for all problem dimensions. However, a relatively low effect is observed for the current-to-best/1 mutation. For $20 \leq D \leq 50$, ASF/DE/rand/1 improves the IGD values of ASF/DE on at least 11 experimental cases. When $D \geq 100$, the number of "$-$" gradually increases greater than that of "$+$".

The results demonstrate the possibility of further improving ASF/DE by tuning the DE settings. More generally, a possible advantage of ASF-based SAEAs is that we can use various insights in the field of single-objective optimization. Although ASF/DE has been designed in a straightforward manner, it is expected that ASF-based SAEAs may be advanced further in this regard.

C. Impact of training dataset

The training samples in AOF/DE and ASF/DE are defined as the top N solutions, as ranked by the non-dominated sorting. This definition is better suited to AOF/DE because superior solutions that are well distributed in the objective space can be obtained. Accordingly, we discuss other possible definitions of the training samples suitable for ASF/DE.

A typical strategy is the use of all evaluated solutions. However, as mentioned in Section V-B, it may be more effective to specialize an approximated scalarization function to a good region for each corresponding sub-problem. Accordingly, we test the following two definitions for the training samples:

- All evaluated solutions in $\mathcal{A}$ are used as the training samples. ASF/DE with this definition is denoted as ASF/DE/$\mathcal{A}$.
- The top N solutions ranked with the values of each scalarization function are used as the training samples, that is,

$$\mathcal{X}_i = \{\boldsymbol{x}^k \mid \boldsymbol{x}^k \in \mathcal{A}_i^g\}_{k=1}^{N}, \tag{12}$$

where $\mathcal{A}_i^g$ is a duplicate of $\mathcal{A}$ in which the solutions have been sorted by $g(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ in the ascending order. Thus,

TABLE VIII
STATISTICAL RESULTS WITH DIFFERENT TRAINING SAMPLES ($+/-/\approx$).

D	ASF/DE/$\mathcal{A}$	ASF/DE/$\mathcal{A}_i^g$
10	4/0/25	8/0/21
20	2/1/30	6/0/27
30	1/0/32	8/0/25
50	2/3/28	5/3/25
100	3/2/28	8/2/23
150	3/1/29	6/2/25
200	2/1/30	3/1/29

the set of training samples $\mathcal{X}_i$ is constructed for each sub-problem. ASF/DE with this definition is denoted as ASF/DE/$\mathcal{A}_i^g$.

We set $D = \{10, 20, 30, 50, 100, 150, 200\}$. The statistical results are summarized in Table VIII, in which "$+$," "$-$," and "$\approx$" indicate that the IGD value of the ASF/DE variant is significantly better than, worse than, and competitive with that of the default version of ASF/DE, respectively. The IGD values are reported in Table S-IV in Section A of the Supplementary Material.

From the table, there is almost no sufficient difference between ASF/DE/$\mathcal{A}$ and ASF/DE. Thus, ASF/DE can achieve similar performance to ASF/DE/$\mathcal{A}$ while reducing the number of training samples. However, ASF/DE/$\mathcal{A}_i^g$ improves the IGD value of ASF/DE on certain experimental cases. This tendency is consistent with known insights, as discussed in Section V-B.

VI. COMPLEXITY ANALYSIS

As revealed in Section IV, ASF/DE performs faster than AOF/DE. Although AOF/DE and ASF/DE use almost the same algorithmic procedures, their model construction and prediction frequencies differ. Therefore, we expect this difference to be the main factor in determining the computational efficiency. The prime goal here is to theoretically confirm this insight by conducting a complexity analysis.

Apart from the algorithmic consistency between AOF/DE and ASF/DE, the design of existing SAEAs has been varied in terms of evolutionary algorithms, machine learning, and model management strategies. For example, different from AOF/DE, K-RVEA runs RVEA [68] with M GP models, and $|u|$ new solutions per model construction are evaluated with the original objective functions. Thus, it is worth to understand the theoretical boundary that ASF-based SAEAs perform faster or slower than AOF-based SAEAs, under the freedom of such methodological varieties. A complete comparison is non-trivial, but the computational efficiencies of AOF/ASF-based SAEAs can be roughly compared, as demonstrated in this section. Thus, the secondly goal of this section is to provide general insights in this regard.

Our main assumption is to focus only on the computational cost consumed for the model construction and prediction. The big-O notation is generally used to represent the time complexity of algorithms, which is suitable for asymptotic analysis, such as $O(n^2)$ for $n^2 + 1000n$ as $n \to \infty$. However, this asymptotic representation may hinder a comparative discussion of the computational efficiency of SAEAs because possible factors of the computational order are typically restricted to small-scale values, and the impact of the coefficient parameters of these factors may be unignorable. Accordingly, we compare the number of computations without asymptotic analysis.

Algorithm 4 Generalized AOF/ASF-based SAEA

1: Generate $\mathcal{P}$ with N initial evaluated solutions
2: Set FE to $FE = N$
3: Set η to 0
4: **while** True **do**
5: Update η as $\eta \leftarrow \eta + 1$
6: **if** AOF-based SAEA **then**
7: Construct M approximation models of objective functions trained with n samples
8: **else if** ASF-based SAEA **then**
9: Construct N approximation models of scalarization functions trained with n samples
10: **end if**
11: Generate n_u solutions $\{\boldsymbol{u}^i\}_{i=1}^{n_u}$ by optimizing the approximation models through an optimizer; the optimizer generates n_c candidate solutions during the search, and their quality is estimated with the model predictions
12: **for** $i = 1$ **to** n_u **do**
13: Evaluate $\boldsymbol{u}^i$ with $FE \leftarrow FE + 1$
14: Update $\mathcal{P}$ with $\boldsymbol{u}^i$
15: **if** $FE = FE_{\max}$ **then**
16: **return** $\mathcal{P}$
17: **end if**
18: **end for**
19: **end while**

A. Preparation

Here, we suppose generalized descriptions of AOF/ASF-based SAEAs, as described in Algorithm 4, in which a solution-generation process with an optimizer remains a black-box procedure in line 11, and η indicates the number of generations. We use the following four assumptions. First, the optimizer generates n_c candidate solutions during the search. The quality of all n_c candidate solutions must be estimated with the model predictions, and no additional prediction is performed. Furthermore, n_u solutions to be evaluated with the original objective functions are selected from the n_c candidate solutions for each execution of the optimizer, thereby consuming n_u FEs. Finally, AOF/ASF-based SAEAs use common values of n_c and n_u as well as the same optimizer and the same machine learning algorithm, for a fair comparison.

Algorithm 4 has the freedom in selecting the optimizer and machine learning as well as n_c and n_u, which increases the generality of our analytical conclusions. Moreover, the actual frameworks of AOF/DE and ASF/DE can be precisely converted into Algorithm 4 (see Section VI-C).

We use the following notations:

$T_{const}^{model}, T_{pred}^{model}$ The number of computations consumed when constructing a single approximation model and calculating its single prediction, respectively.

$n_{const}^{*}, n_{pred}^{*}$ The number of model constructions and predictions consumed until the termination of the algorithm, respectively.

T^*_{const}, T^*_{pred} The number of computations consumed in the model construction and prediction until the termination of the algorithm, respectively.

T^*_{total} The total number of computations consumed in the model construction and prediction in the algorithm.

In the above, $*$ is either *AOF* or *ASF*, and T^*_{total} is obtained as

$$\begin{aligned} T^*_{total} &= T^*_{const} + T^*_{pred} \\ &= n^*_{const} \times T^{model}_{const} + n^*_{pred} \times T^{model}_{pred}, \end{aligned} \tag{13}$$

where the time complexity of the algorithms is obtained as $O(T^*_{total})$ if the big-O notation is used.

Given $FE_{\max}$, the maximum number of generations $\eta_{\max}$ is determined as $\eta_{\max} = \lceil (FE_{\max} - N)/n_u \rceil$. The number of model constructions n^*_{const} is determined as

$$n^{AOF}_{const} = \left\lceil \frac{FE_{\max} - N}{n_u} \right\rceil \times M, \tag{14}$$

$$n^{ASF}_{const} = \left\lceil \frac{FE_{\max} - N}{n_u} \right\rceil \times N, \tag{15}$$

because AOF/ASF-based SAEAs build M and N approximation models, respectively, for each generation.

Here, $\eta_{\max} \times n_c$ candidate solutions are eventually generated until the termination of the algorithm. Thus, the number of model predictions n^*_{pred} is determined as

$$n^{AOF}_{pred} = \left\lceil \frac{FE_{\max} - N}{n_u} \right\rceil \times n_c \times M, \tag{16}$$

$$n^{ASF}_{pred} = \left\lceil \frac{FE_{\max} - N}{n_u} \right\rceil \times n_c, \tag{17}$$

because AOF-based SAEAs require M predictions for each candidate solution. Thus, according to (13),

$$T^{AOF}_{total} = \left\lceil \frac{FE_{\max} - N}{n_u} \right\rceil (T^{model}_{const} \times M + T^{model}_{pred} \times n_c \times M), \tag{18}$$

$$T^{ASF}_{total} = \left\lceil \frac{FE_{\max} - N}{n_u} \right\rceil (T^{model}_{const} \times N + T^{model}_{pred} \times n_c), \tag{19}$$

can be obtained.

B. Theoretical boundary

Based on (14) to (17), the following equalities hold:

$$n^{ASF}_{const} = \frac{N}{M} n^{AOF}_{const}, \quad n^{ASF}_{pred} = \frac{1}{M} n^{AOF}_{pred}. \tag{20}$$

These equalities reveal the insights mentioned in Section I; ASF-based SAEAs require a higher computational cost for the model construction than AOF-based SAEAs because $N > M$ typically holds, and it can save the model prediction while eliminating the dependence on the number of objectives.

Recalling our assumption that the computational efficiency of SAEAs is discussed only for the model construction and prediction, ASF-based SAEAs perform faster than AOF-based SAEAs if the following condition is satisfied:

$$T^{ASF}_{total} < T^{AOF}_{total}, \tag{21}$$

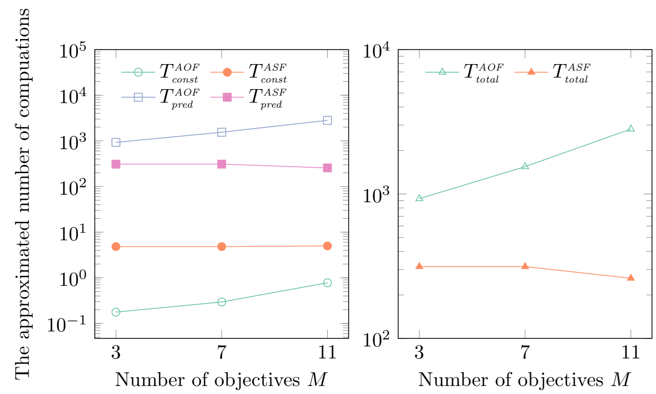


Fig. 4. Approximated number of computations of AOF/DE and ASF/DE with parameter settings employed in Section IV.

which can now be specified as

$$n_c > \frac{N - M}{M - 1} \frac{T^{model}_{const}}{T^{model}_{pred}}, \tag{22}$$

according to (18) and (19). Note that T^{model}_{const} and T^{model}_{pred} must be greater than zero. A solution of n_c that satisfies the above inequality always exists, as its right term is always smaller than ∞ because $M > 1$ holds on MOPs. The difference in the computational efficiency of AOF/ASF-based SAEAs does not depend on n_u but on N, M, or the employed approximation model. Importantly, (22) can be an "easy-to-use" criterion for determining whether ASF-based SAEAs perform faster than AOF-based SAEAs.

Although it is difficult to formalize T^{model}_{const} and T^{model}_{pred} strictly, they can be substituted with the runtime consumed in processing the T^{model}_{const} and T^{model}_{pred} computations, respectively. For example, according to our experiment, the employed RBF models consumed 0.0118 and 0.0036 seconds for single model construction/prediction with 100 training samples, respectively; and thus, $T^{model}_{const}/T^{model}_{pred} \simeq 3.278$.

C. AOF/DE vs. ASF/DE

As a case study, we validate our theoretical results in Section VI-B using the actual frameworks of AOF/DE and ASF/DE.

To begin with, we identify the specific values of n_u and n_c for both algorithms. These execute DEs N times (that is, for N sub-problems) for each model construction, wherein each DE produces $T(1 + \omega_{\max})$ solutions including the T initial ones. Note that T is the population size of DE. Therefore, we can consider that both algorithms execute an aggregated optimizer of N DEs per model construction, and that these evaluate N offspring solutions selected from $TN(1 + \omega_{\max})$ candidate solutions. Thus, $n_u = N$ and $n_c = TN(1 + \omega_{\max})$ are obtained. Moreover, we use $T^{model}_{const}/T^{model}_{pred} \simeq 3.278$.

Fig. 4 shows the theoretical curves of T^*_{const}, T^*_{pred}, and T^*_{total} calculated from (13) with the parameter settings employed in Section IV. As noted in (20), T^{ASF}_{const} is greater than T^{AOF}_{const}, and T^{ASF}_{pred} is smaller than T^{AOF}_{pred}. Importantly, T^*_{const} is sufficiently smaller than T^*_{pred}, indicating that the computational cost of the model construction is unignorable in our case. Consequently,

T_{total}^{ASF} is sufficiently smaller than T_{toal}^{AOF}; therefore, it is expected that ASF/DE performs faster than AOF/DE under our experimental settings. Furthermore, the landscapes of T_{toal}^{AOF} and T_{toal}^{ASF} are similar to those of the actual runtime of both algorithms (see right figure in Fig. 3).

Finally, we theoretically demonstrate that the condition of (22) has been satisfied with our parameter settings. Recalling that $n_c = TN(1+\omega_{\max})$, (22) can be rewritten as

$$\omega_{\max} > \frac{N-M}{TN(M-1)} \frac{T_{const}^{model}}{T_{pred}^{model}} - 1. \tag{23}$$

The above inequality holds if the following is satisfied:

$$\omega_{\max} > \frac{1}{T(M-1)} \frac{T_{const}^{model}}{T_{pred}^{model}}. \tag{24}$$

Accordingly, by substituting our parameter settings into (24); that is, $T = \lceil 0.1N \rceil$ and $T_{const}^{model}/T_{pred}^{model} \simeq 3.278$, we can identify that $\omega_{\max} \geq 1$ for $M = \{3, 7, 11\}$. Therefore, we conclude that ASF/DE has performed faster than AOF/DE because we set $\omega_{\max} = 20$.

More generally, (24) indicates that ASF/DE always performs faster than AOF/DE if

$$T(M-1) > \frac{T_{const}^{model}}{T_{pred}^{model}} \tag{25}$$

holds, because this inequality always satisfies $\omega_{\max} \geq 1$. For example, because we set $T = 8$ for $\{N, M\} = \{77, 11\}$, ASF/DE should perform faster than AOF/DE unless T_{const}^{model} is more than 80 times greater than T_{pred}^{model}.

VII. Conclusion

This study first hypothesized that the approximation of scalarization functions is reasonable to hedge the risk of unreliable approximations, and that this benefit should emerge in high-dimensional EMOPs. We introduced and compared two surrogate-assisted extensions of MOEA/D, namely AOF/DE and ASF/DE, in which the objective and scalarization functions are approximated, respectively. The experimental results demonstrated that ASF/DE significantly outperforms AOF/DE on problems with up to 200 dimensions, thereby supporting the hypothesis of this study.

In addition to the above fundamental insights, the results revealed that ASF/DE is sufficiently competitive with state-of-the-art SAEAs adapted for high-dimensional EMOPs, while reducing its runtime. Therefore, we suggest that the approximation of scalarization functions may be a promising new strategy for handling the high dimensionality of problems.

There probably exists no consensus for the computational efficiency of ASF-based approaches. Therefore, we attempted mathematical organization in this regard. Our theoretical results suggest that ASF-based SAEA can be computationally efficient unless the computational time for the model construction is significantly, say 10 times, greater than that for the model prediction. The results also indicated an "easy-to-use" criterion for the approximate comparison of the computational efficiency of AOF/ASF-based SAEAs.

Because ASF/DE has been designed in a straightforward manner, we expect that the potential of ASF-based SAEAs can be further derived using modern insights and extensions that have been accumulated in single-/multi-objective SAEAs. We will explore this issue in future work.

References

[1] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[2] L. M. Pavelski, M. R. Delgado, C. P. Almeida, R. A. Gonçalves, and S. M. Venske, "Extreme Learning Surrogate Models in Multi-objective Optimization based on Decomposition," *Neurocomputing*, vol. 180, pp. 55–67, 2016.

[3] L. V. Santana-Quintero, A. A. Montano, and C. A. Coello Coello, "A Review of Techniques for Handling Expensive Functions in Evolutionary Multi-Objective Optimization," *Computational Intelligence in Expensive Optimization Problems*, pp. 29–59, 2010.

[4] M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya, "A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods," *Structural and Multidisciplinary Optimization*, vol. 52, pp. 1–25, 2015.

[5] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, vol. 41, pp. 219–241, 2010.

[6] J.-Y. Li, Z.-H. Zhan, and J. Zhang, "Evolutionary Computation for Expensive Optimization: A Survey," *Machine Intelligence Research*, vol. 19, pp. 3–23, 2022.

[7] A. Oyama, T. Kohira, H. Kemmotsu, T. Tatsukawa, and T. Watanabe, "Simultaneous structure design optimization of multiple car models using the K computer," in *IEEE Symposium Series on Computational Intelligence*, 2017, pp. 1–4.

[8] T. Kohira, H. Kemmotsu, O. Akira, and T. Tatsukawa, "Proposal of benchmark problem based on real-world car structure design optimization," in *ACM Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 183–184.

[9] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[10] A. Díaz-Manríquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal, "A review of surrogate assisted multiobjective evolutionary algorithms," *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–14, 2016.

[11] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, vol. 23, pp. 3137–3166, 2019.

[12] L. Shi and K. Rasheed, "A survey of Fitness Approximation Methods Applied in Evolutionary Algorithms," *Computational Intelligence in Expensive Optimization Problems*, vol. 2, pp. 3–28, 2010.

[13] X.-F. Lu and K. Tang, "Classification-and Regression-Assisted Differential Evolution for Computationally Expensive Problems," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 1024–1034, 2012.

[14] R. Cheng, Y. Jin, M. Olhofer, and s. Bernhard, "Test Problems for Large-Scale Multiobjective and Many-Objective Optimization," *IEEE Trans. on Cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2016.

[15] S. Zapotecas Martínez and C. A. Coello Coello, "MOEA/D assisted by RBF networks for expensive multi-objective optimization problems," in *ACM Genetic and Evolutionary Computation Conference*, 2013, pp. 1405–1412.

[16] F. Liu, Q. Zhang, and Z. Han, "MOEA/D with gradient-enhanced kriging for expensive multiobjective optimization," *Natural Computing*, pp. 1–11, 2022.

[17] M. Wu, L. Wang, J. Xu, P. Hu, and P. Xu, "Adaptive surrogate-assisted multi-objective evolutionary algorithm using an efficient infill technique," *Swarm and Evolutionary Computation*, vol. 75, p. 101170, 2022.

[18] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization," *Information Sciences*, vol. 519, pp. 317–331, 2020.

[19] F. Li, L. Gao, A. Garg, W. Shen, and S. Huang, "Two infill criteria driven surrogate-assisted multi-objective evolutionary algorithms for computationally expensive problems with medium dimensions," *Swarm and Evolutionary Computation*, vol. 60, p. 100774, 2021.

[20] N. Azzouz, S. Bechikh, and L. Ben Said, "Steady state IBEA assisted by MLP neural networks for expensive multi-objective optimization problems," in *ACM Genetic and Evolutionary Computation Conference*, 2014, pp. 581–588.

[21] N. Zheng, H. Wang, and B. Yuan, "An adaptive model switch-based surrogate-assisted evolutionary algorithm for noisy expensive multi-objective optimization," *Complex & Intelligent Systems*, vol. 8, pp. 1–18, 2022.

[22] Z. Song, H. Wang, C. He, and Y. Jin, "A Kriging-Assisted Two-Archive Evolutionary Algorithm for Expensive Many-Objective Optimization," *IEEE Trans. on Evolutionary Computation*, vol. 25, no. 6, pp. 1013–1027, 2021.

[23] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A Classification-Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization," *IEEE Trans. on Evolutionary Computation*, vol. 23, no. 1, pp. 74–88, 2019.

[24] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model," *IEEE Trans. on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.

[25] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization," *IEEE Trans. on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2018.

[26] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[27] D. Guo, X. Wang, K. Gao, Y. Jin, J. Ding, and T. Chai, "Evolutionary Optimization of High-Dimensional Multiobjective and Many-Objective Expensive Problems Assisted by a Dropout Neural Network," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2084–2097, 2022.

[28] T. Sonoda and M. Nakata, "Multiple Classifiers-Assisted Evolutionary Algorithm Based on Decomposition for High-Dimensional Multi-Objective Problems," *IEEE Trans. on Evolutionary Computation (Early Access)*, pp. 1–1, 2022.

[29] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous Ensemble-Based Infill Criterion for Evolutionary Multiobjective Optimization of Expensive Problems," *IEEE Trans. on Cybernetics*, vol. 49, no. 3, pp. 1012–1025, 2019.

[30] M. Zhao, K. Zhang, G. Chen, X. Zhao, C. Yao, H. Sun, Z. Huang, and J. Yao, "A surrogate-assisted multi-objective evolutionary algorithm with dimension-reduction for production optimization," *Journal of Petroleum Science and Engineering*, vol. 192, p. 107192, 2020.

[31] J. Lin, C. He, and R. Cheng, "Adaptive dropout for high-dimensional expensive multiobjective optimization," *Complex & Intelligent Systems*, vol. 8, pp. 271–285, 2022.

[32] C. E. Rasmussen, *Gaussian Processes in Machine Learning*. Springer Berlin Heidelberg, 2004, pp. 63–71.

[33] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *IEEE Congress on Evolutionary Computation*, 2015, pp. 2883–2890.

[34] K. Deb, R. Hussein, P. C. Roy, and G. Toscano-Pulido, "A Taxonomy for Metamodeling Frameworks for Evolutionary Multiobjective Optimization," *IEEE Trans. on Evolutionary Computation*, vol. 23, no. 1, pp. 104–116, 2019.

[35] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning forecasting methods: Concerns and ways forward," *PloS ONE*, vol. 13, no. 3, p. e0194889, 2018.

[36] Q. Lin, X. Wu, L. Ma, J. Li, M. Gong, and C. A. Coello Coello, "An Ensemble Surrogate-Based Framework for Expensive Multiobjective Evolutionary Optimization," *IEEE Trans. on Evolutionary Computation*, vol. 26, no. 4, pp. 631–645, 2022.

[37] D. Horn, T. Wagner, D. Biermann, C. Weihs, and B. Bischl, "Model-Based Multi-objective Optimization: Taxonomy, Multi-Point Proposal, Toolbox and Benchmark," in *Springer Evolutionary Multi-Criterion Optimization*, 2015, pp. 64–78.

[38] A. A. Rahat, R. M. Everson, and J. E. Fieldsend, "Alternative infill strategies for expensive multi-objective optimisation," in *ACM Genetic and Evolutionary Computation Conference*, 2017, pp. 873–880.

[39] N. Berveglieri, B. Derbel, A. Liefooghe, H. Aguirre, and K. Tanaka, "Surrogate-assisted multiobjective optimization based on decomposition: a comprehensive comparative analysis," in *ACM Genetic and Evolutionary Computation Conference*, 2019, pp. 507–515.

[40] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[41] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Royal Signals and Radar Establishment Malvern (United Kingdom), Tech. Rep., 1988.

[42] Y. Liu, J. Liu, S. Tan, Y. Yang, and F. Li, "A bagging-based surrogate-assisted evolutionary algorithm for expensive multi-objective optimization," *Neural Computing and Applications*, pp. 1–22, 2022.

[43] H. Wang, H. Xu, and Y. Yuan, "High-dimensional expensive multi-objective optimization via additive structure," *Intelligent Systems with Applications*, vol. 14, p. 200062, 2022.

[44] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted $\mathcal{S}$-Metric Selection," in *Springer Parallel Problem Solving from Nature*, 2008, pp. 784–794.

[45] D. Zhan, Y. Cheng, and J. Liu, "Expected Improvement Matrix-Based Infill Criteria for Expensive Multiobjective Optimization," *IEEE Trans. on Evolutionary Computation*, vol. 21, no. 6, pp. 956–975, 2017.

[46] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing Surrogate-Assisted Evolutionary Computation," *IEEE Trans. on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

[47] M. Tabatabaei, M. Hartikainen, K. Sindhya, J. Hakanen, and K. Miettinen, "An interactive surrogate-based method for computationally expensive multiobjective optimisation," *Journal of the Operational Research Society*, vol. 70, no. 6, pp. 898–914, 2019.

[48] R. Hussein and K. Deb, "A Generative Kriging Surrogate Model for Constrained and Unconstrained Multi-objective Optimization," in *ACM Genetic and Evolutionary Computation Conference*, 2016, pp. 573–580.

[49] P. S. Palar, T. Tsuchiya, and G. T. Parks, "A comparative study of local search within a surrogate-assisted multi-objective memetic algorithm framework for expensive problems," *Applied Soft Computing*, vol. 43, pp. 1–19, 2016.

[50] T. Chugh, "Scalarizing Functions in Bayesian Multiobjective Optimization," in *IEEE Congress on Evolutionary Computation*, 2020, pp. 1–8.

[51] G. Chen, K. Zhang, X. Xue, L. Zhang, C. Yao, J. Wang, and J. Yao, "A radial basis function surrogate model assisted evolutionary algorithm for high-dimensional expensive optimization problems," *Applied Soft Computing*, vol. 116, p. 108353, 2022.

[52] C. Praveen and D. Regis, "Radial Basis Functions and Kriging Metamodels for Aerodynamic Optimization," *INRIA*, 2007.

[53] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Information Sciences*, vol. 454–455, pp. 59–72, 2018.

[54] V. Skala, "RBF Interpolation with CSRBF of Large Data Sets," *Procedia Computer Science*, vol. 108, pp. 2433–2437, 2017.

[55] N. Flyer, E. Lehto, S. Blaise, G. B. Wright, and A. St-Cyr, "A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere," *Journal of Computational Physics*, vol. 231, no. 11, pp. 4078–4095, 2012.

[56] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3D objects with radial basis functions," in *ACM Computer Graphics and Interactive Techniques*, 2001, pp. 67–76.

[57] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II," in *Springer Parallel Problem Solving from Nature*. Springer, 2000, pp. 849–858.

[58] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[59] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, pp. 67–81, 2017.

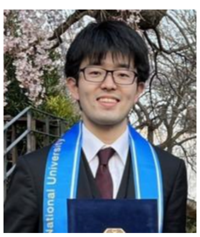
[60] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[61] M. D. McKay, R. J. Beckman, and W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[62] C. A. Coello Coello and N. C. Cortés, "Solving Multiobjective Optimization Problems Using an Artificial Immune System," *Genetic Programming and Evolvable Machines*, vol. 6, pp. 163–190, 2005.

[63] M. Cui, L. Li, and M. Zhou, "An Autoencoder-embedded Evolutionary Optimization Framework for High-dimensional Problems," in *IEEE Systems, Man, and Cybernetics*, 2020, pp. 1046–1051.
[64] M. Cui, L. Li, M. Zhou, and A. Abusorrah, "Surrogate-Assisted Autoencoder-Embedded Evolutionary Optimization Algorithm to Solve High-Dimensional Expensive Problems," *IEEE Trans. on Evolutionary Computation*, vol. 26, no. 4, pp. 676–689, 2022.
[65] H. Hao, A. Zhou, H. Qian, and H. Zhang, "Expensive Multiobjective Optimization by Relation Learning and Prediction," *IEEE Trans. on Evolutionary Computation*, vol. 26, no. 5, pp. 1157–1170, 2022.
[66] G. Chen, Y. Li, K. Zhang, X. Xue, J. Wang, Q. Luo, C. Yao, and J. Yao, "Efficient hierarchical surrogate-assisted differential evolution for high-dimensional expensive optimization," *Information Sciences*, vol. 542, pp. 228–246, 2021.
[67] F. Yu, W. Gong, and H. Zhen, "A data-driven evolutionary algorithm with multi-evolutionary sampling strategy for expensive optimization," *Knowledge-Based Systems*, vol. 242, p. 108436, 2022.
[68] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization," *IEEE Trans. on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

Yuma Horaguchi is an undergraduate student at the College of Engineering Science, Yokohama National University, Japan. His research interests include multi-objective optimization and surrogate-assisted evolutionary optimization.



Kei Nishihara is a PhD candidate at the Graduate School of Engineering Science, Yokohama National University, Japan. He received his M. Eng. from Yokohama National University in 2022. His research interests include evolutionary algorithms and surrogate-assisted evolutionary optimization.

Masaya Nakata is an associate professor at the Faculty of Engineering, Yokohama National University, Japan. He received his Ph.D. in informatics from the University of Electro-Communications, Japan, in 2016. He has mainly been working on evolutionary machine learning and its theory. Since 2019, he has focused his research on surrogate-assisted evolutionary algorithms. His contributions have been published in more than 15 journal papers and 35 conference papers, including IEEE T-EVC, IEEE CEC, and GECCO.

Supplementary Material for "Impact of Approximating Scalarization Functions on High-dimensional Multiobjective Optimization: A Fast and Scalable Approach"

Yuma Horaguchi, Kei Nishihara, and Masaya Nakata, *Member, IEEE,*

This supplementary material provides additional results for the main manuscript entitled "Impact of Approximating Scalarization Functions on High-dimensional Multiobjective Optimization: A Fast and Scalable Approach".

A. IGD VALUES

This section reports the obtained IGD values on experiments conducted on the main manuscript.

Tables S-I and S-II report the IGD values for $D = \{100, 200\}$ and $\{10, 20, 30\}$ on the MaF problems (21 trials), respectively. The symbols, "$+$," "$-$," and "$\approx$" indicate that the IGD value of the compared algorithm is significantly better than, worse than, and competitive with that of ASF/DE, respectively. Note that MaF7, 10, 11, and 12 for $\{D = 10, M = 11\}$ have been eliminated because these problems define D as greater than M.

Table S-III reports the IGD values of two alternatives, ASF/DE/rand/1 and ASF/DE/LHS. Note that "$+$," "$-$," and "$\approx$" indicate that the IGD value of the alternative is significantly better than, worse than, and competitive with that of the default version of ASF/DE, respectively.

Tables S-IV reports the IGD values of two alternatives, ASF/DE/$\mathcal{A}$ and ASF/DE/$\mathcal{A}_i^g$. The symbols, "$+$," "$-$," and "$\approx$" indicate that the IGD value of the alternative is significantly better than, worse than, and competitive with that of the default version of ASF/DE.

TABLE S-I
AVERAGE IGD VALUES FOR $D = \{100, 200\}$ (21 TRIALS).

(a) $D = 100$

	M	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO	CPS-MOEA	CSEA	MCEA/D	AOF/DE	ASF/DE
MaF1	3	6.464e+00 −	1.986e+00 −	7.074e+00 −	2.580e+00 −	4.191e+00 −	4.083e+00 −	6.902e-01 −	1.164e+00 −	4.670e-01
	7	1.254e+01 −	3.487e+00 −	1.329e+01 −	4.731e+00 −	7.704e+00 −	6.615e+00 −	8.953e-01 −	1.982e+00 −	7.166e-01
	11	1.593e+01 −	3.765e+00 −	1.652e+01 −	5.967e+00 −	9.935e+00 −	8.369e+00 −	8.901e-01 −	3.008e+00 −	6.728e-01
MaF2	3	5.305e-01 −	2.476e-01 −	5.411e-01 −	2.655e-01 −	4.115e-01 −	3.443e-01 −	1.581e-01 −	1.733e-01 −	1.001e-01
	7	3.890e-01 −	2.863e-01 +	3.668e-01 −	3.350e-01 ≈	3.627e-01 −	3.973e-01 −	3.553e-01 ≈	3.417e-01 ≈	3.445e-01
	11	4.022e-01 +	2.982e-01 +	3.789e-01 +	3.491e-01 +	3.668e-01 +	4.304e-01 ≈	4.158e-01 +	4.320e-01 ≈	4.334e-01
MaF3	3	2.658e+08 −	7.817e+07 −	8.612e+07 −	1.974e+07 −	4.743e+07 −	1.206e+08 −	7.150e+06 +	2.626e+08 −	1.266e+07
	7	1.755e+08 −	9.493e+07 −	8.067e+07 −	7.953e+07 −	4.826e+07 −	1.327e+08 −	8.157e+06 −	1.713e+08 −	6.405e+06
	11	2.707e+08 −	9.548e+07 −	7.419e+07 −	1.125e+08 −	4.623e+07 −	1.088e+08 −	7.944e+06 −	2.707e+08 −	5.065e+06
MaF4	3	2.582e+04 −	2.701e+04 −	2.520e+04 −	1.712e+04 −	2.074e+04 −	1.783e+04 −	1.001e+04 ≈	1.782e+04 −	9.778e+03
	7	4.259e+05 −	4.449e+05 −	4.251e+05 −	3.250e+05 −	3.579e+05 −	3.451e+05 −	1.940e+05 −	2.748e+05 −	1.682e+05
	11	6.209e+06 −	6.746e+06 −	6.390e+06 −	5.142e+06 −	5.588e+06 −	5.388e+06 −	2.882e+06 ≈	4.389e+06 −	3.125e+06
MaF5	3	4.150e+01 −	1.681e+01 −	1.734e+01 −	7.546e+00 −	1.662e+01 −	1.039e+01 −	7.786e+00 −	3.261e+01 −	5.694e+00
	7	2.999e+02 −	1.312e+02 −	4.104e+01 ≈	4.444e+01 ≈	4.662e+01 +	3.783e+01 ≈	4.551e+01 +	1.247e+02 −	4.666e+01
	11	3.821e+03 −	1.085e+03 −	3.417e+02 ≈	4.325e+02 ≈	4.601e+02 −	3.075e+02 +	4.409e+02 −	9.069e+02 −	3.762e+02
MaF6	3	5.885e+02 −	1.980e+02 −	6.154e+02 −	2.382e+02 −	3.664e+02 −	3.639e+02 −	4.567e+01 −	1.287e+02 −	3.336e+01
	7	5.934e+02 −	2.055e+02 −	5.808e+02 −	4.399e+02 −	3.810e+02 −	3.963e+02 −	4.983e+01 ≈	1.540e+02 −	4.541e+01
	11	5.541e+02 −	1.814e+02 −	5.522e+02 −	3.933e+02 −	3.767e+02 −	3.660e+02 −	8.968e+01 −	1.958e+02 −	6.464e+01
MaF7	3	1.065e+01 −	6.645e+00 +	5.655e+00 +	6.762e+00 +	9.676e+00 −	6.214e+00 +	9.678e+00 −	9.832e+00 −	7.743e+00
	7	2.511e+01 −	2.037e+01 −	1.561e+01 −	2.251e+01 −	2.438e+01 −	2.593e+01 −	2.209e+01 −	1.990e+01 −	1.269e+01
	11	3.786e+01 −	3.552e+01 −	2.756e+01 −	3.467e+01 −	3.843e+01 −	4.186e+01 −	3.548e+01 −	3.304e+01 −	2.373e+01
MaF10	3	2.373e+00 −	2.079e+00 −	2.095e+00 −	2.057e+00 −	2.260e+00 −	1.588e+00 +	2.229e+00 −	1.924e+00 ≈	1.923e+00
	7	3.031e+00 −	2.918e+00 ≈	2.811e+00 +	2.834e+00 +	2.964e+00 −	2.618e+00 +	2.983e+00 −	2.842e+00 +	2.913e+00
	11	3.691e+00 −	3.611e+00 ≈	3.502e+00 +	3.475e+00 +	3.627e+00 ≈	3.487e+00 +	3.641e+00 ≈	3.624e+00 ≈	3.637e+00
MaF11	3	1.075e+00 −	7.462e-01 +	8.298e-01 ≈	7.235e-01 +	8.402e-01 ≈	6.629e-01 +	7.038e-01 +	7.937e-01 ≈	8.086e-01
	7	3.120e+00 −	2.154e+00 ≈	1.677e+00 +	1.732e+00 +	1.812e+00 ≈	1.988e+00 ≈	1.870e+00 ≈	2.356e+00 ≈	2.171e+00
	11	5.439e+00 −	3.902e+00 ≈	2.921e+00 +	3.034e+00 +	3.138e+00 +	4.396e+00 ≈	3.142e+00 +	4.600e+00 ≈	4.003e+00
MaF12	3	1.101e+00 −	9.218e-01 −	1.005e+00 −	9.107e-01 −	9.627e-01 −	9.349e-01 −	7.846e-01 −	8.708e-01 −	6.828e-01
	7	4.294e+00 −	3.534e+00 −	3.683e+00 −	3.971e+00 −	3.728e+00 −	4.040e+00 −	3.694e+00 −	3.703e+00 −	3.326e+00
	11	9.752e+00 −	8.012e+00 +	9.026e+00 −	9.287e+00 −	8.892e+00 ≈	9.580e+00 −	8.978e+00 ≈	9.198e+00 −	8.665e+00
MaF13	3	3.814e+00 −	2.059e+00 −	3.469e+00 −	1.037e+00 ≈	2.426e+00 −	1.800e+00 −	1.133e+00 −	2.480e+00 −	9.922e-01
	7	3.547e+04 −	1.843e+02 −	7.811e+03 −	6.544e+00 +	4.073e+03 −	7.000e+01 −	3.918e+00 +	3.570e+04 −	2.987e+01
	11	6.204e+04 −	2.928e+02 −	1.473e+04 −	7.867e+00 +	3.566e+03 −	1.327e+02 +	3.412e+00 +	3.517e+04 −	1.331e+02
$+/-/\approx$		1/32/0	5/24/4	6/24/3	9/20/4	3/26/4	7/22/4	7/19/7	1/25/7	-

(b) $D = 200$

	M	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO	CPS-MOEA	CSEA	MCEA/D	AOF/DE	ASF/DE
MaF1	3	1.594e+01 −	9.208e+00 −	1.592e+01 −	6.776e+00 −	8.752e+00 −	1.090e+01 −	1.252e+00 −	4.128e+00 −	8.529e-01
	7	3.156e+01 −	1.713e+01 −	3.067e+01 −	1.324e+01 −	1.652e+01 −	1.938e+01 −	1.556e+00 −	5.816e+00 −	1.242e+00
	11	4.022e+01 −	2.102e+01 −	3.967e+01 −	1.548e+01 −	2.163e+01 −	2.504e+01 −	1.581e+00 −	6.865e+00 −	1.198e+00
MaF2	3	1.170e+00 −	6.627e-01 −	1.145e+00 −	5.968e-01 −	6.906e-01 −	8.165e-01 −	2.260e-01 −	3.621e-01 −	1.209e-01
	7	6.481e-01 −	4.006e-01 −	6.143e-01 −	5.783e-01 −	5.404e-01 −	6.107e-01 −	4.303e-01 −	4.120e-01 −	3.824e-01
	11	5.517e-01 −	3.734e-01 +	5.036e-01 ≈	4.874e-01 ≈	5.049e-01 ≈	5.469e-01 −	5.007e-01 ≈	5.272e-01 −	4.890e-01
MaF3	3	3.512e+09 −	4.121e+08 −	3.927e+08 −	9.263e+07 −	2.085e+08 −	1.516e+09 −	3.263e+07 +	3.513e+09 −	4.731e+07
	7	2.317e+09 −	6.871e+08 −	3.757e+08 −	1.001e+09 −	2.376e+08 −	1.320e+09 −	3.959e+07 −	2.327e+09 −	2.538e+07
	11	4.738e+09 −	6.913e+08 −	3.513e+08 −	1.199e+09 −	2.192e+08 −	1.219e+09 −	3.696e+07 −	4.737e+09 −	2.267e+07
MaF4	3	5.736e+04 −	5.593e+04 −	5.475e+04 −	3.693e+04 −	4.491e+04 −	4.471e+04 −	2.371e+04 −	3.754e+04 −	2.137e+04
	7	9.755e+05 −	9.539e+05 −	9.275e+05 −	8.931e+05 −	8.011e+05 −	9.308e+05 −	4.503e+05 −	5.262e+05 −	3.640e+05
	11	1.562e+07 −	1.472e+07 −	1.450e+07 −	1.333e+07 −	1.248e+07 −	1.466e+07 −	6.046e+06 ≈	8.933e+06 −	5.391e+06
MaF5	3	8.455e+01 −	3.839e+01 −	3.405e+01 −	1.602e+01 −	3.399e+01 −	2.539e+01 −	1.205e+01 −	6.497e+01 −	7.048e+00
	7	5.690e+02 −	2.684e+02 −	5.636e+01 ≈	6.021e+01 ≈	5.873e+01 −	5.217e+01 ≈	5.158e+01 ≈	2.261e+02 −	5.616e+01
	11	6.574e+03 −	2.304e+03 −	4.108e+02 ≈	4.707e+02 ≈	5.115e+02 −	3.770e+02 ≈	5.380e+02 −	2.036e+03 −	4.175e+02
MaF6	3	1.375e+03 −	7.194e+02 −	1.368e+03 −	7.097e+02 −	7.577e+02 −	9.932e+02 −	9.101e+01 −	4.522e+02 −	6.950e+01
	7	1.348e+03 −	5.393e+02 −	1.322e+03 −	1.117e+03 −	7.548e+02 −	1.053e+03 −	1.664e+02 −	4.391e+02 −	1.079e+02
	11	1.308e+03 −	4.739e+02 −	1.300e+03 −	1.083e+03 −	7.568e+02 −	1.010e+03 −	2.263e+02 −	5.224e+02 −	1.296e+02
MaF7	3	1.109e+01 −	9.264e+00 +	7.943e+00 +	7.702e+00 +	1.059e+01 −	7.641e+00 +	1.047e+01 −	1.064e+01 −	9.680e+00
	7	2.700e+01 −	2.435e+01 −	2.182e+01 ≈	2.526e+01 −	2.566e+01 −	2.704e+01 −	2.491e+01 −	2.330e+01 −	2.149e+01
	11	4.340e+01 −	4.053e+01 −	3.469e+01 ≈	3.902e+01 −	4.118e+01 −	4.350e+01 −	3.981e+01 −	3.866e+01 −	3.431e+01
MaF10	3	2.398e+00 −	2.130e+00 −	2.096e+00 −	2.098e+00 −	2.253e+00 −	1.616e+00 +	2.217e+00 −	1.867e+00 ≈	1.925e+00
	7	3.028e+00 −	2.920e+00 ≈	2.842e+00 +	2.797e+00 +	2.964e+00 −	2.676e+00 +	2.991e+00 −	2.908e+00 ≈	2.906e+00
	11	3.689e+00 −	3.627e+00 ≈	3.522e+00 +	3.476e+00 +	3.629e+00 ≈	3.512e+00 +	3.646e+00 ≈	3.611e+00 ≈	3.642e+00
MaF11	3	1.085e+00 −	7.902e-01 ≈	8.327e-01 −	7.320e-01 ≈	8.571e-01 −	7.462e-01 ≈	7.045e-01 ≈	8.361e-01 −	7.613e-01
	7	3.130e+00 −	2.217e+00 ≈	1.662e+00 +	1.694e+00 +	1.789e+00 +	2.105e+00 ≈	1.861e+00 +	2.359e+00 ≈	2.374e+00
	11	5.570e+00 −	3.792e+00 +	3.104e+00 +	3.143e+00 +	3.058e+00 +	4.521e+00 ≈	3.315e+00 +	4.225e+00 ≈	4.455e+00
MaF12	3	1.145e+00 −	9.146e-01 −	1.023e+00 −	9.447e-01 −	1.018e+00 −	9.655e-01 −	7.651e-01 −	9.212e-01 −	6.323e-01
	7	4.503e+00 −	3.298e+00 ≈	3.741e+00 −	3.900e+00 −	3.692e+00 −	3.890e+00 −	3.611e+00 −	3.679e+00 −	3.139e+00
	11	1.062e+01 −	7.492e+00 +	8.982e+00 −	9.158e+00 −	8.753e+00 −	9.180e+00 −	8.906e+00 −	8.787e+00 −	8.146e+00
MaF13	3	4.131e+00 −	2.777e+00 −	3.912e+00 −	1.074e+00 ≈	2.605e+00 −	2.484e+00 −	1.140e+00 −	2.573e+00 −	9.993e-01
	7	5.346e+04 −	2.652e+03 −	1.819e+04 −	8.435e+00 ≈	4.343e+03 −	4.750e+02 −	4.848e+00 +	4.901e+04 −	5.533e+01
	11	1.021e+05 −	3.671e+03 −	2.566e+04 −	9.038e+00 ≈	5.387e+03 −	7.201e+02 −	4.465e+00 +	9.104e+04 −	8.259e+01
$+/-/\approx$		0/33/0	4/24/5	5/23/5	5/21/7	2/29/2	4/24/5	5/23/5	0/28/5	-

TABLE S-II
AVERAGE IGD VALUES FOR $D = \{10, 20, 30\}$ (21 TRIALS).

(a) $D = 10$

	M	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO	CPS-MOEA	CSEA	MCEA/D	AOF/DE	ASF/DE
MaF1	3	4.495e-02 +	1.603e-01 −	1.181e-01 −	1.001e-01 −	2.728e-01 −	1.177e-01 −	9.165e-02 −	5.639e-02 ≈	5.767e-02
	7	2.624e-01 −	4.177e-01 −	3.295e-01 −	2.898e-01 −	3.719e-01 −	2.514e-01 ≈	3.333e-01 −	2.920e-01 −	2.402e-01
	11	3.844e-01 −	3.468e-01 −	4.071e-01 −	3.468e-01 −	3.628e-01 −	2.659e-01 +	3.528e-01 −	2.961e-01 ≈	3.046e-01
MaF2	3	3.060e-02 +	4.003e-02 ≈	5.058e-02 −	4.674e-02 −	4.720e-02 −	4.921e-02 −	3.674e-02 +	3.324e-02 +	4.028e-02
	7	2.392e-01 ≈	2.139e-01 +	2.194e-01 +	1.741e-01 +	1.603e-01 +	2.624e-01 −	2.213e-01 +	2.507e-01 −	2.375e-01
	11	3.505e-01 ≈	2.960e-01 +	3.175e-01 +	2.461e-01 +	2.680e-01 +	3.591e-01 ≈	3.412e-01 ≈	3.503e-01 ≈	3.603e-01
MaF3	3	4.506e+05 −	1.867e+05 −	2.146e+05 −	1.553e+04 +	9.264e+04 −	1.276e+05 ≈	1.983e+04 +	4.392e+05 −	3.215e+04
	7	6.690e+04 −	2.558e+04 −	1.432e+04 −	8.181e+03 ≈	6.618e+03 −	1.055e+04 ≈	1.671e+03 ≈	6.848e+04 −	2.855e+03
	11	7.953e-02 +	1.554e-01 ≈	8.126e-02 +	1.273e-01 +	1.347e-01 +	1.561e-01 ≈	1.308e-01 +	1.690e-01 −	1.529e-01
MaF4	3	4.918e+02 ≈	1.043e+03 −	9.363e+02 −	3.301e+02 +	7.882e+02 −	3.326e+02 +	5.142e+02 ≈	5.245e+02 ≈	5.678e+02
	7	1.322e+03 +	3.693e+03 −	3.519e+03 ≈	1.472e+03 +	3.561e+03 −	1.177e+03 +	2.474e+03 ≈	2.396e+03 ≈	2.628e+03
	11	2.539e+02 −	1.284e+02 −	2.418e+02 −	1.173e+02 ≈	1.335e+02 −	2.059e+02 −	1.093e+02 ≈	1.146e+02 −	1.099e+02
MaF5	3	8.428e-01 +	2.672e+00 −	1.110e+00 +	1.396e+00 +	2.431e+00 ≈	1.617e+00 +	2.467e+00 ≈	3.420e+00 −	2.326e+00
	7	1.116e+01 +	2.792e+01 −	1.464e+01 +	1.904e+01 ≈	2.047e+01 −	1.440e+01 +	2.005e+01 ≈	2.601e+01 −	1.881e+01
	11	1.073e+02 +	3.183e+02 −	1.443e+02 +	1.994e+02 −	1.876e+02 −	1.901e+02 ≈	2.055e+02 −	2.902e+02 −	1.691e+02
MaF6	3	7.444e-01 −	4.673e-01 −	1.450e+00 −	1.185e+00 −	1.056e+01 −	9.895e-01 −	4.515e-01 −	2.905e-01 −	8.992e-02
	7	9.861e-02 −	3.368e-01 −	4.433e-01 −	2.295e-01 −	1.381e+00 −	2.627e-01 −	2.302e-01 −	1.520e-01 −	5.106e-02
	11	5.194e-03 −	2.345e-03 +	3.211e-03 +	3.741e-03 +	1.892e-03 +	3.658e-03 +	2.622e-03 +	4.084e-03 +	4.802e-03
MaF7	3	1.154e-01 +	1.119e+00 −	1.085e+00 −	2.174e+00 −	4.271e+00 −	1.069e+00 −	5.973e-01 −	2.173e+00 −	3.995e-01
	7	6.546e-01 +	4.944e+00 −	1.461e+00 −	3.698e+00 −	3.061e+00 −	6.546e+00 −	1.168e+00 ≈	1.116e+00 ≈	1.153e+00
	11	-	-	-	-	-	-	-	-	-
MaF10	3	1.693e+00 +	2.008e+00 −	1.896e+00 −	1.689e+00 +	2.253e+00 −	1.599e+00 +	2.173e+00 −	1.744e+00 +	1.825e+00
	7	2.593e+00 +	2.821e+00 ≈	2.675e+00 +	2.562e+00 +	2.950e+00 −	2.540e+00 +	2.963e+00 −	2.736e+00 +	2.823e+00
	11	-	-	-	-	-	-	-	-	-
MaF11	3	3.108e-01 +	6.672e-01 +	5.939e-01 +	4.749e-01 +	7.072e-01 +	4.331e-01 +	6.615e-01 +	6.743e-01 ≈	7.563e-01
	7	8.145e-01 +	2.501e+00 −	1.127e+00 +	1.164e+00 +	1.688e+00 +	1.461e+00 +	1.985e+00 ≈	2.655e+00 −	2.071e+00
	11	-	-	-	-	-	-	-	-	-
MaF12	3	5.052e-01 +	7.510e-01 −	6.705e-01 −	5.082e-01 +	6.888e-01 −	5.466e-01 +	6.461e-01 ≈	6.462e-01 ≈	6.200e-01
	7	2.494e+00 +	3.663e+00 +	2.740e+00 +	3.023e+00 +	3.557e+00 +	3.936e+00 ≈	3.666e+00 +	3.802e+00 +	4.146e+00
	11	-	-	-	-	-	-	-	-	-
MaF13	3	2.324e-01 +	7.099e-01 −	5.459e-01 ≈	5.167e-01 ≈	1.124e+00 −	4.235e-01 +	5.878e-01 ≈	6.037e-01 ≈	6.202e-01
	7	5.802e+01 −	2.370e+00 +	3.435e+00 +	9.959e-01 +	8.671e+00 ≈	8.805e-01 +	9.808e-01 +	8.154e+01 −	2.895e+01
	11	7.362e+01 −	3.507e+00 +	8.758e+00 ≈	1.287e+00 +	1.360e+01 ≈	9.307e-01 +	1.044e+00 +	9.715e+01 −	3.323e+01
$+/-/\approx$		16/10/3	7/19/3	12/14/3	16/9/4	7/19/3	14/8/7	9/9/11	5/15/9	-

(b) $D = 20$

	M	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO	CPS-MOEA	CSEA	MCEA/D	AOF/DE	ASF/DE
MaF1	3	1.642e-01 −	1.897e-01 −	5.006e-01 −	2.357e-01 −	7.131e-01 −	2.467e-01 −	1.942e-01 −	9.655e-02 −	8.535e-02
	7	4.343e-01 −	4.967e-01 −	5.272e-01 −	5.133e-01 −	9.785e-01 −	4.942e-01 −	4.855e-01 −	3.802e-01 −	3.233e-01
	11	5.263e-01 −	5.710e-01 −	5.428e-01 −	5.294e-01 −	8.723e-01 −	5.182e-01 −	5.374e-01 −	4.956e-01 −	4.117e-01
MaF2	3	4.404e-02 +	5.797e-02 ≈	9.306e-02 −	8.066e-02 −	8.999e-02 −	8.155e-02 −	5.397e-02 ≈	4.867e-02 +	5.485e-02
	7	2.159e-01 +	2.163e-01 +	2.235e-01 +	1.695e-01 +	1.661e-01 +	2.587e-01 ≈	2.210e-01 +	2.508e-01 ≈	2.471e-01
	11	3.482e-01 +	3.210e-01 +	3.376e-01 +	2.932e-01 +	2.909e-01 +	3.900e-01 ≈	3.826e-01 ≈	4.011e-01 ≈	3.962e-01
MaF3	3	3.768e+06 −	1.258e+06 −	1.917e+06 −	2.699e+05 ≈	9.640e+05 −	2.019e+06 −	1.306e+05 ≈	3.615e+06 −	2.069e+05
	7	2.043e+06 −	1.384e+06 −	9.289e+05 −	9.270e+05 −	5.572e+05 −	1.313e+06 −	9.225e+04 ≈	2.089e+06 −	9.891e+04
	11	9.148e+05 −	6.560e+05 −	3.737e+05 −	5.127e+05 −	2.019e+05 −	5.160e+05 −	4.763e+04 −	9.609e+05 −	3.933e+04
MaF4	3	2.618e+03 −	3.850e+03 −	3.269e+03 −	1.477e+03 ≈	2.761e+03 −	1.438e+03 ≈	1.573e+03 ≈	2.234e+03 −	1.630e+03
	7	2.194e+04 ≈	4.531e+04 −	4.121e+04 −	2.546e+04 ≈	3.601e+04 −	1.289e+04 +	1.957e+04 ≈	2.982e+04 ≈	2.181e+04
	11	1.586e+05 +	4.220e+05 −	4.232e+05 −	1.850e+05 +	3.399e+05 −	1.146e+05 +	2.022e+05 +	3.200e+05 ≈	2.557e+05
MaF5	3	2.403e+00 +	4.312e+00 ≈	3.131e+00 ≈	2.116e+00 +	4.330e+00 ≈	2.296e+00 +	3.282e+00 ≈	7.010e+00 −	3.665e+00
	7	1.682e+01 +	4.480e+01 −	1.853e+01 +	2.755e+01 ≈	3.097e+01 −	1.785e+01 +	3.028e+01 −	4.550e+01 −	2.485e+01
	11	1.556e+02 +	4.146e+02 −	1.791e+02 +	2.656e+02 −	3.070e+02 −	2.297e+02 ≈	3.178e+02 −	4.262e+02 −	2.407e+02
MaF6	3	2.148e+01 −	1.777e+00 −	2.475e+01 −	6.750e+00 −	4.462e+01 −	9.947e+00 −	3.156e+00 −	1.891e+00 −	6.864e-01
	7	1.363e+01 −	2.406e+00 −	3.082e+01 −	1.866e+01 −	3.154e+01 −	4.892e+00 −	2.552e+00 −	1.348e+00 −	7.090e-01
	11	4.953e+00 −	1.540e+00 −	1.140e+01 −	1.464e+01 −	1.816e+01 −	3.096e+00 −	2.377e+00 −	2.880e+00 −	7.510e-01
MaF7	3	5.040e-01 +	2.077e+00 −	1.881e+00 −	3.097e+00 −	6.588e+00 −	2.931e+00 −	4.045e+00 −	2.221e+00 −	7.593e-01
	7	7.876e-01 +	1.102e+01 −	3.923e+00 −	1.125e+01 −	1.307e+01 −	1.607e+01 −	6.219e+00 −	2.518e+00 ≈	1.293e+00
	11	1.284e+00 +	1.897e+01 −	4.172e+00 −	1.665e+01 −	1.823e+01 −	2.736e+01 −	5.857e+00 −	2.418e+00 −	1.723e+00
MaF10	3	1.784e+00 ≈	1.977e+00 −	1.946e+00 −	1.732e+00 +	2.248e+00 −	1.602e+00 +	2.195e+00 −	1.757e+00 +	1.858e+00
	7	2.654e+00 +	2.810e+00 +	2.687e+00 +	2.658e+00 +	2.961e+00 −	2.653e+00 +	2.982e+00 −	2.807e+00 +	2.870e+00
	11	3.416e+00 +	3.559e+00 +	3.435e+00 +	3.392e+00 +	3.624e+00 −	3.466e+00 +	3.641e+00 −	3.564e+00 ≈	3.589e+00
MaF11	3	5.249e-01 +	6.159e-01 +	6.889e-01 +	5.987e-01 +	7.716e-01 ≈	4.988e-01 +	7.062e-01 +	7.700e-01 ≈	8.232e-01
	7	1.036e+00 +	2.434e+00 −	1.405e+00 +	1.399e+00 +	1.843e+00 +	1.671e+00 +	2.027e+00 ≈	2.187e+00 ≈	2.156e+00
	11	1.505e+00 +	4.172e+00 ≈	2.304e+00 +	2.637e+00 +	3.188e+00 +	4.034e+00 ≈	3.336e+00 +	4.303e+00 ≈	4.330e+00
MaF12	3	6.930e-01 ≈	7.854e-01 −	8.291e-01 −	6.986e-01 ≈	7.936e-01 −	6.833e-01 ≈	7.406e-01 −	7.786e-01 −	6.541e-01
	7	2.811e+00 +	3.466e+00 +	3.040e+00 +	3.700e+00 ≈	3.774e+00 ≈	4.066e+00 ≈	3.777e+00 ≈	3.922e+00 ≈	3.921e+00
	11	6.744e+00 +	8.591e+00 +	7.463e+00 +	8.785e+00 ≈	8.921e+00 ≈	9.782e+00 ≈	9.211e+00 ≈	9.604e+00 ≈	9.627e+00
MaF13	3	3.624e-01 +	9.723e-01 −	9.953e-01 −	6.782e-01 ≈	2.069e+00 −	5.714e-01 ≈	8.719e-01 −	9.699e-01 −	5.893e-01
	7	1.429e+03 −	8.425e+00 +	1.300e+01 ≈	3.305e+00 +	1.127e+02 −	1.506e+00 +	1.704e+00 +	6.880e+02 −	4.485e+01
	11	8.278e+03 −	9.690e+00 +	2.348e+01 ≈	2.485e+00 +	1.725e+02 +	2.602e+00 +	1.575e+00 +	7.488e+03 −	2.881e+02
$+/-/\approx$		18/12/3	9/21/3	11/19/3	13/13/7	6/24/3	11/13/9	6/17/10	3/19/11	-

(c) $D = 30$

	M	K-RVEA	HeE-MOEA	EDN-ARMOEA	ADSAPSO	CPS-MOEA	CSEA	MCEA/D	AOF/DE	ASF/DE
MaF1	3	5.533e-01 $-$	2.141e-01 $-$	1.187e+00 $-$	4.702e-01 $-$	1.119e+00 $-$	4.699e-01 $-$	2.888e-01 $-$	1.444e-01 $\approx$	1.265e-01
	7	8.333e-01 $-$	5.425e-01 $-$	1.326e+00 $-$	9.544e-01 $-$	1.759e+00 $-$	8.126e-01 $-$	5.632e-01 $-$	4.968e-01 $-$	3.573e-01
	11	1.103e+00 $-$	6.312e-01 $-$	1.084e+00 $-$	8.473e-01 $-$	1.779e+00 $-$	7.598e-01 $-$	6.079e-01 $-$	6.016e-01 $-$	4.465e-01
MaF2	3	9.421e-02 $-$	6.907e-02 $\approx$	1.474e-01 $-$	1.109e-01 $-$	1.381e-01 $-$	1.156e-01 $-$	7.114e-02 $\approx$	5.626e-02 $+$	6.588e-02
	7	2.205e-01 $+$	2.293e-01 $+$	2.366e-01 $+$	1.952e-01 $+$	1.891e-01 $+$	2.780e-01 $-$	2.444e-01 $+$	2.676e-01 $\approx$	2.611e-01
	11	3.303e-01 $+$	3.327e-01 $+$	3.325e-01 $+$	2.848e-01 $+$	2.929e-01 $+$	3.911e-01 $\approx$	3.808e-01 $\approx$	3.960e-01 $\approx$	3.945e-01
MaF3	3	9.234e+06 $-$	3.834e+06 $-$	5.637e+06 $-$	7.605e+05 $\approx$	2.759e+06 $-$	5.672e+06 $-$	3.959e+05 $\approx$	8.876e+06 $-$	6.794e+05
	7	6.605e+06 $-$	4.369e+06 $-$	3.834e+06 $-$	4.990e+06 $-$	2.289e+06 $-$	5.420e+06 $-$	3.223e+05 $\approx$	6.497e+06 $-$	3.279e+05
	11	4.664e+06 $-$	3.592e+06 $-$	2.450e+06 $-$	3.256e+06 $-$	1.425e+06 $-$	3.644e+06 $-$	1.957e+05 $\approx$	4.575e+06 $-$	2.254e+05
MaF4	3	4.918e+03 $-$	6.734e+03 $-$	5.714e+03 $-$	2.915e+03 $\approx$	4.833e+03 $-$	2.631e+03 $\approx$	2.219e+03 $+$	4.484e+03 $-$	3.159e+03
	7	5.225e+04 $-$	9.009e+04 $-$	8.538e+04 $-$	6.691e+04 $-$	7.291e+04 $-$	3.721e+04 $\approx$	3.895e+04 $\approx$	5.791e+04 $-$	4.282e+04
	11	6.223e+05 $\approx$	1.169e+06 $-$	1.073e+06 $-$	8.187e+05 $\approx$	9.034e+05 $-$	4.142e+05 $+$	5.653e+05 $\approx$	9.529e+05 $-$	7.030e+05
MaF5	3	3.946e+00 $\approx$	5.145e+00 $-$	5.730e+00 $-$	2.885e+00 $+$	6.095e+00 $-$	2.803e+00 $+$	4.169e+00 $\approx$	9.221e+00 $-$	4.032e+00
	7	2.538e+01 $+$	5.313e+01 $-$	2.358e+01 $+$	3.056e+01 $\approx$	3.542e+01 $-$	2.071e+01 $+$	3.384e+01 $-$	5.274e+01 $-$	2.901e+01
	11	2.841e+02 $-$	5.008e+02 $-$	2.165e+02 $+$	3.186e+02 $-$	3.478e+02 $-$	2.496e+02 $+$	3.606e+02 $-$	3.411e+02 $-$	2.810e+02
MaF6	3	6.742e+01 $-$	4.698e+00 $-$	9.624e+01 $-$	2.563e+01 $-$	8.305e+01 $-$	3.039e+01 $-$	6.633e+00 $-$	1.074e+01 $-$	2.463e+00
	7	6.786e+01 $-$	8.061e+00 $-$	9.498e+01 $-$	6.458e+01 $-$	6.736e+01 $-$	2.836e+01 $-$	6.440e+00 $-$	7.858e+00 $-$	2.233e+00
	11	4.181e+01 $-$	7.106e+00 $-$	7.086e+01 $-$	5.540e+01 $-$	5.586e+01 $-$	2.315e+01 $-$	1.005e+01 $-$	1.306e+01 $-$	4.740e+00
MaF7	3	1.262e+00 $+$	2.732e+00 $-$	2.904e+00 $-$	4.004e+00 $-$	7.859e+00 $-$	3.759e+00 $-$	7.061e+00 $-$	4.423e+00 $-$	1.640e+00
	7	1.069e+00 $+$	1.364e+01 $-$	5.170e+00 $-$	1.526e+01 $-$	1.736e+01 $-$	1.964e+01 $-$	1.296e+01 $-$	4.739e+00 $-$	1.498e+00
	11	3.068e+00 $\approx$	2.378e+01 $-$	8.661e+00 $-$	2.347e+01 $-$	2.737e+01 $-$	3.286e+01 $-$	1.534e+01 $-$	8.987e+00 $-$	1.961e+00
MaF10	3	1.950e+00 $\approx$	2.003e+00 $-$	1.993e+00 $-$	1.938e+00 $\approx$	2.236e+00 $-$	1.575e+00 $+$	2.190e+00 $-$	1.883e+00 $\approx$	1.880e+00
	7	2.782e+00 $+$	2.839e+00 $\approx$	2.738e+00 $+$	2.686e+00 $+$	2.960e+00 $-$	2.615e+00 $+$	2.980e+00 $-$	2.780e+00 $+$	2.869e+00
	11	3.458e+00 $+$	3.566e+00 $+$	3.453e+00 $+$	3.423e+00 $+$	3.625e+00 $\approx$	3.500e+00 $+$	3.656e+00 $-$	3.586e+00 $\approx$	3.604e+00
MaF11	3	6.745e-01 $+$	6.503e-01 $+$	7.440e-01 $+$	6.768e-01 $+$	8.076e-01 $\approx$	5.198e-01 $+$	7.145e-01 $+$	7.826e-01 $\approx$	8.177e-01
	7	1.455e+00 $+$	2.207e+00 $\approx$	1.456e+00 $+$	1.537e+00 $+$	1.781e+00 $+$	1.955e+00 $\approx$	1.874e+00 $+$	2.402e+00 $\approx$	2.286e+00
	11	2.634e+00 $+$	4.081e+00 $\approx$	2.490e+00 $+$	2.921e+00 $+$	3.252e+00 $+$	4.175e+00 $\approx$	3.308e+00 $+$	4.563e+00 $\approx$	4.275e+00
MaF12	3	8.361e-01 $-$	8.207e-01 $-$	8.904e-01 $-$	7.254e-01 $-$	8.563e-01 $-$	7.641e-01 $-$	7.126e-01 $-$	7.870e-01 $-$	6.417e-01
	7	2.877e+00 $+$	3.668e+00 $\approx$	3.238e+00 $+$	3.815e+00 $\approx$	3.726e+00 $\approx$	4.071e+00 $\approx$	3.712e+00 $\approx$	3.828e+00 $\approx$	3.818e+00
	11	6.625e+00 $+$	8.502e+00 $+$	8.080e+00 $+$	8.693e+00 $+$	8.982e+00 $+$	9.929e+00 $\approx$	9.043e+00 $+$	9.767e+00 $\approx$	9.736e+00
MaF13	3	6.094e-01 $+$	9.169e-01 $-$	1.631e+00 $-$	7.609e-01 $-$	2.210e+00 $-$	7.255e-01 $-$	9.616e-01 $-$	1.323e+00 $-$	6.488e-01
	7	7.710e+03 $-$	3.107e+01 $\approx$	8.471e+01 $\approx$	3.439e+00 $+$	3.193e+02 $\approx$	4.030e+00 $+$	2.940e+00 $+$	7.607e+03 $-$	5.891e+02
	11	1.544e+04 $-$	3.290e+01 $\approx$	1.848e+02 $-$	4.325e+00 $+$	1.190e+03 $-$	3.316e+00 $+$	2.286e+00 $+$	1.422e+04 $-$	1.325e+02
$+/-/\approx$		13/16/4	5/21/7	11/21/1	11/16/6	5/24/4	10/16/7	8/16/9	2/21/10	-

TABLE S-III
AVERAGE IGD VALUES OF ASF/DE/RAND/1 AND ASF/DE/LHS (21 TRIALS).

(a) $D = 10$

	M	ASF/DE/rand/1	ASF/DE/LHS
MaF1	3	7.050e-02 −	1.124e-01 −
	7	2.184e-01 +	2.917e-01 −
	11	2.818e-01 +	3.271e-01 −
MaF2	3	4.309e-02 −	3.604e-02 +
	7	2.203e-01 +	2.133e-01 +
	11	3.614e-01 ≈	3.449e-01 +
MaF3	3	3.658e+04 −	4.024e+04 −
	7	5.267e+03 −	6.256e+03 −
	11	1.509e-01 ≈	1.489e-01 ≈
MaF4	3	5.964e+02 ≈	7.646e+02 −
	7	3.817e+03 −	3.724e+03 −
	11	1.056e+02 +	1.196e+02 −
MaF5	3	1.560e+00 +	2.099e+00 ≈
	7	1.924e+01 ≈	2.048e+01 −
	11	1.811e+02 ≈	1.961e+02 −
MaF6	3	5.629e-02 +	1.749e+00 −
	7	8.580e-02 −	3.680e-01 −
	11	5.064e-03 −	4.169e-03 +
MaF7	3	3.322e-01 ≈	2.108e+00 −
	7	1.044e+00 +	1.589e+00 −
	11	-	-
MaF10	3	1.763e+00 +	2.191e+00 −
	7	2.794e+00 ≈	2.731e+00 +
	11	-	-
MaF11	3	8.293e-01 −	1.033e+00 −
	7	2.012e+00 ≈	3.051e+00 −
	11	-	-
MaF12	3	6.602e-01 ≈	8.933e-01 −
	7	4.352e+00 ≈	4.831e+00 −
	11	-	-
MaF13	3	6.694e-01 ≈	3.855e-01 +
	7	4.905e+01 ≈	1.417e+00 +
	11	3.929e+01 −	2.103e+00 +
+/−/≈		8/9/12	8/19/2

(b) $D = 20$

	M	ASF/DE/rand/1	ASF/DE/LHS
MaF1	3	9.231e-02 ≈	2.671e-01 −
	7	2.864e-01 +	5.347e-01 −
	11	4.009e-01 ≈	5.821e-01 −
MaF2	3	6.253e-02 −	5.374e-02 ≈
	7	2.313e-01 +	2.178e-01 +
	11	3.885e-01 ≈	3.576e-01 +
MaF3	3	1.872e+05 ≈	5.088e+05 −
	7	1.050e+05 −	2.005e+05 −
	11	5.114e+04 −	8.325e+04 −
MaF4	3	1.770e+03 ≈	2.770e+03 −
	7	2.680e+04 ≈	3.727e+04 −
	11	2.493e+05 ≈	3.164e+05 −
MaF5	3	3.164e+00 ≈	3.261e+00 ≈
	7	2.632e+01 ≈	2.987e+01 −
	11	2.361e+02 ≈	3.115e+02 −
MaF6	3	2.490e-01 +	9.987e+00 −
	7	5.046e-01 ≈	8.319e+00 −
	11	8.012e-01 ≈	4.676e+00 −
MaF7	3	7.559e-01 ≈	5.115e+00 −
	7	1.178e+00 +	1.110e+01 −
	11	1.519e+00 +	1.413e+01 −
MaF10	3	1.741e+00 +	2.156e+00 −
	7	2.774e+00 +	2.737e+00 +
	11	3.517e+00 +	3.393e+00 +
MaF11	3	8.225e-01 ≈	1.057e+00 −
	7	1.896e+00 +	3.136e+00 −
	11	3.249e+00 +	5.611e+00 −
MaF12	3	5.926e-01 +	8.745e-01 −
	7	4.320e+00 −	4.890e+00 −
	11	1.051e+01 −	1.127e+01 −
MaF13	3	5.814e-01 ≈	4.753e-01 +
	7	1.989e+02 ≈	1.920e+00 +
	11	1.627e+03 ≈	2.564e+00 +
+/−/≈		11/5/17	7/24/2

(c) $D = 30$

	M	ASF/DE/rand/1	ASF/DE/LHS
MaF1	3	1.195e-01 ≈	3.968e-01 −
	7	3.286e-01 +	6.914e-01 −
	11	4.374e-01 ≈	7.060e-01 −
MaF2	3	7.790e-02 −	7.445e-02 −
	7	2.483e-01 +	2.343e-01 +
	11	3.849e-01 ≈	3.449e-01 +
MaF3	3	5.788e+05 ≈	2.573e+06 −
	7	3.275e+05 ≈	9.664e+05 −
	11	2.191e+05 ≈	9.002e+05 −
MaF4	3	2.764e+03 ≈	5.376e+03 −
	7	4.027e+04 ≈	7.564e+04 −
	11	7.443e+05 ≈	9.465e+05 −
MaF5	3	3.250e+00 +	4.788e+00 −
	7	2.926e+01 ≈	3.534e+01 −
	11	2.730e+02 ≈	3.422e+02 −
MaF6	3	6.455e-01 +	2.184e+01 −
	7	1.484e+00 +	2.017e+01 −
	11	4.586e+00 ≈	1.386e+01 −
MaF7	3	9.511e-01 +	6.446e+00 −
	7	1.315e+00 +	1.568e+01 −
	11	1.661e+00 +	2.385e+01 −
MaF10	3	1.766e+00 +	2.189e+00 −
	7	2.750e+00 +	2.745e+00 +
	11	3.515e+00 +	3.396e+00 +
MaF11	3	8.702e-01 −	1.073e+00 −
	7	1.838e+00 +	3.139e+00 −
	11	3.377e+00 +	5.602e+00 −
MaF12	3	5.768e-01 +	9.073e-01 −
	7	4.025e+00 ≈	4.866e+00 −
	11	1.014e+01 ≈	1.118e+01 −
MaF13	3	6.421e-01 ≈	4.938e-01 +
	7	2.571e+03 ≈	2.025e+00 +
	11	3.158e+03 ≈	2.629e+00 +
+/−/≈		14/2/17	7/26/0

(d) $D = 50$

	M	ASF/DE/rand/1	ASF/DE/LHS
MaF1	3	2.148e-01 ≈	6.103e-01 −
	7	3.954e-01 +	1.050e+00 −
	11	5.158e-01 ≈	1.038e+00 −
MaF2	3	9.594e-02 −	1.106e-01 −
	7	2.737e-01 ≈	2.517e-01 +
	11	3.926e-01 ≈	3.595e-01 +
MaF3	3	2.555e+06 ≈	1.007e+07 −
	7	1.173e+06 ≈	5.703e+06 −
	11	9.279e+05 ≈	3.300e+06 −
MaF4	3	5.632e+03 ≈	1.058e+04 −
	7	1.057e+05 ≈	1.620e+05 −
	11	1.179e+06 ≈	2.082e+06 −
MaF5	3	4.407e+00 +	6.978e+00 −
	7	3.368e+01 +	3.952e+01 ≈
	11	2.803e+02 ≈	3.855e+02 −
MaF6	3	4.968e+00 +	4.256e+01 −
	7	1.096e+01 ≈	4.736e+01 −
	11	2.293e+01 ≈	4.292e+01 −
MaF7	3	1.699e+00 +	7.973e+00 −
	7	1.728e+00 +	1.929e+01 −
	11	2.018e+00 +	3.045e+01 −
MaF10	3	1.812e+00 +	2.224e+00 −
	7	2.834e+00 +	2.757e+00 +
	11	3.537e+00 +	3.398e+00 +
MaF11	3	8.567e-01 ≈	1.088e+00 −
	7	1.824e+00 +	3.151e+00 −
	11	3.279e+00 +	5.575e+00 −
MaF12	3	6.412e-01 ≈	9.599e-01 −
	7	3.862e+00 ≈	4.848e+00 −
	11	9.862e+00 −	1.114e+01 −
MaF13	3	7.553e-01 ≈	5.369e-01 +
	7	7.427e+02 ≈	2.126e+00 +
	11	8.422e+03 −	2.873e+00 +
+/−/≈		12/3/18	7/25/1

(e) $D = 100$

	M	ASF/DE/rand/1	ASF/DE/LHS
MaF1	3	6.045e-01 −	1.167e+00 −
	7	8.060e-01 −	2.044e+00 −
	11	9.228e-01 −	2.079e+00 −
MaF2	3	1.462e-01 −	1.852e-01 −
	7	3.284e-01 ≈	3.205e-01 +
	11	4.209e-01 ≈	3.893e-01 +
MaF3	3	8.531e+06 ≈	4.823e+07 −
	7	5.711e+06 ≈	2.688e+07 −
	11	4.992e+06 ≈	1.852e+07 −
MaF4	3	1.210e+04 −	2.041e+04 −
	7	1.994e+05 −	3.124e+05 −
	11	2.859e+06 ≈	4.556e+06 −
MaF5	3	5.795e+00 ≈	1.072e+01 −
	7	4.429e+01 ≈	4.637e+01 +
	11	3.258e+02 +	4.754e+02 −
MaF6	3	5.962e+01 −	1.065e+02 −
	7	7.378e+01 −	1.150e+02 −
	11	1.120e+02 −	1.016e+02 −
MaF7	3	4.290e+00 +	9.256e+00 −
	7	6.255e+00 +	2.297e+01 −
	11	8.812e+00 +	3.671e+01 −
MaF10	3	1.768e+00 +	2.208e+00 −
	7	2.852e+00 +	2.748e+00 +
	11	3.570e+00 +	3.394e+00 +
MaF11	3	8.379e-01 ≈	1.085e+00 −
	7	1.849e+00 +	3.140e+00 −
	11	3.354e+00 ≈	5.570e+00 −
MaF12	3	6.798e-01 ≈	9.974e-01 −
	7	3.749e+00 −	4.823e+00 −
	11	9.606e+00 −	1.108e+01 −
MaF13	3	1.157e+00 ≈	5.765e-01 +
	7	2.613e+03 ≈	2.268e+00 +
	11	1.581e+03 ≈	3.189e+00 +
+/−/≈		8/11/14	8/25/0

(f) $D = 150$

	M	ASF/DE/rand/1	ASF/DE/LHS
MaF1	3	1.124e+00 −	1.972e+00 −
	7	1.405e+00 −	3.370e+00 −
	11	2.023e+00 −	3.236e+00 −
MaF2	3	1.772e-01 −	2.422e-01 −
	7	3.697e-01 ≈	3.676e-01 ≈
	11	4.731e-01 ≈	4.295e-01 +
MaF3	3	1.583e+07 ≈	9.962e+07 −
	7	1.302e+07 ≈	5.409e+07 −
	11	1.227e+07 ≈	4.760e+07 −
MaF4	3	1.807e+04 ≈	2.791e+04 −
	7	2.902e+05 ≈	4.475e+05 −
	11	4.525e+06 ≈	7.092e+06 −
MaF5	3	8.612e+00 −	1.399e+01 −
	7	4.822e+01 ≈	4.935e+01 ≈
	11	3.609e+02 ≈	4.798e+02 −
MaF6	3	1.117e+02 −	1.743e+02 −
	7	1.455e+02 −	1.992e+02 −
	11	1.692e+02 −	1.503e+02 −
MaF7	3	7.187e+00 +	9.934e+00 −
	7	1.286e+01 +	2.443e+01 −
	11	1.988e+01 +	3.959e+01 −
MaF10	3	1.797e+00 +	2.226e+00 −
	7	2.845e+00 +	2.755e+00 +
	11	3.560e+00 +	3.400e+00 +
MaF11	3	8.303e-01 ≈	1.089e+00 −
	7	1.858e+00 +	3.143e+00 −
	11	3.135e+00 +	5.572e+00 −
MaF12	3	6.493e-01 ≈	1.015e+00 −
	7	3.671e+00 −	4.933e+00 −
	11	9.147e+00 −	1.091e+01 −
MaF13	3	1.270e+00 −	5.993e-01 +
	7	5.377e+02 −	2.524e+00 +
	11	5.957e+03 −	3.406e+00 +
+/−/≈		8/13/12	6/25/2

(g) $D = 200$

	M	ASF/DE/rand/1	ASF/DE/LHS
MaF1	3	1.841e+00 −	2.441e+00 −
	7	2.368e+00 −	4.380e+00 −
	11	3.213e+00 −	4.365e+00 −
MaF2	3	2.244e-01 −	2.857e-01 −
	7	3.755e-01 ≈	4.046e-01 −
	11	4.869e-01 ≈	4.556e-01 +
MaF3	3	4.281e+07 +	1.672e+08 −
	7	2.414e+07 ≈	9.715e+07 −
	11	2.364e+07 ≈	6.403e+07 −
MaF4	3	2.215e+04 ≈	3.619e+04 −
	7	3.821e+05 ≈	6.043e+05 −
	11	5.855e+06 ≈	9.028e+06 −
MaF5	3	1.094e+01 −	1.775e+01 −
	7	5.130e+01 ≈	5.579e+01 ≈
	11	3.878e+02 ≈	5.023e+02 −
MaF6	3	1.767e+02 −	2.227e+02 −
	7	2.039e+02 −	2.432e+02 −
	11	2.518e+02 −	2.015e+02 −
MaF7	3	8.226e+00 +	1.034e+01 −
	7	1.676e+01 +	2.508e+01 −
	11	2.525e+01 +	4.042e+01 −
MaF10	3	1.829e+00 +	2.206e+00 −
	7	2.856e+00 +	2.757e+00 +
	11	3.550e+00 +	3.393e+00 +
MaF11	3	8.539e-01 −	1.086e+00 −
	7	1.883e+00 +	3.139e+00 −
	11	3.441e+00 +	5.571e+00 −
MaF12	3	6.486e-01 ≈	1.006e+00 −
	7	3.520e+00 −	4.718e+00 −
	11	8.824e+00 −	1.105e+01 −
MaF13	3	1.411e+00 −	6.049e-01 +
	7	1.973e+03 −	2.516e+00 +
	11	5.374e+02 −	3.567e+00 +
+/−/≈		9/14/10	6/26/1

TABLE S-IV
AVERAGE IGD VALUES OF ASF/DE/$\mathcal{A}$ AND ASF/DE/$\mathcal{A}_i^g$ (21 TRIALS).

(a) $D = 10$

	M	ASF/DE/$\mathcal{A}$	ASF/DE/$\mathcal{A}_i^g$
MaF1	3	6.106e-02 ≈	5.729e-02 ≈
	7	2.296e-01 ≈	2.372e-01 ≈
	11	2.828e-01 +	2.878e-01 +
MaF2	3	4.157e-02 ≈	3.889e-02 +
	7	2.262e-01 ≈	2.346e-01 ≈
	11	3.567e-01 ≈	3.580e-01 ≈
MaF3	3	3.005e+04 ≈	2.769e+04 ≈
	7	3.986e+03 ≈	3.103e+03 ≈
	11	1.287e-01 +	1.364e-01 +
MaF4	3	6.054e+02 ≈	4.528e+02 +
	7	2.961e+03 ≈	2.544e+03 ≈
	11	1.065e+02 ≈	1.088e+02 ≈
MaF5	3	2.232e+00 ≈	2.627e+00 ≈
	7	1.734e+01 +	1.796e+01 ≈
	11	1.655e+02 ≈	1.780e+02 ≈
MaF6	3	9.601e-02 ≈	7.690e-02 ≈
	7	5.443e-02 ≈	5.161e-02 ≈
	11	4.388e-03 +	3.623e-03 +
MaF7	3	4.976e-01 ≈	4.313e-01 ≈
	7	1.079e+00 ≈	1.174e+00 ≈
	11	-	-
MaF10	3	1.845e+00 ≈	1.820e+00 ≈
	7	2.815e+00 ≈	2.805e+00 ≈
	11	-	-
MaF11	3	8.190e-01 ≈	7.113e-01 ≈
	7	2.284e+00 ≈	2.121e+00 ≈
	11	-	-
MaF12	3	6.321e-01 ≈	5.959e-01 ≈
	7	4.041e+00 ≈	3.756e+00 +
	11	-	-
MaF13	3	7.047e-01 ≈	5.412e-01 ≈
	7	4.028e+01 ≈	2.265e+01 +
	11	1.704e+01 ≈	6.143e+00 +
+/−/≈		4/0/25	8/0/21

(b) $D = 20$

	M	ASF/DE/$\mathcal{A}$	ASF/DE/$\mathcal{A}_i^g$
MaF1	3	9.097e-02 ≈	8.230e-02 ≈
	7	3.144e-01 ≈	3.257e-01 ≈
	11	3.947e-01 +	4.195e-01 ≈
MaF2	3	5.886e-02 −	5.014e-02 +
	7	2.421e-01 ≈	2.414e-01 ≈
	11	3.922e-01 ≈	3.881e-01 ≈
MaF3	3	2.035e+05 ≈	2.179e+05 ≈
	7	9.291e+04 ≈	9.216e+04 ≈
	11	4.028e+04 ≈	4.502e+04 ≈
MaF4	3	1.688e+03 ≈	1.639e+03 ≈
	7	1.923e+04 +	2.282e+04 ≈
	11	2.708e+05 ≈	1.720e+05 +
MaF5	3	3.679e+00 ≈	4.131e+00 ≈
	7	2.632e+01 ≈	2.560e+01 ≈
	11	2.443e+02 ≈	2.479e+02 ≈
MaF6	3	8.779e-01 ≈	6.556e-01 ≈
	7	7.579e-01 ≈	2.744e-01 +
	11	8.494e-01 ≈	3.641e-01 +
MaF7	3	7.543e-01 ≈	6.021e-01 ≈
	7	1.236e+00 ≈	1.221e+00 ≈
	11	1.805e+00 ≈	1.778e+00 ≈
MaF10	3	1.865e+00 ≈	1.818e+00 ≈
	7	2.866e+00 ≈	2.832e+00 ≈
	11	3.588e+00 ≈	3.571e+00 ≈
MaF11	3	8.308e-01 ≈	7.755e-01 ≈
	7	2.271e+00 ≈	2.198e+00 ≈
	11	4.209e+00 ≈	3.852e+00 ≈
MaF12	3	5.887e-01 ≈	6.133e-01 ≈
	7	4.072e+00 ≈	3.838e+00 ≈
	11	9.739e+00 ≈	9.122e+00 ≈
MaF13	3	7.044e-01 ≈	5.351e-01 ≈
	7	4.881e+01 ≈	4.058e+00 +
	11	1.371e+02 ≈	7.264e+00 +
+/−/≈		2/1/30	6/0/27

(c) $D = 30$

	M	ASF/DE/$\mathcal{A}$	ASF/DE/$\mathcal{A}_i^g$
MaF1	3	1.357e-01 ≈	1.235e-01 ≈
	7	3.481e-01 ≈	3.576e-01 ≈
	11	4.369e-01 +	4.534e-01 ≈
MaF2	3	6.935e-02 ≈	5.581e-02 +
	7	2.563e-01 ≈	2.525e-01 ≈
	11	3.880e-01 ≈	3.935e-01 ≈
MaF3	3	4.930e+05 ≈	6.349e+05 ≈
	7	3.288e+05 ≈	3.317e+05 ≈
	11	2.050e+05 ≈	2.208e+05 ≈
MaF4	3	2.713e+03 ≈	2.860e+03 ≈
	7	4.247e+04 ≈	3.679e+04 ≈
	11	6.321e+05 ≈	5.640e+05 ≈
MaF5	3	3.919e+00 ≈	4.454e+00 ≈
	7	2.972e+01 ≈	2.724e+01 +
	11	2.823e+02 ≈	2.706e+02 ≈
MaF6	3	2.510e+00 ≈	2.248e+00 ≈
	7	3.024e+00 ≈	1.378e+00 +
	11	4.068e+00 ≈	2.004e+00 +
MaF7	3	1.228e+00 ≈	1.318e+00 ≈
	7	1.857e+00 ≈	2.001e+00 ≈
	11	2.156e+00 ≈	2.306e+00 ≈
MaF10	3	1.854e+00 ≈	1.863e+00 ≈
	7	2.854e+00 ≈	2.856e+00 ≈
	11	3.597e+00 ≈	3.585e+00 ≈
MaF11	3	8.524e-01 ≈	7.642e-01 +
	7	2.419e+00 ≈	2.245e+00 ≈
	11	3.954e+00 ≈	3.957e+00 ≈
MaF12	3	6.402e-01 ≈	6.337e-01 ≈
	7	3.805e+00 ≈	3.601e+00 ≈
	11	9.587e+00 ≈	9.118e+00 +
MaF13	3	6.634e-01 ≈	5.941e-01 ≈
	7	2.310e+02 ≈	2.620e+00 +
	11	3.144e+02 ≈	5.768e+00 +
+/−/≈		1/0/32	8/0/25

(d) $D = 50$

	M	ASF/DE/$\mathcal{A}$	ASF/DE/$\mathcal{A}_i^g$
MaF1	3	2.135e-01 ≈	2.061e-01 ≈
	7	4.145e-01 ≈	4.171e-01 ≈
	11	4.807e-01 +	4.953e-01 ≈
MaF2	3	7.950e-02 ≈	6.931e-02 ≈
	7	2.853e-01 ≈	2.838e-01 ≈
	11	4.003e-01 ≈	3.988e-01 ≈
MaF3	3	2.252e+06 ≈	1.870e+06 ≈
	7	1.248e+06 ≈	1.070e+06 ≈
	11	9.398e+05 ≈	9.447e+05 ≈
MaF4	3	5.734e+03 ≈	5.696e+03 ≈
	7	7.780e+04 ≈	7.413e+04 ≈
	11	1.105e+06 ≈	1.082e+06 ≈
MaF5	3	4.632e+00 +	4.912e+00 −
	7	3.981e+01 ≈	3.525e+01 +
	11	3.242e+02 −	2.954e+02 ≈
MaF6	3	9.600e+00 ≈	9.080e+00 ≈
	7	9.579e+00 ≈	6.785e+00 +
	11	1.615e+01 ≈	9.188e+00 +
MaF7	3	3.633e+00 ≈	3.935e+00 ≈
	7	5.680e+00 −	7.161e+00 −
	11	1.251e+01 −	1.410e+01 −
MaF10	3	1.907e+00 ≈	1.879e+00 ≈
	7	2.877e+00 ≈	2.870e+00 ≈
	11	3.616e+00 ≈	3.610e+00 ≈
MaF11	3	8.354e-01 ≈	7.705e-01 ≈
	7	2.303e+00 ≈	2.304e+00 ≈
	11	4.306e+00 ≈	3.995e+00 ≈
MaF12	3	6.624e-01 ≈	6.401e-01 ≈
	7	3.854e+00 ≈	3.710e+00 ≈
	11	9.410e+00 ≈	8.746e+00 ≈
MaF13	3	7.352e-01 ≈	7.484e-01 ≈
	7	2.835e+02 ≈	7.539e+00 +
	11	1.729e+02 ≈	6.527e+00 +
+/−/≈		2/3/28	5/3/25

(e) $D = 100$

	M	ASF/DE/$\mathcal{A}$	ASF/DE/$\mathcal{A}_i^g$
MaF1	3	4.684e-01 ≈	4.552e-01 ≈
	7	6.887e-01 ≈	7.063e-01 ≈
	11	6.252e-01 +	6.599e-01 ≈
MaF2	3	9.876e-02 ≈	8.911e-02 +
	7	3.491e-01 ≈	3.406e-01 ≈
	11	4.315e-01 ≈	4.263e-01 ≈
MaF3	3	1.091e+07 ≈	7.695e+06 ≈
	7	5.709e+06 ≈	5.543e+06 ≈
	11	5.101e+06 ≈	4.923e+06 ≈
MaF4	3	9.659e+03 ≈	1.086e+04 ≈
	7	1.716e+05 ≈	1.667e+05 ≈
	11	2.601e+06 ≈	2.842e+06 ≈
MaF5	3	5.288e+00 +	5.493e+00 +
	7	4.927e+01 −	4.702e+01 ≈
	11	3.608e+02 ≈	3.705e+02 ≈
MaF6	3	3.068e+01 ≈	3.027e+01 ≈
	7	4.206e+01 ≈	3.525e+01 +
	11	5.849e+01 ≈	4.851e+01 +
MaF7	3	7.491e+00 ≈	7.732e+00 ≈
	7	1.494e+01 ≈	1.700e+01 −
	11	2.551e+01 ≈	2.914e+01 −
MaF10	3	1.933e+00 ≈	1.868e+00 +
	7	2.910e+00 ≈	2.902e+00 ≈
	11	3.623e+00 ≈	3.605e+00 +
MaF11	3	7.983e-01 ≈	7.498e-01 +
	7	2.147e+00 ≈	2.271e+00 ≈
	11	4.341e+00 ≈	4.323e+00 ≈
MaF12	3	6.378e-01 ≈	6.471e-01 ≈
	7	3.562e+00 −	3.366e+00 ≈
	11	8.821e+00 ≈	8.403e+00 ≈
MaF13	3	8.502e-01 +	9.303e-01 ≈
	7	5.045e+02 ≈	7.645e+00 +
	11	3.235e+02 ≈	2.770e+01 ≈
+/−/≈		3/2/28	8/2/23

(f) $D = 150$

	M	ASF/DE/$\mathcal{A}$	ASF/DE/$\mathcal{A}_i^g$
MaF1	3	6.276e-01 ≈	6.113e-01 ≈
	7	9.570e-01 ≈	9.419e-01 ≈
	11	8.286e-01 +	8.752e-01 ≈
MaF2	3	1.191e-01 ≈	1.077e-01 ≈
	7	3.593e-01 ≈	3.528e-01 +
	11	4.710e-01 ≈	4.665e-01 ≈
MaF3	3	2.651e+07 ≈	1.309e+07 ≈
	7	1.357e+07 ≈	1.279e+07 ≈
	11	1.210e+07 ≈	1.225e+07 ≈
MaF4	3	1.461e+04 ≈	1.377e+04 ≈
	7	2.530e+05 ≈	2.226e+05 ≈
	11	3.982e+06 ≈	3.478e+06 ≈
MaF5	3	5.972e+00 +	6.146e+00 +
	7	5.880e+01 ≈	8.144e+01 −
	11	4.035e+02 ≈	3.992e+02 ≈
MaF6	3	4.969e+01 ≈	4.779e+01 ≈
	7	7.071e+01 ≈	6.216e+01 +
	11	9.153e+01 ≈	8.137e+01 ≈
MaF7	3	8.949e+00 ≈	9.020e+00 ≈
	7	2.068e+01 −	2.077e+01 −
	11	3.144e+01 ≈	3.337e+01 ≈
MaF10	3	1.954e+00 ≈	1.877e+00 +
	7	2.887e+00 ≈	2.889e+00 ≈
	11	3.621e+00 ≈	3.609e+00 ≈
MaF11	3	8.066e-01 ≈	7.654e-01 ≈
	7	2.394e+00 ≈	2.293e+00 ≈
	11	3.853e+00 +	4.007e+00 ≈
MaF12	3	6.233e-01 ≈	6.205e-01 +
	7	3.451e+00 ≈	3.273e+00 ≈
	11	8.608e+00 ≈	8.320e+00 ≈
MaF13	3	8.773e-01 ≈	9.038e-01 ≈
	7	1.029e+02 ≈	6.738e+00 +
	11	2.322e+02 ≈	1.193e+01 ≈
+/−/≈		3/1/29	6/2/25

(g) $D = 200$

	M	ASF/DE/$\mathcal{A}$	ASF/DE/$\mathcal{A}_i^g$
MaF1	3	8.598e-01 ≈	7.689e-01 ≈
	7	1.189e+00 ≈	1.182e+00 ≈
	11	1.093e+00 +	1.116e+00 +
MaF2	3	1.296e-01 −	1.207e-01 ≈
	7	3.585e-01 +	3.671e-01 ≈
	11	4.892e-01 ≈	4.816e-01 ≈
MaF3	3	4.394e+07 ≈	2.527e+07 ≈
	7	2.759e+07 ≈	2.424e+07 ≈
	11	2.298e+07 ≈	2.255e+07 ≈
MaF4	3	2.216e+04 ≈	1.957e+04 ≈
	7	3.887e+05 ≈	2.954e+05 ≈
	11	5.964e+06 ≈	4.034e+06 +
MaF5	3	6.837e+00 ≈	6.759e+00 +
	7	5.806e+01 ≈	6.697e+01 ≈
	11	4.419e+02 ≈	4.358e+02 ≈
MaF6	3	6.917e+01 ≈	6.370e+01 ≈
	7	1.080e+02 ≈	9.912e+01 ≈
	11	1.279e+02 ≈	1.127e+02 ≈
MaF7	3	9.568e+00 ≈	9.698e+00 ≈
	7	2.233e+01 ≈	2.277e+01 ≈
	11	3.519e+01 ≈	3.647e+01 −
MaF10	3	1.922e+00 ≈	1.880e+00 ≈
	7	2.896e+00 ≈	2.896e+00 ≈
	11	3.642e+00 ≈	3.633e+00 ≈
MaF11	3	8.092e-01 ≈	7.349e-01 ≈
	7	2.304e+00 ≈	2.350e+00 ≈
	11	4.213e+00 ≈	4.255e+00 ≈
MaF12	3	6.354e-01 ≈	6.033e-01 ≈
	7	3.293e+00 ≈	3.158e+00 ≈
	11	8.211e+00 ≈	8.107e+00 ≈
MaF13	3	1.012e+00 ≈	9.314e-01 ≈
	7	3.554e+02 ≈	7.460e+00 ≈
	11	8.327e+02 ≈	2.221e+01 ≈
+/−/≈		2/1/30	3/1/29