

Multiple Classifiers-Assisted Evolutionary Algorithm Based on Decomposition for High-Dimensional Multi-Objective Problems

Takumi Sonoda and Masaya Nakata, *Member, IEEE*

Abstract—Surrogate-assisted multi-objective evolutionary algorithms have advanced the field of computationally expensive optimization, but their progress is often restricted to low-dimensional problems. This manuscript presents a multiple classifiers-assisted evolutionary algorithm based on decomposition, which is adapted for high-dimensional expensive problems in terms of the following two insights. Compared to approximation-based surrogates, the accuracy of classification-based surrogates is robust for few high-dimensional training samples. Further, multiple local classifiers can hedge the risk of over-fitting issues. Accordingly, the proposed algorithm builds multiple classifiers with support vector machines on a decomposition-based multi-objective algorithm, wherein each local classifier is trained for a corresponding scalarization function. Experimental results statistically confirm that the proposed algorithm is competitive to the state-of-the-art algorithms and computationally efficient as well.

Index Terms—Classification-based surrogate, high-dimensional multi-objective optimization, surrogate-assisted evolutionary algorithm.

I. INTRODUCTION

IN real-world optimization problems, for example, photonic waveguide design [1], often expensive multi-objective optimization problems (EMOPs) are encountered, wherein the evaluation of objective functions is computationally expensive [2]. In recent years, the importance of EMOPs has increased with the growth of the field of automated machine learning. For instance, in the case of multi-objective neural architecture search [3]–[6], the GPU-based evaluation can be computationally and economically expensive. Multi-objective evolutionary algorithms (MOEAs) are a powerful black box optimizer, but are not suitable to solve EMOPs under the restriction on the number of fitness evaluations (FEs) [7].

Surrogate-assisted evolutionary algorithms (SAEAs) [2], [8] are a representative approach to solve EMOPs. A general concept of SAEAs is to perform pre-screening to candidate solutions. Specifically, SAEAs utilize outputs of surrogate models to estimate high-quality solutions to be evaluated; and machine learning (ML) techniques are often employed to build the surrogate models. SAEAs have been proven their effectiveness in significantly reducing the number of expensive FEs both on single- and multi-objective optimization problems

[2]. Hereinafter, multi-objective SAEAs are denoted as SAEAs if not stated differently.

During the last decade, various SAEAs have made great progress in solving complex EMOPs [9], [10]. One of the main directions is to tackle expensive many-objective optimization problems (EMaOPs), wherein more than three objectives are considered. For instance, K-RVEA [11] and CSEA [12] have been adapted for EMaOPs beyond early trials [13]–[15]. Another direction is to reduce the computational time in building surrogate models [11]. However, a critical problem difficulty of the number of decision variables has been considered very recently. A fundamental approach to high-dimensional EMOPs/EMaOPs is to integrate a dimension reduction technique into an SAEA framework, aiming to improve the model accuracy. This approach has been implemented in SA-RVEA-PCA [16] and ADSAPSO [17]. According to [18], a challenge to adapt surrogate models for high-dimensional EMOPs/EMaOPs was first considered in 2020. Specifically, EDN-ARMOEA [18] utilizes a computationally efficient neural network to approximate objective functions. The effectiveness of EDN-ARMOEA has been confirmed on EMOPs/EMaOPs with up to 100 decision variables and 20 objectives. With this recent insight as a start, further explorations are needed to adapt surrogate models for high-dimensional EMOPs/EMaOPs.

For surrogate models, SAEAs can be roughly classified as either an approximation-based SAEA or a classification-based SAEA, which are briefly summarized below (see Section II for a detailed review).

- An approximation-based SAEA builds surrogate models that approximate objective functions and/or defined metric functions. For instance, EDN-ARMOEA, SA-RVEA-PCA, and ADSAPSO belong to this category. Typically, approximation-based SAEAs possess a rich pre-screening capacity to estimate high-quality solutions because an approximation model can rank any candidate solution with real-valued outputs obtained from its model [19]. However, obtaining a sufficient model accuracy may be hindered by a limited number of high-dimensional training samples [16].
- A classification-based SAEA builds surrogate models that predict a plausible class of a candidate solution. For instance, CSEA predicts the dominance relationship between candidate solutions and reference solutions. A classification model may be robust for few high-dimensional training samples [12]. However, its screening capacity

T. Sonoda is with Graduate School of Engineering Science, Yokohama National University, Kanagawa, Japan e-mail: sonoda-takumi-gp@ynu.jp.

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

is relatively poor as it returns less informative outputs compared to approximation models [20].

While the above advantages and disadvantages should depend on MLs and MOEAs employed, there is a possible tradeoff between the approximation/classification-based SAEAs.

Thus, the choice of approximation/classification models can be an important consideration when designing SAEAs suitable for high-dimensional EMOPs/EMaOPs. This insight has been partially supported by the observation in [12]; CSEA outperforms approximation-based SAEAs on middle-scale EMaOPs with up to 30 decision variables. This observation demonstrates the potential of the classification-based SAEAs. However, no classification-based SAEAs have been adapted for high-dimensional EMOPs/EMaOPs, as ADSAPSO, SA-RVEA-PCA, and EDN-ARMOEA are approximation-based SAEAs. Moreover, there is no comparison of approximation/classification-based SAEAs on high-dimensional EMOPs/EMaOPs. Consequently, the potential of the classification-based SAEA on high-dimensional EMOPs/EMaOPs has remained unclear.

This manuscript presents a novel classification-based SAEA adapted for high-dimensional EMOPs/EMaOPs. Our strategy involves utilizing multiple local classifiers as surrogate models, where each local classifier is trained for a corresponding subproblem defined in a decomposition-based MOEA. We employ Support Vector Machine (SVM) [21] and MOEA/D [22] as a surrogate model and base optimizer, respectively. Specifically, our approach, called multiple classifiers-assisted MOEA/D or MCEA/D, is designed with the following three insights. First, SVM is robust for high-dimensional classification problems even with few training samples [23]. Second, the utilization of multiple local surrogates can be a reasonable strategy to hedge the problematic over-fitting issue [19]. Third, we enhance the pre-screening capacity of MCEA/D by ranking candidate solutions with decision scores obtained from SVM-based models. The contributions of this manuscript are summarized below.

- To the best of our knowledge, two novel attempts are considered in MCEA/D: the first approach of the classification-based SAEA adapted for high-dimensional EMOPs/EMaOPs; and the first combination of multiple local surrogate models and the classification-based SAEA for EMOPs/EMaOPs;
- The first comparison of approximation/classification-based SAEAs on high-dimensional EMOPs/EMaOPs is conducted with up to 150 decision variables and 11 objectives. Experimental results reveal the potential of classification-based SAEAs as well as the effectiveness of MCEA/D.

Note that our preliminary work, that is, MOEA/D-S³ [24], provided a preliminary insight for local surrogate modeling on the classification-based SAEA. MOEA/D-S³ trains each local classifier for a corresponding subproblem, but its pre-screening capacity is poor as it does not rank offspring solutions. In addition, it defines the top 50% of solutions in a population as having a “good” category, making it difficult for a surrogate model to specialize for a target sub-problem. Consequently, the effectiveness of MOEA/D-S³ has been restricted on small-

scale problems with up to 17 decision variables and 30000 FEs. MCEA/D inherits the concept of MOEA/D-S³, but it has been adapted for high-dimensional EMOPs/EMaOPs with up to 150 decision variables and 300 FEs.

This manuscript is organized as follows. Section II provides a review of related works, including recent works as an update of the latest survey [10]. Section III explains the frameworks of MOEA/D and SVM. Thereafter, the detailed mechanism of MCEA/D is introduced in Section IV. In Section V, we conduct experiments on benchmark problems and present a comparison of MCEA/D with state-of-the-art SAEAs, for example, EDN-ARMOEA. Section VI provides analytical results to investigate the computational efficiency and the sensitivity of MCEA/D to hyper-parameters. Finally, Section VII summarizes this manuscript with future work.

II. RELATED WORK

This section summarizes related works in terms of the approximation/classification-based models. Table I summarizes representative SAEA variants with the configuration of surrogate models and the experimental setting; M , D , FE_{\max} represent the numbers of objectives, of decision variables, and of FEs, respectively. For instance, “MCEA/D” denotes that it builds classification-based models to predict high-quality solutions considering the scalarization functions; and its performance is validated on problems with $M = \{3, 7, 11\}$, $D = \{50, 100, 150\}$, and $FE_{\max} = 300$. We mainly list recent works for the category “approximation-based SAEAs approximating objective functions” because many existing SAEAs belong to this category and they are summarized in [2], [9], [10], [39]. In addition, we focus on SAEAs that utilize surrogate models to predict the quality of solutions. In addition, other approaches, for instance, dimension reduction [40]–[42], objective extraction [43], [44], transfer learning [45], and constraint handling [46]–[49] have been considered as well.

A. Approximation-based SAEAs

As mentioned in Section I, the approximation-based SAEA utilizes surrogate models that approximate objective functions and/or defined metric functions. Efficient Global Optimization (EGO) [50] has been frequently employed as a search method with surrogate models.

A general approach in this category is to approximate all the objective functions with multiple surrogate models. Certain works demonstrate the effectiveness of the decomposition-based SAEAs, for example, MOEA/D-EGO [26], where the Expected Improvement (EI) criteria are frequently used to select candidate solutions. MOEA/D-EGO builds Kriging models to estimate values of scalarization functions defined on the MOEA/D framework. The Expected Improvement Matrix (EIM) is used in EIM-EGO [28], instead of the EI criteria, aiming to reduce the computational cost. Further, K-RVEA utilizes Kriging models on the RVEA framework [51] to boost the convergence and the diversity of solutions.

Different from the above approach, a defined metric function is also considered as a target function to be approximated. For

TABLE I

CLASSIFICATION OF MULTI-OBJECTIVE SAEAS. “ML”, “APPROX./CLASS.”, AND “TARGET” DENOTE EMPLOYED ML TECHNIQUES, THE APPROXIMATION/CLASSIFICATION-BASED MODELS, AND AN OBJECTIVE TO BE PREDICTED BY SURROGATE MODELS, RESPECTIVELY; M , D , FE_{\max} REPRESENT THE NUMBERS OF OBJECTIVES, OF DECISION VARIABLES, AND OF FITNESS EVALUATIONS, RESPECTIVELY.

Algorithm	Surrogate model			Experimental setting		
	ML	Approx./Class.	Target	M	D	FE_{\max}
SMS-EGO [15]	Kriging	Approx.	Hypervolume	{2, 3, 5}	{3, 6}	130
ParEGO [25]	Kriging	Approx.	Scalarization function	{2, 3}	{2, 3, 6, 8}	250
EDN-ARMOEA [18]	ANN	Approx.	Objective functions	{3, 5, 10, 20}	{20, 40, 60, 100}	11D+119
MOEA/D-EGO [26]	Kriging	Approx.	Objective functions	{2, 3}	{2, 6, 8}	{200, 300}
K-RVEA [11]	Kriging	Approx.	Objective functions	{3, 4, 6, 8, 10}	10	300
AB-MOEA [27]	Kriging	Approx.	Objective functions	{2, 3}	{7, 10, 12, 22}	300
EIM-EGO [28]	Kriging	Approx.	Objective functions	{2, 3, 4, 6}	6	{100, 200}
KTA2 [29]	Kriging	Approx.	Objective functions	{3, 4, 6, 8, 10}	{9, 10, 11}	300
TIC-SMEA [30]	Kriging, RBF	Approx.	Objective functions	{2, 3}	{8, 10, 20, 30}	300
SA-RVEA-PCA [16]	Kriging, PCA	Approx.	Objective functions	2	{100, 130, 160}	{1000, 1200, 4200}
HSMEA [31]	Kriging, RSM, RBF	Approx.	Objective functions	{3, 4, 6, 8, 10}	{9, 10, 11}	300
MOEA/D-RBF [32]	RBF	Approx.	Objective functions	2	{10, 30}	3000
ADSAPSO [17]	RBF	Approx.	Objective functions	{2, 3}	{50, 100, 200}	1000
SAMOE/SVM [33]	SVM	Approx.	Objective functions	2	{8, 10, 12, 30}	{200, 1000, 5000}
HeE-MOEA [34]	SVM, RBF, PCA	Approx.	Objective functions	3	{10, 20, 40, 80}	11D+119
PARETO-SVM [35]	SVM	Class.	Pareto dominance	2	{10, 30}	100000
CSEA [12]	FNN	Class.	Pareto dominance	{2-4, 6, 8-10}	{9, 10, 11, 20, 30}	{300, 600, 900}
CPS-MOEA [36]	KNN	Class.	Pareto dominance	{2, 3}	30	{2000, 40000, 100000}
MOEA/D-CPS [37]	KNN	Class.	Pareto dominance	{2, 3}	30	{150000, 297500}
MOEA/D-SVM [38]	SVM	Class.	Scalarization function	{2, 3}	{10, 30}	{100000, 300000}
MOEA/D-S ³ [24]	SVM	Class.	Scalarization function	{4, 6, 8}	{13, 15, 17}	30000
MCEA/D	SVM	Class.	Scalarization function	{3, 7, 11}	{50, 100, 150}	300

instance, ParEGO [25] builds a Kriging model to approximate a scalarization function converted from a set of objective functions. It controls a weight vector of the scalarization function dependent on the current search status to enhance the diversity of solutions with the single surrogate model. In SMS-EGO [15], a Kriging model is used to approximate a hypervolume of when candidate solutions are added to a population.

An ensemble framework of multiple surrogate models can be an effective strategy to hedge the over-fitting issue. MOEA/D-RBF [32] estimates the quality of solutions from three Radial Basis Function (RBF) networks with different kernels. Whereas, SAMOE/SVM [33] is based on an ensemble approach of SVM-based surrogate models, and it adaptively adjust those models with their hyper-parameter settings. Further, HeE-MOEA [34] and HSMEA [31] utilize multiple heterogeneous surrogates with different ML techniques. For instance, in HSMEA, Kriging, Polynomial Response Surface Method (RSM), and RBF are cooperated to effectively approximate objective functions. In recent years, certain modern works have adapted infill criteria to select promising candidate solutions for exact evaluations. AB-MOEA [27] tunes hyper-parameters of Angle-Penalized Distance [51] (APD) and acquisition functions, for example, Lower Confidence Bounds (LCB) [52]. Whereas, KTA2 [29] introduces an adaptive infill criterion to obtain a plausible strategy to sample solutions to be evaluated. Similar to KTA2, TIC-SMEA [30] collaboratively uses two infill criteria regard to convergence and diversity.

For high-dimensional problems, SA-RVEA-PCA [16] builds a Gaussian process model with Principal Component Analysis (PCA) to improve the model accuracy for each objective function. SA-RVEA-PCA has proven its effectiveness on problems with up to 160 decision variables. Similar to SA-RVEA-PCA, ADSAPSO [17] extracts important decision variables in solving problems, performing as a dimension reduction tech-

nique. Further, ADSAPSO derives competitive performances to the state-of-the-art approximation-based SAEAs, that is, MOEA/D-EGO and K-RVEA, on problems with up to 200 decision variables. EDN-ARMOEA [18] employs a dropout artificial neural network (ANN) model [53] to prevent the over-fitting issue, and it significantly reduces the computational time on EMOPs/EMaOPs with up to 100 decision variables.

B. Classification-based SAEAs

In multi-objective optimization, objective values may be less important to be approximated if the dominance relationship is known [8]. Thus, the dominance relationship is often set to a target to be predicted by the surrogate models.

CPS-MOEA [36] utilizes a K-Nearest Neighbor (KNN) model to filter candidate solutions. The KNN model builds “good” clusters of non-dominated solutions, and offspring solutions belonging to those clusters are evaluated. MOEA/D-CPS [37] is based on a similar idea to CPS-MOEA, but it has been extended to the MOEA/D framework. Further, CSEA [12] utilizes a feed-forward neural network (FNN) model to predict the dominance relationship between candidate solutions and reference solutions. PARETO-SVM [35] predicts the dominance relationship through an abnormality determination scheme. Specifically, it builds a one-class SVM as a characteristic function that estimates a degree of non-dominated solution; and then it optimizes the model outputs to obtain the candidate solutions likely to being non-dominated. In contrast, MOEA/D-SVM [38] defines classes according to the value of the scalarization functions.

In decomposition-based SAEAs, a scalarization function is also set to a target to be predicted by the surrogate models. In MOEA/D-SVM, a positive class set is composed of N current best solutions of scalarization functions while a negative class set is composed of N sub-best solutions. Subsequently, MOEA/D-SVM builds an SVM model that pre-

dicts whether candidate solutions exist in a positive region of the best solutions. Thus, MOEA/D-SVM intends to optimize all the scalarization functions with a global surrogate model. However, to the best of our knowledge, no work except for our preliminary work (MOEA/D-S³) [24] intends to assign a classification-based model to each sub-problem.

III. PRELIMINARIES

Although many variants of MOEA/D have been proposed [54], [55], for example, MOEA/D-DRA [56], MOEA/D-ACO [57], and MOEA/D-MO [58], we use MOEA/D-DE [59] as a base optimizer of MCEA/D. In addition, we use a nonlinear SVM, simply denoted as SVM in this manuscript. Accordingly, this section describes the frameworks of MOEA/D-DE and SVM.

A. MOEA/D-DE

MOEA/D-DE evolves a set of N solutions $\mathcal{P} = \{\mathbf{x}^i\}_{i=1}^N$ to approximate a Pareto optimal front on a multi-criteria optimization problem [60], expressed as;

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), \dots, f_M(\mathbf{x})\} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{S}, \end{aligned} \quad (1)$$

where $\mathbf{x} \in \mathcal{S}$ denotes a decision vector \mathbf{x} which belongs to a feasible region \mathcal{S} and $\mathbf{F}(\mathbf{x})$ is a set of M objective functions f_j and $M \geq 2$. This manuscript considers real-valued optimization problems with $\mathbf{x} \in \mathbb{R}^D$, where D is the number of decision variables. A fundamental idea of MOEA/D-DE as well as MOEA/D is to divide a particular problem into N sub-problems using scalarization functions; and then a differential evolution algorithm is executed to produce an offspring solution \mathbf{y}^i for i -th sub-problem. Specifically, we use the Tchebycheff function as a scalarization function, where the i -th scalarization function is expressed as;

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

where $\boldsymbol{\lambda}^i = \{\lambda_1^i, \dots, \lambda_M^i\}$ is a weight vector such that $\lambda_j^i \geq 0$ and $\sum_{j=1}^M \lambda_j^i = 1$ for all $i = 1, \dots, N$, and $\mathbf{z} = \{z_1, \dots, z_M\}$ is a set of reference points determined as the optimal value of each objective function.

After the definition of N sub-problems, MOEA/D-DE defines an index set of neighbor sub-problems for the i -th one, denoted by $\mathbf{B}(i)$. In particular, $\mathbf{B}(i)$ contains T indices i_1, \dots, i_T , where $\boldsymbol{\lambda}^{i_1}, \dots, \boldsymbol{\lambda}^{i_T}$ are the T closest weight vectors to $\boldsymbol{\lambda}^i$. Subsequently, the N initial solutions are generated and then evaluated. Next, to generate an offspring solution \mathbf{y}^i for i -th sub-problem, MOEA/D-DE builds an index set of parent candidates \mathbf{P} , expressed as;

$$\mathbf{P} := \begin{cases} \mathbf{B}(i) & \text{with probability } \delta, \\ \{1, \dots, N\} & \text{otherwise,} \end{cases} \quad (3)$$

Thus, parent candidates are set to solutions assigned to neighbor sub-problems indexed by $\mathbf{B}(i)$ with a probability δ ; otherwise, all the N solutions are considered as parent candidates. Two indices r_1 and r_2 are randomly selected from \mathbf{P} ; and then two parents \mathbf{x}^{r_1} and \mathbf{x}^{r_2} are determined.

Algorithm 1 MOEA/D-DE

```

1: Input:  $N, T, \{\boldsymbol{\lambda}^1, \dots, \boldsymbol{\lambda}^N\}, F, CR, \eta, p_m, n_r, \delta$ 
2: Output:  $EP$ 
3: Initialize  $EP$  as  $EP \leftarrow \emptyset$ ;
4: Set  $\mathbf{B}(i)$  to indices of the  $T$  closest weight vectors to  $\boldsymbol{\lambda}^i \forall i \in \{1, \dots, N\}$ ;
5: Set  $\mathcal{P}$  to initial solutions  $\{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ ;
6: Evaluate  $\forall \mathbf{x} \in \mathcal{P}$ ;
7: Initialize  $z_j$  as  $z_j \leftarrow \min_{\mathbf{x} \in \mathcal{P}} f_j(\mathbf{x}) \forall j \in \{1, \dots, M\}$ ;
8: while termination criteria are not met do
9:   for  $i = 1$  to  $N$  do
10:    Set  $\mathbf{P}$  with (3);
11:    Generate  $\mathbf{y}^i$  with  $\mathbf{P}$ ;
12:    Evaluate  $\mathbf{y}^i$ ;
13:    Update  $z_j \leftarrow \min\{z_j, f_j(\mathbf{y}^i)\} \forall j \in \{1, \dots, M\}$ ;
14:    Randomly shuffle indices of  $\mathbf{P}$ ;
15:     $count \leftarrow 0$ ;
16:    for each  $j \in \mathbf{P}$  do
17:      if  $g(\mathbf{y}^i | \boldsymbol{\lambda}^j, \mathbf{z}) \leq g(\mathbf{x}^j | \boldsymbol{\lambda}^j, \mathbf{z})$  and  $count < n_r$  then
18:         $\mathbf{x}^j \leftarrow \mathbf{y}^i$ ;
19:         $count \leftarrow count + 1$ ;
20:      end if
21:    end for
22:    Remove from  $EP$  all the solutions dominated by  $\mathbf{y}^i$ ;
23:    Add  $\mathbf{y}^i$  to  $EP$  if  $\mathbf{y}^i$  is the Pareto solution in  $EP$ ;
24:  end for
25: end while

```

Subsequently, the offspring solution \mathbf{y}^i is generated through crossover and mutation operators. As a crossover operator, a k -th decision variable of \mathbf{y}^i , that is, y_k^i , is determined with \mathbf{x}^i , \mathbf{x}^{r_1} , and \mathbf{x}^{r_2} as;

$$y_k^i = \begin{cases} x_k^i + F \times (x_k^{r_1} - x_k^{r_2}) & \text{with probability } CR, \\ x_k^i & \text{otherwise,} \end{cases} \quad (4)$$

where F and CR are scaling factor and crossover rate, respectively. Next, y_k^i is further updated with the polynomial mutation as;

$$y_k^i \leftarrow \begin{cases} y_k^i + \sigma_k \times (b_k - a_k) & \text{with probability } p_m, \\ y_k^i & \text{otherwise,} \end{cases} \quad (5)$$

where p_m is a mutation probability; a_k and b_k are the lower/upper bounds of the k -th decision variable, that is, $y_k^i \in [a_k, b_k]$. σ_k is further expressed as;

$$\sigma_k = \begin{cases} (2r)^{1/(\eta+1)} - 1 & \text{if } r < 0.5, \\ 1 - (2 - 2r)^{1/(\eta+1)} & \text{otherwise,} \end{cases} \quad (6)$$

where η is a distribution index of polynomial mutation and $r \in [0, 1]$ is a uniformly-sampled random value. After completing the calculation of y_k^i for all $k = 1, \dots, D$, \mathbf{y}^i is evaluated with the objective functions. Next, existing solutions $\mathbf{x}^k \forall k \in \mathbf{P}$ may be replaced with \mathbf{y}^i if $g(\mathbf{y}^i | \boldsymbol{\lambda}^k, \mathbf{z}) \leq g(\mathbf{x}^k | \boldsymbol{\lambda}^k, \mathbf{z})$. In addition, the number of replacements of \mathbf{y}^i is restricted to n_r ; if there are more than n_r solutions worse than \mathbf{y}^i , n_r ones are randomly selected to be replaced with \mathbf{y}^i . These processes are repeated for all the N sub-problems. Algorithm 1 describes the procedures of MOEA/D-DE. Note that an external archive EP is used to store Pareto solutions obtained.

B. SVM

For a particular binary classification problem, where an input $\mathbf{x} \in \mathbb{R}^D$ is classified as a defined class $c \in \{+1, -1\}$,

SVM builds a classifier as a decision function $c(\mathbf{x})$, expressed as;

$$c(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \phi(\mathbf{x}) + w_0), \quad (7)$$

where $\phi(\mathbf{x})$ is a mapping function from the input space \mathcal{X} to a high-dimensional feature space \mathcal{F} , and the weight vector $\mathbf{w} \in \mathbb{R}^D$ and bias $w_0 \in \mathbb{R}$ are the parameters to be optimized. Note that a sign function $\text{sgn}(x)$ for $x \in \mathbb{R}$ returns $+1$, 0 and -1 for $x > 1$, $x = 0$, and $x < 1$, respectively. For a particular training dataset \mathcal{D} consisting of $|\mathcal{D}|$ samples (\mathbf{x}_i, c_i) , SVM solves the following optimization problem;

$$\begin{aligned} \min_{\mathbf{w}, w_0, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{|\mathcal{D}|} \xi_i \\ \text{s.t.} \quad & c_i(\mathbf{w}^T \phi(\mathbf{x}_i) + w_0) \geq 1 - \xi_i, \xi_i \geq 0, \end{aligned} \quad (8)$$

where ξ_i is a slack variable; and a hyper-parameter C controls the balance between the margin and empirical loss. Here, the problem of (8) can be further simplified using the Lagrangian and the KKT (Karush-Kuhn-Tucker) condition [61], expressed as;

$$\begin{aligned} \max \quad & \sum_{i=1}^{|\mathcal{D}|} a_i - \frac{1}{2} \sum_{i,j=1}^{|\mathcal{D}|} a_i a_j c_i c_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^{|\mathcal{D}|} a_i c_i = 0, 0 \leq a_i \leq C, \end{aligned} \quad (9)$$

where $\{a_i\}_{i=1}^{|\mathcal{D}|}$ is a set of the Lagrange multiplier. To avoid an expensive calculation of the inner products in (9), $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ is replaced with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. Then, (9) with $K(\mathbf{x}_i, \mathbf{x}_j)$ can be optimized for $\{a_i\}_{i=1}^{|\mathcal{D}|}$ by for example, the steepest descent method. Let $\{a_i^*\}_{i=1}^{|\mathcal{D}|}$ be a set of solution of the problem (9). Then, (7), that is, the decision function, can be written as;

$$c(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^{|\mathcal{D}|} a_i^* c_i K(\mathbf{x}_i, \mathbf{x}) + w_0^* \right), \quad (10)$$

where $w_0^* = c_i + \mathbf{x}_i^T \mathbf{w}^*$ and $\mathbf{w}^* = \sum_{i=1}^{|\mathcal{D}|} a_i^* c_i \mathbf{x}_i$. In this study, we use the Radial Basis Function (RBF) kernel, expressed as;

$$K(\mathbf{x}_i, \mathbf{x}) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}\|) \quad (11)$$

where $\gamma > 0$ controls the complexity of the decision boundary.

IV. MCEA/D

This section begins by introducing a detailed characteristic of MCEA/D, and subsequently its algorithm is described.

A. Characteristic

MCEA/D is motivated to obtain reliable models by building local SVM-based models and to boost its pre-screening capacity while saving the computational time. Detailed characteristics are summarized below.

- *Hedge of the over-fitting issue.* The following two strategies are employed in MCEA/D to improve the model accuracy on the high-dimensional EMOPs/EMaOPs. First,

a surrogate model is assigned to each sub-problem; and each model is used to generate a solution \mathbf{y}^i for the i -th sub-problem to prevent a propagation of the negative impact of an over-fitted model to the whole working of MCEA/D. Second, in building a dataset for the i -th sub-problem, the current best solutions of neighbor sub-problems are defined as having a positive class (i.e., a “good” category). This is to design “easy-to-learn” correlations between samples having the positive class under an assumption of MOEA/D that optimal solutions of neighbor sub-problems exist in similar region.

- *Boosting computational efficiency and pre-screening capacity.* A straightforward approach to utilize the SVM classifiers is to repeatedly generate a candidate solution till it is predicted as having the positive class. However, this loop increases the computational time. Thus, we define the maximum repeat time to save the computational time. Further, as a backup strategy, we select the candidate solution closest to a decision boundary by ranking candidates with *decision values* obtained from the SVM model if there is no candidate solution predicted as the positive class. This backup strategy contributes to improving the pre-screening capacity of MCEA/D, as it can rank solutions such as the approximation-based SAEAs.

B. Algorithm

To begin with, we use the following mathematical notations;

- \mathcal{A} an archive set consisting of all evaluated solutions $\mathbf{x} \in \mathbb{R}^D$ with their objective values $\{f_1(\mathbf{x}), \dots, f_M(\mathbf{x})\}$;
- \mathcal{D}_i a dataset consisting of training samples (\mathbf{x}_j^i, c_j^i) designed for i -th sub-problem;
- \mathcal{C}_i a set of current best solutions of sub-problems for the i -th sub-problem;
- \mathcal{Y}_i a set of candidate solutions of offspring \mathbf{y}^i for i -th sub-problem;
- $c_i^*(\mathbf{x})$ a decision function for i -th sub-problem, referred to as an SVM classifier, where Lagrange multipliers $\{a_j^{i*}\}_{j=1}^{|\mathcal{D}_i|}$ have been optimized for \mathcal{D}_i ;
- $d_i^*(\mathbf{x})$ a decision score function determined with $\{a_j^{i*}\}_{j=1}^{|\mathcal{D}_i|}$;
- R_{\max} the maximum repeat time to control the number of candidate solutions, that is, $|\mathcal{Y}_i|$.

Other notations used in this section are inherited from the previous section.

MCEA/D performs in an almost similar manner as in MOEA/D-DE except for the solution generation process. Algorithm 2 describes the whole procedure of MCEA/D, where new procedures are highlighted with underline. At line 8, N initial solutions with their objective values are inserted to the archive set \mathcal{A} . For each sub-problem, MCEA/D builds a decision function $c_i^*(\mathbf{x})$ and then it generates an offspring solution \mathbf{y}^i by utilizing $c_i^*(\mathbf{x})$, which are denoted by *Model-construction* at line 11 and *Solution-generation* at line 13, respectively. Subsequently, \mathbf{y}^i is inserted to \mathcal{A} . Three additional hyper-parameters C , γ , and R_{\max} are required in MCEA/D.

Algorithm 2 MCEA/D

```

1: Input:  $N, T, \{\lambda^1, \dots, \lambda^N\}, F, CR, \eta, p_m, n_r, \delta, \gamma, C, R_{\max}$ 
2: Output:  $EP$ 
3: Initialize  $EP$  as  $EP \leftarrow \emptyset$ ;
4: Set  $B(i)$  to indices of the  $T$  closest weight vectors to  $\lambda^i \forall i \in \{1, \dots, N\}$ ;
5: Set  $\mathcal{P}$  to initial solutions  $\{x^1, \dots, x^N\}$ ;
6: Evaluate  $\forall x \in \mathcal{P}$ ;
7: Initialize  $z_j$  as  $z_j \leftarrow \min_{x \in \mathcal{P}} f_j(x) \forall j \in \{1, \dots, M\}$ ;
8: Initialize  $\mathcal{A}$  as  $\mathcal{A} \leftarrow \mathcal{P}$ ;
9: while termination criteria are not met do
10:   for  $i = 1$  to  $N$  do
11:     Build surrogate as  $c_i^*(x) \leftarrow \text{Model-Construction}(\mathcal{A}, \lambda^i)$ 
12:     Set  $\mathcal{P}$  with (3);
13:     Generate  $y^i$  as  $y^i \leftarrow \text{Solution-generation}(\mathcal{P}, \mathcal{P}, c_i^*(x))$ ;
14:     Evaluate  $y^i$ ;
15:     Update  $z_j \leftarrow \min\{z_j, f_j(y^i)\} \forall j \in \{1, \dots, M\}$ ;
16:     Randomly shuffle indices of  $\mathcal{P}$ ;
17:      $count \leftarrow 0$ ;
18:     for each  $j \in \mathcal{P}$  do
19:       if  $g(y^i | \lambda^j, z) \leq g(x^j | \lambda^j, z)$  and  $count < n_r$  then
20:          $x^j \leftarrow y^i$ ;
21:          $count \leftarrow count + 1$ ;
22:       end if
23:     end for
24:     Update  $\mathcal{A}$  as  $\mathcal{A} \cup \{y^i\}$ ;
25:     Remove from  $EP$  all the solutions dominated by  $y^i$ ;
26:     Add  $y^i$  to  $EP$  if  $y^i$  is the Pareto solution in  $EP$ ;
27:   end for
28: end while

```

Strictly formalizing the computational complexity of MCEA/D is difficult due to the complexity of the SVM learning mechanism. Specifically, the computational complexity for building an SVM model, denoted by O_{SVM} , can be at least $O(|\mathcal{D}|^2)$ when C is small and $O(|\mathcal{D}|^3)$ when C is large [62], [63]. Thus, it is difficult to identify a strict value of O_{SVM} . However, because we set $C = 1$ as a small value (see Section V), we can expect that O_{SVM} is sufficiently smaller than $O(|\mathcal{D}|^3)$. Then, MCEA/D builds N SVM models and thus, $O(NO_{SVM})$ is required to build surrogate models. Further, the computational complexity for building a Gaussian process (GP) model is $O(|\mathcal{D}|^3)$. Then, if N is usually smaller than $|\mathcal{D}|$ and $O_{SVM} \ll O(|\mathcal{D}|^3)$, MCEA/D should perform faster than GP-based SAEAs, for example, MOEA/D-EGO. The computational time of MCEA/D is evaluated in Section VI.

The rest of this section explains the detailed procedure of *Model-construction* and *Solution-generation*.

1) *Model construction*: For i -th sub-problem, MCEA/D builds a dataset \mathcal{D}_i . All the solutions contained in \mathcal{A} are used for training inputs, and thus $\mathcal{D}_i = \{(x_j^i, c_j^i)\}_{j=1}^{|\mathcal{A}|}$; and the class c_j^i of x_j^i is determined as;

$$c_j^i = \begin{cases} +1 & \text{if } x_j^i \in \mathcal{C}_i \\ -1 & \text{otherwise,} \end{cases} \quad (12)$$

Algorithm 3 Model construction

```

1:  $\mathcal{D}_i, \mathcal{C}_i \leftarrow \emptyset, \emptyset$ 
2: for each  $k \in B(i)$  do
3:    $x_k^* \leftarrow \arg \min_{x \in \mathcal{A} \wedge x \notin \mathcal{C}_i} g(x | \lambda^k, z)$ 
4:    $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{x_k^*\}$ 
5: end for
6: for each  $x \in \mathcal{A}$  do
7:   if  $x \in \mathcal{C}_i$  then
8:      $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{(x, +1)\}$ 
9:   else
10:     $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{(x, -1)\}$ 
11:   end if
12: end for
13:  $c_i^*(x) \leftarrow$  build the decision function trained with  $\mathcal{D}_i$ 
14: return  $c_i^*(x)$ 

```

where \mathcal{C}_i is a set of current best solutions of neighbor sub-problems indexed by $B(i)$, that is;

$$\mathcal{C}_i = \left\{ x \in \mathcal{A} \mid \arg \min_{x \in \mathcal{A}} g(x | \lambda^k, z), \forall k \in B(i) \right\}, \quad (13)$$

Thus, T current best solutions x_k^{i*} are contained because $|B(i)| = T$, that is, $\mathcal{C}_i = \{x_1^{i*}, \dots, x_T^{i*}\}$. Accordingly, \mathcal{D}_i is designed such that $c_i^*(x)$ captures a good region, which may be improving $g(x | \lambda^i, z)$ together with its neighbor scalarization functions. In addition, each $c_i^*(x)$ may be similar to but still different from ones built for neighbor sub-problems. Accordingly, each sub-problem is conservatively explored with different surrogate models.

As an exceptional case, few of the neighbor sub-problems may share the same solution. This results in decreasing the number of positive samples (having $c = +1$), making it difficult for SVM to improve the generalization capacity. In this case, we add the sub-best solution to \mathcal{C}_i to avoid duplicative selection. Technically, we insert a selected solution x_k^{i*} to \mathcal{C}_i sequentially with the order of $k = 1, \dots, T$, where $x_k^{i*} \notin \mathcal{C}_i$ is selected from \mathcal{A} for $k > 1$. With this exception handling, it is guaranteed that \mathcal{D}_i always includes T positive samples. Correspondingly, a ratio of the positive class involved in \mathcal{D}_i is restricted to a small value $T/|\mathcal{A}|$; the increase of counter-examples, that is, negative samples (having $c = -1$) is helpful to build a surrogate model specialized for a targeted sub-problem.

Finally, given hyper-parameters C and γ , MCEA/D builds a decision function $c_i^*(x)$ with $\{a_j^{i*}\}_{j=1}^{|\mathcal{D}_i|}$ optimized for \mathcal{D}_i , where the problem of (9) is solved by the Sequential Minimal Optimization algorithm [64]. Algorithm 3 describes the complete procedure of the model construction, where the exception handling aforementioned is implemented at line 2-5.

2) *Solution generation*: Given $c_i^*(x)$, MCEA/D repeats to generate a candidate solution \hat{y}^i , where the maximum repeat time is bounded by R_{\max} . First, the set of candidate solutions \mathcal{Y}_i is set to an empty set \emptyset . Thereafter, MCEA/D generates \hat{y}^i via the same procedure as in the MOEA/D-DE framework. If $c_i^*(\hat{y}^i) = +1$, y^i is immediately set to \hat{y}^i , and the solution generation process is terminated; otherwise, \hat{y}^i is inserted to \mathcal{Y}_i and \hat{y}^i is re-produced.

If there is no candidate solution predicted as having the positive class, that is, $|\mathcal{Y}_i| = R_{\max}$, y^i is set to the candidate

Algorithm 4 *Solution generation*

```

1:  $\mathcal{Y}_i \leftarrow \emptyset$ 
2: for  $r = 0$  to  $R_{\max}$  do
3:    $\hat{\mathbf{y}}^i \leftarrow$  generate a candidate as in MOEA/D-DE
4:   if  $c_i^*(\hat{\mathbf{y}}^i) = +1$  then
5:      $\mathbf{y}^i \leftarrow \hat{\mathbf{y}}^i$ 
6:     return  $\mathbf{y}^i$ 
7:   else
8:      $\mathcal{Y}_i \leftarrow \mathcal{Y}_i \cup \{\hat{\mathbf{y}}^i\}$ 
9:   end if
10: end for
11:  $\mathbf{y}^i \leftarrow \arg \max_{\hat{\mathbf{y}}^i \in \mathcal{Y}_i} d_i^*(\hat{\mathbf{y}}^i)$ 
12: return  $\mathbf{y}^i$ 

```

solution closest to a decision boundary drawn by $c_i^*(\mathbf{x})$. Specifically, the argument value of $\text{sgn}(\cdot)$ of (10), denoted as a decision score function $d_i^*(\mathbf{x})$, can be a metric to quantify the distance between \mathbf{x} and the decision boundary. From (10), with $\{a_j^{i*}\}_{j=1}^{|\mathcal{D}_i|}$ of $c_i^*(\mathbf{x})$ and \mathcal{D}_i , $d_i^*(\mathbf{x})$ can be defined as;

$$d_i^*(\mathbf{x}) = \sum_{j=1}^{|\mathcal{D}_i|} a_j^{i*} c_j^i K(\mathbf{x}_j^i, \mathbf{x}) + w_0^*, \quad (14)$$

where $d_i^*(\mathbf{x}) > 0$ for $c = +1$ and $d_i^*(\mathbf{x}) < 0$ for $c = -1$. Under the condition of $d_i^*(\hat{\mathbf{y}}^i) < 0 \forall \hat{\mathbf{y}}^i \in \mathcal{Y}_i$, we can identify $\hat{\mathbf{y}}^i$ having the maximum value of $d_i^*(\hat{\mathbf{y}}^i)$ as one closest to the decision boundary, that is, a region of the positive class. Thus, \mathbf{y}^i can be determined as;

$$\mathbf{y}^i = \arg \max_{\hat{\mathbf{y}}^i \in \mathcal{Y}_i} d_i^*(\hat{\mathbf{y}}^i) \quad \text{if } |\mathcal{Y}_i| = R_{\max}. \quad (15)$$

The condition $|\mathcal{Y}_i| = R_{\max}$ is equal to $d_i^*(\hat{\mathbf{y}}^i) < 0 \forall \hat{\mathbf{y}}^i \in \mathcal{Y}_i$. This procedure plays a role to rank the candidate solutions. Note that MCEA/D does not rank candidate solutions having the positive class to save the computational time. Algorithm 4 summarizes the above procedures.

V. EXPERIMENT

This section evaluates the performance of MCEA/D on well-known benchmark problems, i.e., the DTLZ problems [65] and the WFG problems [66]. All experiments are conducted on the Evolutionary multi-objective optimization platform (PlatEMO) [67] with 3.0 GHz CPU and 16GB RAM.

A. Experimental setting

1) *Problem settings*: We employ seven DTLZ (DTLZ1–7) and nine WFG (WFG1–9) problems, where the numbers of objective functions and of decision variables are set to $M = \{3, 7, 11\}$ and $D = \{50, 100, 150\}$, respectively. Further, for the WFG problems, the numbers of position variables and of distance variables are set to $k = M - 1$ and $l = D - k$, respectively [66].

2) *Comparison algorithms*: The following six algorithms are compared: for approximation-based SAEAs, K-RVEA, MOEA/D-EGO, and EDN-ARMOEA; for classification-based SAEAs, CSEA, CPS-MOEA, and MCEA/D. Hyper-parameter settings for each algorithm are summarized below.

- for K-RVEA, the penalty control parameter $\alpha = 2$, the frequency of reference vector adaptation $f_r = 0.1$, $|u| = 5$, $\delta = 0.05N$, and $w_{\max} = 5$ for parameters to manage Kriging models;
- for MOEA/D-EGO, the number of FEs at each generation $K_E = 5$, $L_1 = 80$, $L_2 = 20$ for parameters to manage Kriging models, and $T = \lceil 0.1N \rceil$, $\delta = 0.9$, and $n_r = 2$ for parameters of MOEA/D-DE [26];
- for EDN-ARMOEA, a diversity threshold parameter $\delta = 0.08$, the maximum number of generations for AR-MOEA $iter = 20$, the number of FEs at each generation $k = 5$; for ANN parameters, $J = K = 40$, $wd = 10^{-5}$, $lr = 0.01$, $iter_{\text{train}} = 80000$, $iter_{\text{test}} = 100$, $iter_r = 8000$, $p_I = 0.8$, and $p_R = 0.5$ [18];
- for CSEA, the number of hidden neurons of FNN $H = 10$, the number of reference solutions $K = 6$ [12], and the number of iterations for each learning is 800 [31];
- for CPS-MOEA, the number of offspring is 5 [36];
- and for MCEA/D, $R_{\max} = 10$, $\gamma = 1.0$, $C = 1.0$, and the parameter settings of MOEA/D-DE are the same as in MOEA/D-EGO.

K-RVEA, EDN-ARMOEA, and CSEA use the simulated binary crossover, wherein the distribution index of crossover d_c and the crossover probability p_c are set to 20 and 1.0, respectively. Whereas, MOEA/D-EGO, CPS-MOEA, and MCEA/D use the differential evolution as a crossover operator with $F = 0.5$ and $CR = 1.0$. All the methods use the polynomial mutation with $\eta = 20$ and $p_m = 1/D$.

For the number of initial training samples, we slightly modified a popular setting used in modern works. Specifically, modern SAEAs, for example, K-RVEA, MOEA/D-EGO, EDN-ARMOEA, and CSEA, are designed to generate $11D - 1$ initial training samples. This strategy can be employed to improve the initial accuracy of surrogate models if a sufficient number of evaluated solutions is available in advance. Otherwise, it consumes a non-negligible number of FEs on high-dimensional EMOPs/EMaOPs. Moreover, as it is hard to argue a universal setup guide for the number of initial training inputs, SAEAs robustness for the initial model accuracy can be useful in practice. Thus, in this manuscript, all the six algorithms are designed to generate N initial training samples corresponding to initial solutions. Further, we increase N to 100 although $N = 50$ is frequently used in modern works, for example, [12], [18]. These settings with regard to the initial training samples and N are also employed in [29]. Specifically, for EDN-ARMOEA, CSEA, and CPS-MOEA, $N = 100$ is employed. Moreover, for decomposition-based SAEAs, that is, K-RVEA, MOEA/D-EGO, and MCEA/D, we determine N close to 100 with the two-layered approach [68] to obtain uniformly-distributed weight vectors. Specifically, we set $N = \{91, 91, 77\}$ for $M = \{3, 7, 11\}$ with $(H_1, H_2) = \{(12, 0), (3, 1), (2, 1)\}$. For all the algorithms, initial solutions are generated with the Latin hypercube sampling method [69]. Note that in Section VI-B we reconduct experiments with $N = 50$ to analyze the sensitivity of MCEA/D.

3) *Evaluation scheme*: The Inverted Generational Distance (IGD) [70] is employed as a performance metric. The maximum number of FEs is set to 300 including the N FEs

consumed at the initialization process. Specifically, all the algorithms store all the evaluated solutions in an external archive during the run; each algorithm is forcefully terminated if the number of FEs reaches 300. For fair comparison, the IGD value is calculated from a set of Pareto solutions determined from all the 300 evaluated solutions in the external archive.

The IGD values are reported as an average value of 21 trials with different random seeds. In accordance with a manner of statistical comparison conducted in modern works, the Wilcoxon rank-sum test is applied to each pair of MCEA/D and a compared algorithm. This statistical test confirms whether MCEA/D is significantly better, worse, or competitive for the compared algorithms for each problem. Moreover, we further apply a multiple test to statistically confirm the effectiveness of the algorithms for overall results. The Friedman test was applied to the overall IGD values. If the significant probability was sufficiently small, we applied the Holm test with the Wilcoxon signed-rank test as a post-hoc method to calculate the significant probability for each pair of the algorithms. We consider that there is a significant difference if the significant probability is less than 0.05.

B. Results

Table II summarizes averaged IGD values on the DTLZ problems with $M = \{3, 7, 11\}$ and $D = \{50, 100, 150\}$. The best IGD value is highlighted in color. In this table, statistical results of the Wilcoxon rank-sum test are summarized with symbols wherein “+”, “−”, and “ \approx ” denote that the IGD value of an alternative algorithm is significantly better, worse, and competitive compared to that obtained by MCEA/D, respectively. Note that MOEA/D-EGO failed to solve DTLZ4 with $D = \{100, 150\}$ because Kriging models fail to handle high-dimensional inputs. This observation is also reported in [17]. Fig. 1 shows the average ranks obtained from IGD values in Table II and the statistical results of the multiple test on the DTLZ problems. In the figure, each line indicates a pair of two algorithms confirmed as having the significant difference via the multiple test. Further, for $D = \{100, 150\}$, the average ranks are calculated from the IGD values except for ones on DTLZ4.

As shown in Table II, MCEA/D derives the best IGD value for all the experimental cases except for DTLZ7; MCEA/D is ranked first (see Fig. 1). From results of the Wilcoxon rank-sum test, the IGD values of MCEA/D are significantly better than all the alternative algorithms on more than 13 experimental cases, as summarized in Table II. When we consider the overall results for each D , MCEA/D can be identified as the best algorithm that is significantly better than all the five alternative algorithms.

Table III and Fig. 2 report averaged IGD values and statistical results of the multiple test on the WFG problems, respectively. For $D = 50$, while EDN-ARMOEA and MCEA/D are respectively ranked first and second, there is no clear difference in terms of “+/-/ \approx ” and the multiple test result. When D is further increased to 100 and 150, MCEA/D is ranked first; the number of “−” slightly increases for

EDN-ARMOEA, and it clearly increases for the other four algorithms. Although the effectiveness of MCEA/D is not fully supported by the multiple test results, the performance of MCEA/D tends to improve with the increase of the number of decision variables D .

In addition, the classification-based SAEAs tend to derive better performances than the approximation-based SAEAs except for EDN-ARMOEA on high-dimensional problems. Thus, classification-based surrogate models may be a reasonable choice for improving the scalability of SAEAs against the number of decision variables. Apart from this tendency, EDN-ARMOEA successfully boosts the performance. Hence, as motivated in EDN-ARMOEA, building computationally-efficient approximation models is another reasonable strategy.

VI. ANALYSIS

This section further provides analytical results to investigate the computational efficiency and the sensitivity of MCEA/D for the hyper-parameters N and R_{\max} .

A. Computational time

We compare computational time of all the six algorithms, which are obtained from the experimental results presented in Tables II and III. Table IV presents the averaged computational time [sec] consumed to complete one trial for $D = \{50, 100, 150\}$. For $D = \{100, 150\}$, the averaged computational time is calculated from all experimental results shown in Tables II and III except for ones on DTLZ4. From the table, it is evident that CPS-MOEA performs much faster, as it builds a computationally efficient KNN model. MCEA/D is ranked second, which consumes up to 4.882 seconds at least one order of magnitude smaller than the other algorithms except for CPS-MOEA. As a general trend, the approximation-based SAEAs tend to consume the computational time more than the classification-based SAEAs. The computational time of CPS-MOEA and MCEA/D is less dependent on the increase of D since the computational complexity of their surrogate models mainly depend on the number of training samples. Although MCEA/D builds N surrogate models, it performs sufficiently faster.

B. Impact of Population size

As discussed in Section V, robustness of the SAEA performance to the initial model accuracy can be an important consideration when designing practical SAEAs. Accordingly, we conducted additional experiments with $N = 50$, aiming to analyze the sensitivity of MCEA/D to the initial model accuracy. For EDN-ARMOEA, CSEA, and CPS-MOEA, $N = 50$ is employed. Whereas, for K-RVEA, MOEA/D-EGO, and MCEA/D, we set $N = \{45, 56, 66\}$ for $M = \{3, 7, 11\}$ with $(H_1, H_2) = \{(8, 0), (2, 2), (2, 0)\}$. Compared with $N = 100$, $N = 50$ can increase the frequency of the evolutionary propagation of solutions, which is expected to improve the convergence of the solutions towards the Pareto optimal front; however, this setting decreases initial training samples and thus it may hinder boosting of the initial model accuracy.

TABLE II
AVERAGE IGD VALUES ON DTLZ PROBLEMS. THE BEST IGD VALUE IS HIGHLIGHTED.

(a) $D = 50$

Prob.	M	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA	MCEA/D
DTLZ1	3	1.157e+03 –	1.167e+03 –	1.132e+03 –	9.706e+02 –	9.360e+02 –	7.393e+02
	7	8.184e+02 –	8.698e+02 –	8.048e+02 –	7.358e+02 ≈	6.292e+02 +	6.963e+02
	11	6.528e+02 –	7.312e+02 –	7.291e+02 –	6.709e+02 –	5.770e+02 ≈	5.361e+02
DTLZ2	3	2.918e+00 –	2.573e+00 –	2.448e+00 –	2.029e+00 –	1.954e+00 –	6.789e-01
	7	2.881e+00 –	2.767e+00 –	2.758e+00 –	2.281e+00 –	2.133e+00 –	1.224e+00
	11	2.783e+00 –	2.525e+00 –	2.591e+00 –	2.356e+00 –	2.081e+00 –	1.442e+00
DTLZ3	3	3.847e+03 –	3.546e+03 –	3.094e+03 –	3.070e+03 –	2.886e+03 –	1.738e+03
	7	3.438e+03 –	3.243e+03 –	2.126e+03 –	2.711e+03 –	2.687e+03 –	1.540e+03
	11	3.027e+03 –	3.032e+03 –	1.866e+03 ≈	2.362e+03 –	2.221e+03 –	1.561e+03
DTLZ4	3	2.899e+00 –	3.294e+00 –	2.826e+00 –	2.396e+00 –	1.892e+00 –	1.075e+00
	7	3.268e+00 –	3.339e+00 –	2.772e+00 –	2.364e+00 –	2.028e+00 –	1.297e+00
	11	3.081e+00 –	3.060e+00 –	2.502e+00 –	2.602e+00 –	2.035e+00 –	1.333e+00
DTLZ5	3	2.770e+00 –	2.490e+00 –	2.266e+00 –	1.924e+00 –	1.845e+00 –	5.485e-01
	7	2.561e+00 –	2.517e+00 –	2.255e+00 –	1.898e+00 –	1.842e+00 –	6.087e-01
	11	2.304e+00 –	2.102e+00 –	1.986e+00 –	1.798e+00 –	1.582e+00 –	6.853e-01
DTLZ6	3	4.050e+01 –	4.041e+01 –	2.750e+01 ≈	3.364e+01 –	4.073e+01 –	2.505e+01
	7	3.730e+01 –	3.714e+01 –	2.332e+01 ≈	3.119e+01 –	3.792e+01 –	2.407e+01
	11	3.376e+01 –	3.375e+01 –	2.272e+01 ≈	2.791e+01 –	3.403e+01 –	2.128e+01
DTLZ7	3	4.842e+00 +	8.852e+00 ≈	9.040e+00 ≈	9.214e+00 ≈	6.653e+00 +	9.017e+00
	7	1.093e+01 +	3.339e+01 ≈	2.088e+01 ≈	2.285e+01 –	2.353e+01 –	2.123e+01
	11	1.772e+01 +	2.642e+01 +	3.404e+01 ≈	3.596e+01 –	3.812e+01 –	3.199e+01
+ / – / ≈		3/18/0	1/18/2	0/14/7	0/19/2	2/18/1	-

(b) $D = 100$

Prob.	M	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA	MCEA/D
DTLZ1	3	2.598e+03 –	2.748e+03 –	2.775e+03 –	2.235e+03 –	2.263e+03 –	1.705e+03
	7	1.881e+03 –	2.069e+03 –	2.028e+03 –	1.733e+03 –	1.751e+03 –	1.577e+03
	11	1.600e+03 –	1.925e+03 –	1.893e+03 –	1.604e+03 –	1.605e+03 –	1.394e+03
DTLZ2	3	6.496e+00 –	6.422e+00 –	6.406e+00 –	4.194e+00 –	5.094e+00 –	1.095e+00
	7	6.430e+00 –	6.537e+00 –	6.518e+00 –	4.467e+00 –	5.446e+00 –	1.912e+00
	11	6.245e+00 –	6.324e+00 –	6.418e+00 –	4.599e+00 –	5.339e+00 –	2.075e+00
DTLZ3	3	8.405e+03 –	8.779e+03 –	8.185e+03 –	6.912e+03 –	7.065e+03 –	3.790e+03
	7	8.145e+03 –	8.338e+03 –	7.303e+03 –	6.581e+03 –	7.176e+03 –	3.910e+03
	11	7.630e+03 –	7.848e+03 –	7.254e+03 –	6.305e+03 –	6.664e+03 –	3.889e+03
DTLZ4	3	6.602e+00 –	6.863e+00 –	-	4.824e+00 –	4.826e+00 –	1.328e+00
	7	6.727e+00 –	6.865e+00 –	-	4.968e+00 –	5.063e+00 –	1.517e+00
	11	6.549e+00 –	6.604e+00 –	-	4.680e+00 –	4.833e+00 –	1.470e+00
DTLZ5	3	6.422e+00 –	6.358e+00 –	6.345e+00 –	4.257e+00 –	4.850e+00 –	1.023e+00
	7	6.140e+00 –	6.219e+00 –	6.182e+00 –	4.182e+00 –	4.931e+00 –	1.333e+00
	11	5.869e+00 –	5.868e+00 –	5.900e+00 –	4.144e+00 –	4.681e+00 –	1.529e+00
DTLZ6	3	8.448e+01 –	8.679e+01 –	7.212e+01 –	7.123e+01 –	8.593e+01 –	5.429e+01
	7	8.116e+01 –	8.303e+01 –	7.747e+01 –	6.924e+01 –	8.262e+01 –	5.440e+01
	11	7.784e+01 –	7.961e+01 –	6.908e+01 –	6.537e+01 –	7.907e+01 –	5.093e+01
DTLZ7	3	6.547e+00 +	1.066e+01 –	1.051e+01 –	9.933e+00 ≈	7.525e+00 +	9.991e+00
	7	1.718e+01 +	2.566e+01 –	2.578e+01 –	2.533e+01 –	2.562e+01 –	2.374e+01
	11	2.866e+01 +	4.196e+01 –	4.020e+01 –	4.029e+01 –	4.197e+01 –	3.844e+01
+ / – / ≈		3/18/0	0/21/0	0/18/0	0/20/1	1/20/0	-

(c) $D = 150$

Prob.	M	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA	MCEA/D
DTLZ1	3	4.130e+03 –	4.302e+03 –	4.343e+03 –	3.527e+03 –	3.618e+03 –	2.636e+03
	7	2.911e+03 –	3.231e+03 –	3.198e+03 –	2.711e+03 –	2.915e+03 –	2.376e+03
	11	2.602e+03 –	3.175e+03 –	3.107e+03 –	2.554e+03 –	2.791e+03 –	2.246e+03
DTLZ2	3	1.026e+01 –	1.026e+01 –	1.031e+01 –	6.504e+00 –	8.276e+00 –	1.489e+00
	7	1.017e+01 –	1.031e+01 –	1.022e+01 –	7.265e+00 –	8.898e+00 –	2.647e+00
	11	1.000e+01 –	1.015e+01 –	1.023e+01 –	7.340e+00 –	8.512e+00 –	2.984e+00
DTLZ3	3	1.348e+04 –	1.382e+04 –	1.304e+04 –	1.078e+04 –	1.172e+04 –	6.303e+03
	7	1.289e+04 –	1.336e+04 –	1.244e+04 –	1.049e+04 –	1.175e+04 –	6.114e+03
	11	1.268e+04 –	1.300e+04 –	1.249e+04 –	1.026e+04 –	1.136e+04 –	5.895e+03
DTLZ4	3	1.024e+01 –	1.064e+01 –	-	6.954e+00 –	7.720e+00 –	1.556e+00
	7	1.053e+01 –	1.060e+01 –	-	6.922e+00 –	8.352e+00 –	1.751e+00
	11	1.032e+01 –	1.038e+01 –	-	7.419e+00 –	7.987e+00 –	1.637e+00
DTLZ5	3	1.015e+01 –	1.027e+01 –	1.018e+01 –	6.738e+00 –	8.380e+00 –	1.536e+00
	7	9.914e+00 –	9.982e+00 –	1.005e+01 –	6.391e+00 –	8.636e+00 –	2.149e+00
	11	9.615e+00 –	9.610e+00 –	9.602e+00 –	6.140e+00 –	8.398e+00 –	2.441e+00
DTLZ6	3	1.293e+02 –	1.318e+02 –	1.189e+02 –	1.094e+02 –	1.307e+02 –	8.780e+01
	7	1.254e+02 –	1.280e+02 –	1.229e+02 –	1.073e+02 –	1.271e+02 –	8.293e+01
	11	1.221e+02 –	1.244e+02 –	1.179e+02 –	1.047e+02 –	1.238e+02 –	8.175e+01
DTLZ7	3	7.739e+00 +	1.089e+01 –	1.077e+01 –	1.058e+01 ≈	8.342e+00 +	1.042e+01
	7	2.072e+01 +	2.649e+01 –	2.632e+01 –	2.598e+01 –	2.665e+01 –	2.531e+01
	11	3.354e+01 +	4.300e+01 –	4.215e+01 –	4.154e+01 –	4.264e+01 –	3.994e+01
+ / – / ≈		3/18/0	0/21/0	0/18/0	0/20/1	1/20/0	-

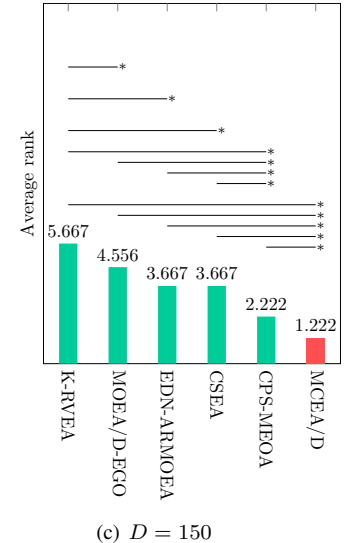
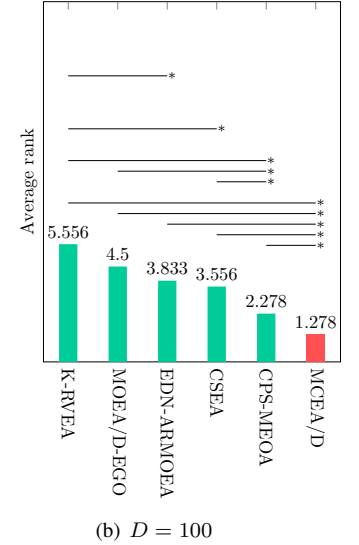
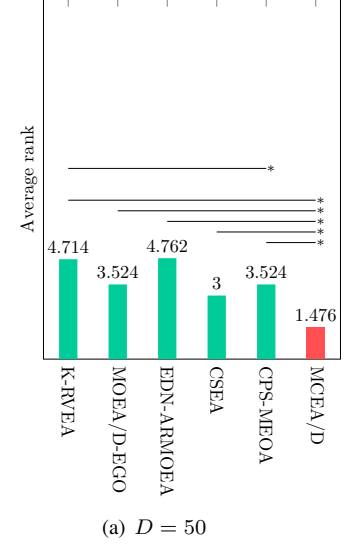


Fig. 1. Average ranks and the multiple test results on DTLZ problems.

TABLE III
AVERAGE IGD VALUES ON WFG PROBLEMS. THE BEST IGD VALUE IS HIGHLIGHTED.

(a) $D = 50$

Prob.	M	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA	MCEA/D
WFG1	3	2.118e+00	2.360e+00	2.079e+00	2.290e+00	1.752e+00	2.283e+00
	7	2.818e+00	2.981e+00	2.868e+00	2.972e+00	2.768e+00	3.005e+00
	11	3.506e+00	3.636e+00	3.610e+00	3.634e+00	3.572e+00	3.659e+00
WFG2	3	8.535e-01	1.011e+00	8.761e-01	8.716e-01	6.801e-01	7.214e-01
	7	1.793e+00	3.005e+00	2.563e+00	2.039e+00	2.152e+00	2.181e+00
	11	3.106e+00	4.819e+00	4.655e+00	3.645e+00	4.773e+00	4.261e+00
WFG3	3	7.971e-01	7.772e-01	7.766e-01	7.985e-01	7.511e-01	5.811e-01
	7	1.421e+00	1.400e+00	1.443e+00	1.402e+00	1.323e+00	1.062e+00
	11	1.947e+00	1.958e+00	1.978e+00	1.900e+00	1.813e+00	1.357e+00
WFG4	3	5.688e-01	5.653e-01	6.318e-01	6.089e-01	5.353e-01	5.434e-01
	7	3.354e+00	2.950e+00	4.699e+00	3.590e+00	4.583e+00	3.534e+00
	11	8.696e+00	7.892e+00	1.127e+01	8.946e+00	1.165e+01	8.958e+00
WFG5	3	7.332e-01	7.446e-01	7.467e-01	6.848e-01	7.372e-01	6.184e-01
	7	3.097e+00	2.798e+00	3.910e+00	3.455e+00	3.635e+00	3.322e+00
	11	8.329e+00	6.240e+00	9.580e+00	8.724e+00	9.160e+00	8.317e+00
WFG6	3	9.456e-01	9.270e-01	9.932e-01	9.727e-01	9.038e-01	8.410e-01
	7	2.784e+00	3.120e+00	4.191e+00	3.632e+00	3.914e+00	3.672e+00
	11	6.551e+00	6.041e+00	9.803e+00	8.874e+00	9.425e+00	9.160e+00
WFG7	3	7.246e-01	7.237e-01	7.825e-01	7.347e-01	7.043e-01	6.252e-01
	7	2.625e+00	2.831e+00	4.394e+00	3.724e+00	3.907e+00	4.030e+00
	11	6.592e+00	6.430e+00	1.037e+01	9.359e+00	9.885e+00	1.026e+01
WFG8	3	8.047e-01	7.702e-01	8.907e-01	8.370e-01	7.774e-01	7.710e-01
	7	2.870e+00	3.092e+00	4.302e+00	3.695e+00	3.943e+00	3.758e+00
	11	6.785e+00	6.159e+00	1.005e+01	8.939e+00	9.766e+00	9.318e+00
WFG9	3	1.001e+00	1.007e+00	1.036e+00	1.004e+00	9.358e-01	7.746e-01
	7	3.977e+00	3.611e+00	4.440e+00	4.195e+00	4.392e+00	3.679e+00
	11	9.374e+00	8.515e+00	1.063e+01	9.730e+00	1.026e+01	9.059e+00
+/-/≈		12/11/4	11/10/6	3/2/2	4/14/9	4/16/7	-

(b) $D = 100$

Prob.	M	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA	MCEA/D
WFG1	3	2.149e+00	2.398e+00	2.044e+00	2.303e+00	1.761e+00	2.284e+00
	7	2.848e+00	3.032e+00	2.909e+00	2.977e+00	2.797e+00	2.995e+00
	11	3.542e+00	3.691e+00	3.624e+00	3.639e+00	3.570e+00	3.664e+00
WFG2	3	8.751e-01	1.077e+00	9.313e-01	8.874e-01	7.580e-01	7.260e-01
	7	1.874e+00	3.141e+00	2.672e+00	1.989e+00	2.283e+00	2.214e+00
	11	3.177e+00	5.512e+00	5.126e+00	3.503e+00	4.298e+00	4.109e+00
WFG3	3	8.348e-01	8.514e-01	8.490e-01	8.282e-01	8.098e-01	6.057e-01
	7	1.450e+00	1.509e+00	1.494e+00	1.434e+00	1.409e+00	1.100e+00
	11	1.980e+00	2.086e+00	2.077e+00	1.950e+00	1.908e+00	1.429e+00
WFG4	3	5.958e-01	7.928e-01	6.535e-01	6.173e-01	5.680e-01	5.569e-01
	7	3.976e+00	5.398e+00	4.976e+00	3.525e+00	4.547e+00	3.525e+00
	11	1.015e+01	1.230e+01	1.190e+01	8.841e+00	1.200e+01	9.140e+00
WFG5	3	7.705e-01	8.573e-01	7.969e-01	7.095e-01	7.616e-01	6.358e-01
	7	3.516e+00	3.980e+00	4.034e+00	3.413e+00	3.522e+00	3.341e+00
	11	9.009e+00	9.457e+00	1.014e+01	8.664e+00	8.897e+00	8.488e+00
WFG6	3	9.651e-01	1.052e+00	1.022e+00	9.936e-01	9.487e-01	8.831e-01
	7	2.854e+00	4.466e+00	4.375e+00	3.673e+00	3.836e+00	3.720e+00
	11	6.825e+00	1.027e+01	1.035e+01	8.930e+00	9.346e+00	9.289e+00
WFG7	3	7.334e-01	8.545e-01	8.114e-01	7.533e-01	7.312e-01	6.439e-01
	7	2.698e+00	4.483e+00	4.359e+00	3.679e+00	3.893e+00	3.862e+00
	11	6.831e+00	1.077e+01	1.068e+01	9.258e+00	9.466e+00	9.958e+00
WFG8	3	7.971e-01	9.087e-01	8.936e-01	8.315e-01	7.904e-01	7.584e-01
	7	2.881e+00	4.404e+00	4.442e+00	3.618e+00	3.964e+00	3.717e+00
	11	7.058e+00	1.017e+01	1.046e+01	9.045e+00	9.539e+00	9.402e+00
WFG9	3	1.054e+00	1.147e+00	1.097e+00	1.057e+00	9.996e-01	8.109e-01
	7	4.028e+00	4.627e+00	4.635e+00	4.112e+00	4.255e+00	3.623e+00
	11	9.590e+00	1.072e+01	1.086e+01	9.651e+00	1.001e+01	8.783e+00
+/-/≈		11/16/0	0/27/0	3/24/0	5/12/10	4/16/7	-

(c) $D = 150$

Prob.	M	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA	MCEA/D
WFG1	3	2.157e+00	2.423e+00	2.022e+00	2.299e+00	1.758e+00	2.269e+00
	7	2.872e+00	3.024e+00	2.913e+00	2.979e+00	2.773e+00	3.003e+00
	11	3.551e+00	3.693e+00	3.618e+00	3.641e+00	3.566e+00	3.663e+00
WFG2	3	8.738e-01	1.085e+00	9.331e-01	8.897e-01	7.864e-01	7.528e-01
	7	1.891e+00	3.174e+00	2.847e+00	2.072e+00	2.499e+00	2.117e+00
	11	3.455e+00	5.607e+00	4.783e+00	3.582e+00	4.289e+00	3.961e+00
WFG3	3	8.467e-01	8.636e-01	8.628e-01	8.292e-01	8.341e-01	6.116e-01
	7	1.453e+00	1.528e+00	1.509e+00	1.457e+00	1.437e+00	1.120e+00
	11	1.988e+00	2.089e+00	2.091e+00	1.958e+00	1.937e+00	1.436e+00
WFG4	3	6.274e-01	8.104e-01	6.505e-01	6.248e-01	5.632e-01	5.525e-01
	7	4.233e+00	5.495e+00	4.918e+00	3.476e+00	4.404e+00	3.488e+00
	11	1.068e+01	1.276e+01	1.199e+01	8.867e+00	1.172e+01	9.123e+00
WFG5	3	7.909e-01	8.606e-01	8.226e-01	7.137e-01	7.756e-01	6.346e-01
	7	3.592e+00	4.092e+00	4.057e+00	3.460e+00	3.590e+00	3.314e+00
	11	8.959e+00	9.974e+00	1.012e+01	8.715e+00	9.016e+00	8.444e+00
WFG6	3	9.727e-01	1.054e+00	1.037e+00	1.002e+00	9.571e-01	8.786e-01
	7	2.892e+00	4.493e+00	4.500e+00	3.595e+00	3.815e+00	3.761e+00
	11	7.114e+00	1.062e+01	1.073e+01	8.874e+00	9.236e+00	9.314e+00
WFG7	3	7.380e-01	8.548e-01	8.171e-01	7.551e-01	7.379e-01	6.507e-01
	7	2.760e+00	4.458e+00	4.533e+00	3.613e+00	3.826e+00	3.830e+00
	11	7.220e+00	1.064e+01	1.077e+01	9.229e+00	9.254e+00	9.620e+00
WFG8	3	7.964e-01	9.122e-01	8.989e-01	8.289e-01	7.985e-01	7.503e-01
	7	2.898e+00	4.402e+00	4.365e+00	3.531e+00	3.881e+00	3.662e+00
	11	7.130e+00	1.042e+01	1.045e+01	8.787e+00	9.540e+00	9.222e+00
WFG9	3	1.069e+00	1.153e+00	1.109e+00	1.077e+00	1.001e+00	8.176e-01
	7	4.049e+00	4.747e+00	4.526e+00	4.081e+00	4.146e+00	3.514e+00
	11	9.605e+00	1.068e+01	1.061e+01	9.616e+00	9.855e+00	8.683e+00
+/-/≈		10/16/1	0/27/0	3/24/0	4/15/8	3/17/7	-

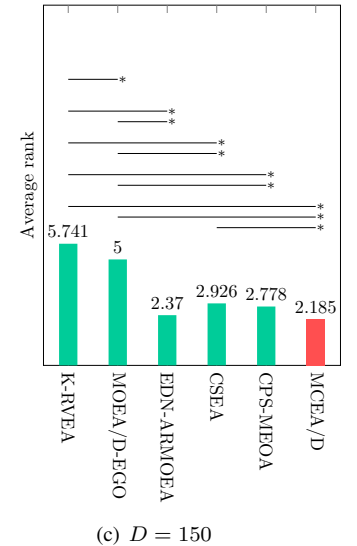
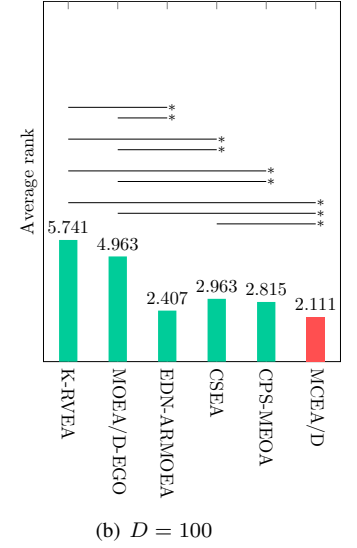
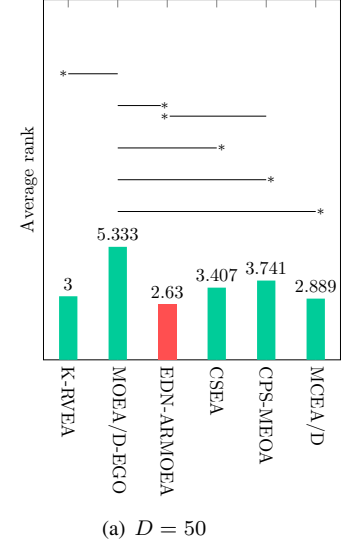


Fig. 2. Average ranks and the multiple test results on WFG problems.

TABLE IV
AVERAGE COMPUTATIONAL TIME [sec] OF ALL THE SIX ALGORITHMS.

D	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA	MCEA/D
50	2.435e+02	1.555e+02	1.053e+03	4.477e-02	4.211e+01	3.203e+00
100	3.580e+02	4.789e+02	3.235e+03	5.284e-02	4.709e+01	4.135e+00
150	5.348e+02	1.015e+03	7.206e+03	5.819e-02	8.004e+01	4.882e+00

TABLE V
THE STATISTICAL RESULTS OF THE WILCOXON RANK-SUM TEST WITH $N = 50$, SUMMARIZED AS A SUMMATION OF “+ / - / \approx ”.

(a) DTLZ

D	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA
50	3/18/0	2/18/1	0/16/5	0/17/4	1/18/2
100	3/18/0	0/19/2	0/17/1	0/17/4	1/20/0
150	3/18/0	0/19/2	0/17/1	0/18/3	1/20/0

(b) WFG

D	EDN-ARMOEA	K-RVEA	MOEA/D-EGO	CPS-MOEA	CSEA
50	10/11/6	10/10/7	3/22/2	11/12/4	4/15/8
100	10/16/1	0/25/2	3/23/1	10/13/4	4/18/5
150	10/16/1	0/27/0	3/24/0	11/13/3	3/18/6

Table V summarizes the statistical results of the Wilcoxon rank-sum test, summarized as “+ / - / \approx ”. The complete summary of averaged IGD values and the multiple test results are reported in Section A of our supplementary material. Note that for $D = \{100, 150\}$, MOEA/D-EGO failed to solve DTLZ4. As shown in the table, different from the results with $N = 100$, the performance of CPS-MOEA improves on the WFG problems. The number of “+” clearly increases compared with that of $N = 100$ (see Table III). Thus, CPS-MOEA can withstand the decrease in the number of initial training samples. However, the overall results support the effectiveness of MCEA/D as it stably derives better performances than the alternative algorithms.

C. Impact of Maximum repeat time

We analyze the impact of both the hyper-parameter R_{\max} and our backup strategy, that is, selecting the candidate solution closest to a decision boundary of an SVM model. We set $R_{\max} = \{1, 10, 100, 1000\}$, where a large value of R_{\max} increases a probability of generating offspring solutions with the positive class. Tables VI and VII report the results of the Wilcoxon rank-sum test and the averaged computational time [sec], respectively, where the same experimental settings were used as in Section V. In Table VI, “+”, “-”, and “ \approx ” denote that the IGD value of MCEA/D with $R_{\max} = \{1, 100, 1000\}$ is significantly better, worse, and competitive compared to that of a default value $R_{\max} = 10$, respectively.

MCEA/D with $R_{\max} = 1$ is worse than that with $R_{\max} = 10$ because $R_{\max} = 1$ is equivalent to not utilizing the surrogate models. $R_{\max} = 1000$ slightly improves the performance of MCEA/D with $R_{\max} = 10$ for $D = 50$. However, from the overall results, $R_{\max} = 10$ can be identified as the best setting in our experimental settings, as it derives competitive performances both on the DTLZ and WFG problems while reducing the computational time one order of magnitude smaller than that of $R_{\max} = \{100, 1000\}$. This observation

suggests that our backup strategy has a sufficient capacity to estimate high-quality solutions, as no significant impact of R_{\max} is obtained for $R_{\max} \geq 10$.

VII. CONCLUSION

In this paper, we have presented a novel classification-based SAEA, called MCEA/D, adapted for high-dimensional EMOPs/EMaOPs. The following three strategies are implemented in MCEA/D. First, SVM-based classifiers are employed to build robust surrogate models for few samples of high-dimensional training inputs. Second, multiple local classifiers are utilized to hedge the problematic over-fitting issue, and thus a decomposition-based MOEA, that is, MOEA/D, is employed to assign a local classifier to each sub-problem. Finally, the pre-screening capacity of MCEA/D is boosted via a backup strategy that selects the candidate solution closest to the decision boundary.

MCEA/D was compared with the state-of-the-art algorithms, such as, EDN-ARMOEA, MOEA/D-EGO, K-RVEA, CPS-MOEA, and CSEA. The results show that MCEA/D is powerful and even computationally efficient on DTLZ and WFG problems with up to 150 decision variables and 11 objectives. The computational time of MCEA/D is reduced at least one order of magnitude smaller than the other algorithms except for CPS-MOEA. Moreover, this effectiveness of MCEA/D is achieved without any specialized experimental settings for high-dimensional problems with regard to the number of initial solutions and of fitness evaluations. For instance, modern works tend to increase the number of training samples (e.g., $11D - 1$ [18]) and the maximum fitness evaluations (e.g., 1000 [17]) on high-dimensional EMOPs/EMaOPs. This study assumed N initial training samples with 300 FEs.

Our results have revealed the potential of classification-based SAEAs on high-dimensional EMOPs/EMaOPs. CPS-MOEA and CSEA were also found to be powerful and even computationally efficient compared to modern SAEAs, that is, K-RVEA and MOEA/D-EGO. Moreover, the effectiveness of EDN-ARMOEA has been also confirmed compared to the classification-based SAEAs. Note that this tendency may be changed depending on experimental conditions. For instance, the effectiveness of the approximation-based SAEAs can be more enhanced with sufficient number of initial training samples. Thus, we assume that the classification-based SAEAs are suitable if the number of pre-evaluated solutions is limited. It would be worth further comparing to the approximation/classification-based SAEAs, but we leave it as future work as it is out of our scope in this manuscript.

In future works, MCEA/D will be extended for constrained EMOPs/EMaOPs, wherein classifiers can be further utilized to estimate feasible solutions. In addition, it would be worth investigating the effect of dimension reduction techniques on the performance of classification-based SAEAs.

REFERENCES

- [1] R. Shiratori, M. Nakata, K. Hayashi, and T. Baba, “Particle swarm optimization of silicon photonic crystal waveguide transition,” *Optics Letters*, vol. 46, no. 8, pp. 1904–1907, 2021.

TABLE VI
RESULTS OF THE WILCOXON RANK-SUM TEST WITH
 $R_{\max} = \{1, 100, 1000\}$ (“+ / - / \approx ”).

(a) DTLZ			
D	$R_{\max} = 1$	$R_{\max} = 100$	$R_{\max} = 1000$
50	0/13/8	8/4/9	9/5/7
100	0/12/9	9/10/2	9/9/3
150	0/11/10	9/10/2	8/10/3

(b) WFG			
D	$R_{\max} = 1$	$R_{\max} = 100$	$R_{\max} = 1000$
50	5/13/9	7/7/13	11/8/8
100	4/13/10	9/9/9	10/9/8
150	1/13/13	9/7/11	7/11/9

TABLE VII
AVERAGE COMPUTATIONAL TIME [sec] OF MCEA/D WITH
 $R_{\max} = \{1, 10, 100, 1000\}$.

(a) The number of decision variables				
D	$R_{\max} = 1$	$R_{\max} = 10$	$R_{\max} = 100$	$R_{\max} = 1000$
50	2.435e+00	3.203e+00	1.028e+01	7.948e+01
100	3.248e+00	4.121e+00	1.191e+01	9.212e+01
150	3.899e+00	4.870e+00	1.332e+01	9.884e+01

(b) The number of objectives				
M	$R_{\max} = 1$	$R_{\max} = 10$	$R_{\max} = 100$	$R_{\max} = 1000$
3	3.179e+00	3.956e+00	1.053e+01	7.585e+01
7	3.192e+00	4.061e+00	1.196e+01	9.196e+01
11	3.212e+00	4.142e+00	1.302e+01	1.026e+02

- [2] Y. Jin, “Surrogate-assisted evolutionary computation: Recent advances and future challenges,” *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
- [3] Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, and W. Banzhaf, “NSGA-Net: neural architecture search using multi-objective genetic algorithm,” in *2019 Genetic and Evolutionary Computation Conference (GECCO)*, 2019, pp. 419–427.
- [4] Z. Lu, K. Deb, E. Goodman, W. Banzhaf, and V. N. Boddeti, “NS-GANetV2: Evolutionary Multi-objective Surrogate-Assisted Neural Architecture Search,” in *2020 European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 35–51.
- [5] T. Elsken, J. H. Metzen, and F. Hutter, “Efficient Multi-objective Neural Architecture Search via Lamarckian Evolution,” *arXiv preprint arXiv:1804.09081*, 2018.
- [6] H. Shi, R. Pi, H. Xu, Z. Li, J. T. Kwok, and T. Zhang, “Multi-objective Neural Architecture Search via Predictive Network Performance Optimization,” *arXiv preprint arXiv:1911.09336*, 2019.
- [7] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.
- [8] R. Allmendinger, M. T. Emmerich, J. Hakanen, Y. Jin, and E. Rigoni, “Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case,” *Journal of Multi-Criteria Decision Analysis*, vol. 24, no. 1–2, pp. 5–24, 2017.
- [9] 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, p. 9420460, 2016.
- [10] 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, no. 9, pp. 3137–3166, 2019.
- [11] 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 Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2016.
- [12] 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 Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 74–88, 2018.
- [13] C. Luo, K. Shimoyama, and S. Obayashi, “Kriging model based many-objective optimization with efficient calculation of expected hypervolume improvement,” in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 1187–1194.
- [14] M. Pilát and R. Neruda, “ASM-MOMA: Multiobjective memetic algorithm with aggregate surrogate model,” in *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, 2011, pp. 1202–1208.
- [15] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, “Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted S -Metric Selection,” in *2008 International Conference on Parallel Problem Solving from Nature (PPSN)*. Springer, 2008, pp. 784–794.
- [16] 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*, p. 107192, 2020.
- [17] J. Lin, C. He, and R. Cheng, “Adaptive dropout for high-dimensional expensive multiobjective optimization,” *Complex & Intelligent Systems*, pp. 1–15, 2021.
- [18] 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 Transactions on Systems, Man, and Cybernetics: Systems*, 2021.
- [19] L. Shi and K. Rasheed, “A Survey of Fitness Approximation Methods Applied in Evolutionary Algorithms,” in *Computational Intelligence in Expensive Optimization Problems*. Springer, 2010, pp. 3–28.
- [20] 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.
- [21] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” in *1992 Annual Workshop on Computational Learning Theory (COLT)*. ACM, 1992, pp. 144–152.
- [22] Q. Zhang and H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [23] V. N. Vapnik, “An Overview of Statistical Learning Theory,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [24] T. Sonoda and M. Nakata, “MOEA/D-S³: MOEA/D using SVM-based Surrogates adjusted to Subproblems for Many objective optimization,” in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.
- [25] J. Knowles, “ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.
- [26] Q. Zhang, W. Liu, E. Tsang, and B. Virginias, “Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2009.
- [27] 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.
- [28] D. Zhan, Y. Cheng, and J. Liu, “Expected Improvement Matrix-Based Infill Criteria for Expensive Multiobjective Optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 956–975, 2017.
- [29] Z. Song, H. Wang, C. He, and Y. Jin, “A Kriging-Assisted Two-Archive Evolutionary Algorithm for Expensive Many-Objective Optimization,” *IEEE Transactions on Evolutionary Computation*, 2021.
- [30] 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.
- [31] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, “A Multiple Surrogate Assisted Decomposition Based Evolutionary Algorithm for Expensive Multi/Many-Objective Optimization,” *IEEE Transactions on Evolutionary Computation*, 2019.
- [32] S. Zapotecas Martínez and C. A. Coello Coello, “MOEA/D assisted by RBF networks for expensive multi-objective optimization problems,” in *2013 Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2013, pp. 1405–1412.
- [33] A. Rosales-Pérez, C. A. C. Coello, J. A. Gonzalez, C. A. Reyes-Garcia, and H. J. Escalante, “A hybrid surrogate-based approach for evolutionary multi-objective optimization,” in *2013 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2013, pp. 2548–2555.
- [34] D. Guo, Y. Jin, J. Ding, and T. Chai, “Heterogeneous Ensemble-Based Infill Criterion for Evolutionary Multiobjective Optimization of Expensive Problems,” *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1012–1025, 2018.

- [35] I. Loshchilov, M. Schoenauer, and M. Sebag, "A Mono Surrogate for Multiobjective Optimization," in *2010 Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2010, pp. 471–478.
- [36] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 2883–2890.
- [37] —, "A Multiobjective Evolutionary Algorithm Based on Decomposition and Preselection," in *Bio-inspired Computing: Theories and Applications*. Springer, 2015, pp. 631–642.
- [38] X. Lin, Q. Zhang, and S. Kwong, "A decomposition based multiobjective evolutionary algorithm with classification," in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 3292–3299.
- [39] N. Berviglieri, B. Derbel, A. Liefoghe, H. Aguirre, and K. Tanaka, "Surrogate-assisted Multiobjective Optimization based on Decomposition: A Comprehensive Comparative Analysis," in *2019 Genetic and Evolutionary Computation Conference (GECCO)*, 2019, pp. 507–515.
- [40] R. Liu, R. Ren, J. Liu, and J. Liu, "A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems," *Applied Soft Computing*, vol. 89, p. 106120, 2020.
- [41] Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin, "Solving Large-Scale Multiobjective Optimization Problems With Sparse Optimal Solutions via Unsupervised Neural Networks," *IEEE Transactions on Cybernetics*, 2020.
- [42] Z. Tan, H. Wang, and S. Liu, "Multi-stage dimension reduction for expensive sparse multi-objective optimization problems," *Neurocomputing*, vol. 440, pp. 159–174, 2021.
- [43] K. Deb, D. Saxena *et al.*, "Searching For Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems," in *2006 IEEE Congress on Evolutionary Computation (CEC)*, 2006, pp. 3352–3360.
- [44] A. Zhou, Y. Wang, and J. Zhang, "Objective extraction via fuzzy clustering in evolutionary many-objective optimization," *Information Sciences*, vol. 509, pp. 343–355, 2020.
- [45] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer Learning-Based Dynamic Multiobjective Optimization Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, 2017.
- [46] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodelling," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, 2006.
- [47] R. G. Regis, "Evolutionary Programming for High-Dimensional Constrained Expensive Black-Box Optimization Using Radial Basis Functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2013.
- [48] P. Singh, I. Couckuyt, F. Ferranti, and T. Dhaene, "A constrained multi-objective surrogate-based optimization algorithm," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 3080–3087.
- [49] R. Datta and R. G. Regis, "A surrogate-assisted evolution strategy for constrained multi-objective optimization," *Expert Systems with Applications*, vol. 57, pp. 270–284, 2016.
- [50] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [51] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
- [52] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design," *arXiv preprint arXiv:0912.3995*, 2009.
- [53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [54] 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.
- [55] Q. Xu, Z. Xu, and T. Ma, "A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition: Variants, Challenges and Future Directions," *IEEE Access*, vol. 8, pp. 41 588–41 614, 2020.
- [56] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 203–208.
- [57] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A Multiobjective Evolutionary Algorithm Using Decomposition and AntColony," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1845–1859, 2013.
- [58] Y. Li, A. Zhou, and G. Zhang, "An MOEA/D with multiple differential evolution mutation operators," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 397–404.
- [59] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
- [60] K. Deb, "Multi-objective Optimization," in *Search Methodologies*. Springer, 2014, pp. 403–449.
- [61] J. Shawe-Taylor and S. Sun, "A review of optimization methodologies in support vector machines," *Neurocomputing*, vol. 74, no. 17, pp. 3609–3618, 2011.
- [62] A. Bordes, S. Ertekin, J. Weston, L. Botton, and N. Cristianini, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.
- [63] A. Abdiansah and R. Wardoyo, "Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM," *International Journal of Computer and Applications*, vol. 128, no. 3, pp. 28–34, 2015.
- [64] J. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," *Technical report*, 1998.
- [65] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multiobjective Optimization," in *Evolutionary Multiobjective Optimization*. Springer, 2005, pp. 105–145.
- [66] S. Huband, L. Barone, L. While, and P. Hingston, "A Scalable Multi-objective Test Problem Toolkit," in *2005 International Conference on Evolutionary Multi-Criterion Optimization (EMO)*. Springer, 2005, pp. 280–295.
- [67] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
- [68] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.
- [69] 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.
- [70] C. A. C. Coello and N. C. Cortés, "Solving Multiobjective Optimization Problems Using an Artificial Immune System," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.



Takumi Sonoda received his B.Eng. in 2019 from Yokohama National University, Japan. He is a graduate student at Graduate school of Engineering Science, Yokohama National University. His research interests include multi-objective optimization and surrogate-assisted evolutionary optimization.



Masaya Nakata is an associate professor at 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 been mainly working on Evolutionary Machine Learning, Data mining, and more specifically, theoretical analysis of evolutionary rule-based learning. Since 2019, he has focused his research on surrogate-assisted evolutionary algorithm. His contributions have been published through more than 15 journal papers and 35 conference papers, for example, IEEE T-EVC, IEEE CEC, GECCO, and PPSN. He was an organizing committee member of International Workshop on Learning Classifier Systems 2015-2016 and 2018-2019 in GECCO.