Grid Classification-based Surrogate-Assisted Particle Swarm Optimization for Expensive Multiobjective Optimization

Qi-Te Yang, *Student Member, IEEE,* Zhi-Hui Zhan, *Fellow, IEEE,* Xiao-Fang Liu, *Member, IEEE,* Jian-Yu Li, *Member, IEEE,* and Jun Zhang, *Fellow, IEEE*

***Abstract*—Surrogate-assisted evolutionary algorithms (SAEAs), mainly including regression-based SAEAs and classification-based SAEAs, are promising for solving expensive multiobjective optimization problems (EMOPs). Regression-based SAEAs usually use complex regression models to approximate the fitness evaluation, which will suffer from high training costs to obtain a fine-accuracy surrogate. In contrast, classification-based SAEAs can achieve solution selection via coarse binary relations predicted by classifiers, thus avoiding high requirements in prediction accuracy and training costs. However, most of the binary relations in existing classification-based SAEAs mainly only involve convergence comparison whereas diversity maintenance is neglected. Considering the capacity of the grid technique in maintaining both convergence and diversity, we propose a new classification method called *grid classification* to discretize the objective space into grids and train a lightweight grid classification-based surrogate (GCS), for which low training costs are needed. The GCS can evaluate the solution performance in terms of both convergence and diversity simultaneously according to the predicted grid locations, which opens up a new field for follow-up research on classification-based SAEAs. Following this, a GCS-assisted particle swarm optimization algorithm is proposed for tackling EMOPs. Experimental results on widely-used benchmark problems (including high-dimensional EMOPs) and a 222-high-dimensional real-world application problem show its competitiveness in terms of both optimization performance and computational cost.**

***Index Terms*—Surrogate-assisted evolutionary algorithm (SAEA), evolutionary computation, expensive multiobjective optimization, particle swarm optimization, grid classification.**

Manuscript received XXXX; revised XXXX; accepted XXXX. This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0120001, in part by the National Natural Science Foundations of China (NSFC) under Grants 62176094, U23B2039, and 62103202, in part by the Tianjin Top Scientist Studio Project under Grant 23JCQNJC00168, and in part by the National Research Foundation of Korea (NRF) under Grant NRF-2022H1D3A2A01093478. (*Corresponding authors: Zhi-Hui Zhan; Jun Zhang.*)

Qi-Te Yang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

Zhi-Hui Zhan is with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China and also with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: zhanapollo@163.com).

Xiao-Fang Liu and Jian-Yu Li are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China.

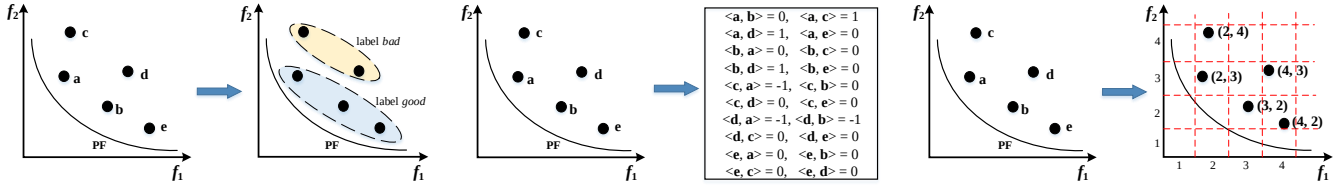
Jun Zhang is with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China and also with the Hanyang University, Ansan 15588, South Korea.

I. Introduction

MULTIOBJECTIVE optimization problems (MOPs), which arise naturally in many real-world applications [1], have been well-solved by various multiobjective evolutionary algorithms (MOEAs) [2]-[4]. These MOEAs are inspired by natural phenomena and can find a set of trade-off optimal (i.e., Pareto) solutions. After decades of development, plenty of MOEAs have been proposed and they can be roughly divided into four categories: dominance-based [5], decomposition-based [6], indicator-based [7], and multipopulation-based [8]-[12] methods. These MOEAs usually require a number of real fitness evaluations (FEs) to search for an optimal solution set. However, real FEs in many practical problems and MOPs are time-consuming or financially costly, such as in computational fluid dynamics [13], modern logistic management [14], blast furnace optimization[15], feature selection [16][17], airfoil design [18], power electronic circuit design [19], and deep neural networks design [20][21]. These MOPs are called expensive MOPs (EMOPs). To solve the EMOPs efficiently, surrogate-assisted evolutionary algorithms (SAEAs) have been proposed.

SAEAs adopt a cheap alternative called surrogate to replace the real expensive FE in EMOPs. Such a surrogate is usually made of machine learning models such as radial basis function network (RBFN) [22], feedforward neural network (FNN), Kriging model [23], or support vector machine (SVM) [24]. These surrogates estimate the fitness value of a solution or determine whether a solution is better than another one. Depending on the effect of surrogates, existing SAEAs can be divided into two categories: regression-based and classification-based.

Regression-based SAEAs use regression models (e.g., Kriging and/or RBFN) to approximate the real expensive FE of the multiple objective functions to reduce the computational cost. The fusion of these regression models forms the surrogate. For example, the Kriging model was used in [25]-[29] to approximate each objective function, and showed strong capacity in solving EMOPs. In [30], the multitask Gaussian process was adopted to build an information transfer surrogate. In [31], Kriging was used to approximate aggregation functions to realize fitness estimation. In [32], a Kriging-based Pareto front (PF) estimation method was adopted to approximate the knee-oriented aggregated function to replace the real FE in EMOPs. RBFN is another regression model commonly used in



(a) rank classification (b) relation classification (c) grid classification

Fig. 1. Different classification ways in classification-based SAEAs.

SAEAs, and it was adopted for approximating FE in [33][34]. Ensemble surrogates are also used in some regression-based SAEAs. For example, the least-square SVM and RBFN were adopted in the heterogeneous ensemble-based MOEA (HeE-MOEA) [35] for ensemble modeling. Four different regression models were used in [36] to approximate each objective, and the one with minimum prediction error would be finally adopted to replace the real FE. For more recent progress in related research, readers can refer to [37]-[39]. However, most of these regression-based SAEAs, especially the Kriging-based SAEAs, are mainly only suitable for low-dimensional EMOPs. For example, Kriging-based SAEAs are generally applicable to problems with less than 20 decision variables [26][35]. This may be due to that when the dimension of EMOPs increases, more data are required for training a refined surrogate, which will significantly increase the training costs (e.g., training time) [40]. Restricted by the quantity of data that can be obtained in real-world EMOP applications, most of these regression-based SAEAs therefore cannot train a refined surrogate with high accuracy.

Classification-based SAEAs are also proposed in the literature by considering that solution selection depends only on the binary relations between solutions [41]. In these SAEAs, surrogates are used to predict the binary relations (i.e., *good* or *bad*) among solutions but not the fitness value. Then the selection decision can be made through these predicted relations. To be specific, SAEAs adopt a classifier as the surrogate to predict whether a candidate solution has potential good performance. Compared with regression-based SAEAs, the merits of classification-based SAEAs mainly lie in two aspects: 1) solution selection does not require the refined fitness values but the coarse binary relations, thus surrogates do not have to be subjected to a high prediction accuracy; 2) according to 1), the surrogates in classification-based SAEAs can be composed of lightweight classification models, which will greatly reduce the training costs. Considering the above merits, the study of this paper also focuses on the classification-based SAEAs.

In existing classification-based SAEAs, the classifier generally works in two ways: *rank classification* and *relation classification*.

In *rank classification*, all already evaluated solutions are collected to form a database (DB), in which all solutions are then labeled as *good* or *bad* according to their performance (e.g., the Pareto domination rank information). These labeled solutions are used to train a classification surrogate, which can determine whether a candidate solution without real fitness value belongs to *good* or *bad*. For example, in [42], all solutions in the population are classified into two categories by a classification tree [43] and *K*-nearest neighbor (KNN) [44] method, and only solutions judged as *good* will be preserved. After that, this classification preselection method was inserted into the regularity model-based multiobjective estimation of distribution algorithm [45][46]. In the classification-based surrogate-assisted evolutionary algorithm (CSEA) [47], solutions in the population were classified into two categories by an FNN. Moreover, the classification surrogate-assisted MOEA (CSA-MOEA) [48] partitioned the population into several clusters by reference vectors, and only solutions with good performance in each cluster were labeled as '1'. Offspring are allowed to be preserved only when labeled as '1' using a classification tree. In the multiple classifiers-assisted MOEA with decomposition (MCEA/D) [49], different SVMs were trained as classifiers for corresponding scalarization functions of each subproblem. In [50], two fuzzy classifiers were trained as the classification surrogate to classify all solutions as non-dominated and dominated. Then these fuzzy classifiers can guide the pre-selection and new offspring generation during the optimization process. In [51], the generative adversarial network (GAN) was adopted as the surrogate to classification solutions as real samples and fake samples, respectively. Then the generator in the GAN can generate potentially excellent new offspring solutions, and the discriminator in the GAN can help to provide the predicted labels for these solutions.

In *relation classification*, all the evaluated solutions are paired and assigned a label to indicate the relation between them. Then a classifier is trained to approximate the binary relation between any two candidate solutions. Compared with the *rank classification*, the *relation classification* has a larger training dataset under the same number of solutions, which is helpful in training a finer surrogate. For example, a DB containing N solutions can form $N \times (N-1)$ solution pairs at most. Therefore, *relation classification* is suitable for EMOPs in which only a small number of real evaluated solutions can be collected. In [52], the pair relation in each objective was approximated by a binary relation classifier. The fusion of all classifiers can approximate the dominance relation between any two candidate solutions. In [53], the pattern recognition method was used to train a Pareto dominance classifier. In the dominance prediction for θ dominance-based evolutionary algorithm (θ-DEA-DP) [54] and the relation model-assisted MOEA (REMO) [55], the FNN was used to approximate the binary relation of solution pairs. Specifically, two different FNNs were used to approximate the θ-dominance and Pareto-dominance relations in θ-DEA-DP. As for REMO, only an FNN was used, while another voting-scoring strategy was adopted for selection.

Fig. 1 illustrates an example of these two classification ways. Through *rank classification* in Fig. 1(a), solutions are

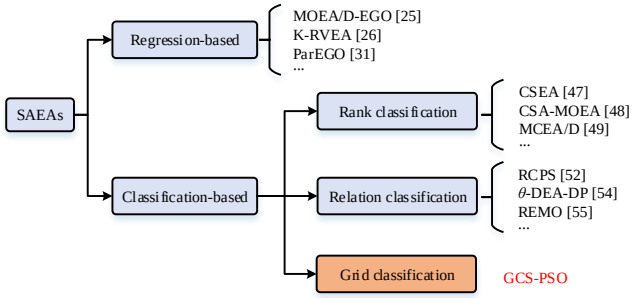


Fig. 2. The taxonomy of existing SAEAs and the novelty of GCS-PSO.

divided into two categories via non-dominated sorting [5], with a label of *good* or *bad*. Through *relation classification* in Fig. 1(b), Pareto dominance relations (denoted as <**x**, **y**>) between any two solutions are recorded, in which '1', '-1', or '0' means **x** dominates, is dominated by, or non-dominated with **y**, respectively.

However, most existing classification surrogates tend to evaluate solutions from the convergence aspect but rarely from the diversity aspect. This makes both the *rank classification*-based and the *relation classification*-based SAEAs require additional strategies to maintain diversity. For example, CSA-MOEA [48] used the promoting method, while θ-DEA-DP [54] used predefined uniform reference vectors to maintain diversity. Moreover, as the diversity information is difficult to be considered in existing classification-based SAEAs, most of them still use complex network models as classifiers. For example, FNN was used as a classifier in CSEA [47] and θ-DEA-DP [54], and GAN was used in [51]. These network models can still be regarded as regression models except that the final output units are processed to express as a binary relation. Compared with classical regression models, training such a network model may even be more time-consuming. This makes these classification-based SAEAs have no significant advantage over the regression-based SAEAs in terms of training costs.

Considering that the grid-based technique has been widely adopted in the multiobjective optimization community to preserve both convergence and diversity information [56]-[58], we try to adopt it for expensive multiobjective optimization and propose a new classification method called *grid classification*. Specifically, the *grid classification* classifies the solutions into different grids in the objective space.

An example of *grid classification* is illustrated in Fig. 1(c). The objective space is discretized into uniform grids so as to locate the grid positions of solutions in DB based on their objective values. These solutions with grid location information are used to train a *grid classification*-based surrogate (GCS) for predicting the grid locations of new candidate solutions. As the grid information reflects both the convergence and diversity information of solutions in the objective space, the benefits of GCS mainly lie in two aspects. First, GCS is a lightweight surrogate and requires a quite shorter training time than most existing SAEAs. Second, different from the *rank classification*-based and *relation classification*-based surrogates, GCS evaluates the performance of solutions in terms of both convergence and diversity, without any additional diversity maintenance strategy. As illustrated in Fig. 2, the *grid classification* opens up a new field in SAEAs for follow-up research.

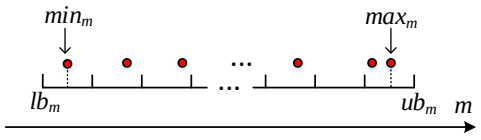


Fig. 3. Grid division in the mth objective space.

The main contributions can be summarized as follows:

1) The proposed GCS is based on grid classification and thus it does not need a large amount of data for training a refined regression surrogate, which means it can be built by lightweight classification models with lower training costs. Therefore, it is easy to be applied to solve high-dimensional EMOPs.

2) By using a particle swarm optimization (PSO) variant as the base optimizer, we propose a GCS-assisted PSO (GCS-PSO) algorithm. Compared with existing SAEAs, the GCS-PSO can use a lightweight GCS to achieve good performance in terms of both convergence and diversity when dealing with EMOPs. The GCS-PSO can even tackle high-dimensional EMOPs without large computational costs.

3) To the best of our knowledge, this is the first attempt to adopt *grid classification* in classification-based SAEAs for solving EMOPs. This opens a new surrogate building method in SAEAs that can achieve both convergence and diversity maintenance with lightweight classification models.

The rest of the paper is organized as follows. Section II introduces some related works about existing SAEAs and some background information. Section III describes our proposed GCS-PSO algorithm in detail. Section IV gives the experimental results as well as some effectiveness and sensitivity analysis. Finally, the conclusion is given in Section V.

II. Preliminary

A. MOP

Real-world problems usually involve multiple conflicting objectives that need to be considered simultaneously, and such complex problems are called MOPs. An MOP with M objectives to be minimized can be described as:

$$\begin{aligned} &\min F(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\right)^{\mathrm{T}}, \\ &s.t.\ \mathbf{x} \in \Omega, \end{aligned} \tag{1}$$

where Ω is the decision space, and $\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$ is a candidate solution. $F: \Omega \rightarrow R^M$ represents M objective functions, and R^M denotes the objective space. Given two solutions $\mathbf{x}, \mathbf{y} \in \Omega$, $\mathbf{x}$ is said to Pareto dominate $\mathbf{y}$ iff $\forall\, i \in \{1,\ldots, M\}$ $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\exists\, j \in \{1,\ldots, M\}$ $f_j(\mathbf{x}) < f_j(\mathbf{y})$. $\mathbf{x}$ is said to be a Pareto optimal solution if there does not exist another $\mathbf{x}^* \in \Omega$ such that $\mathbf{x}^*$ Pareto dominates $\mathbf{x}$. The aim of addressing an MOP is to search for the set of all Pareto optimal solutions, which is called the Pareto set. The corresponding objective vector set of the Pareto set is called the PF.

B. Grid Definition and Grid Dominance

It is well-known that the exact fitness values are unnecessary during selection if the binary relation between solutions

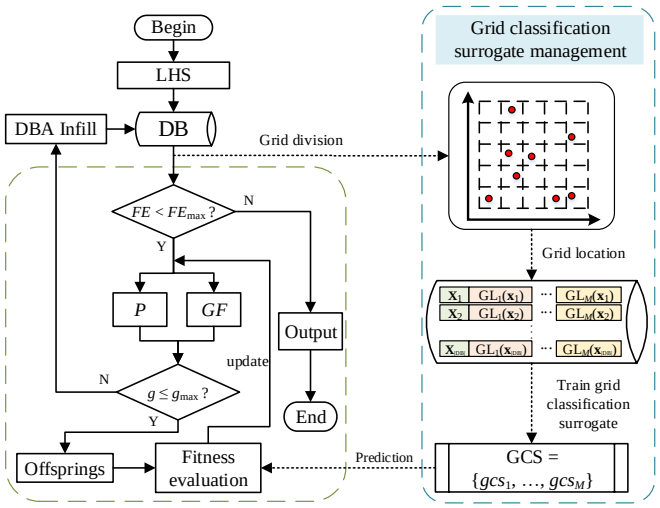


Fig. 4. The diagram of GCS-PSO.

can be found. Coincidentally, grid location is one of the ways that can reflect this relation of solutions. In the grid-based evolutionary algorithm [56], each solution is assigned at a hyper-grid by its objective values. Its convergence and diversity performance can be evaluated through this grid location. Fig. 3 illustrates the grid division in the mth objective, where $m \in \{1, 2, \ldots, M\}$. By given a parameter div, each objective can be divided into div grids, and the lower and upper boundaries of the gird in the mth objective (lb_m and ub_m) can be calculated by:

$$lb_m = min_m - \left(max_m - min_m\right) / (2 \times div) \tag{2}$$

$$ub_m = min_m + \left(max_m - min_m\right) / (2 \times div) \tag{3}$$

where min_m and max_m are the minimum and maximum values regarding the mth objective among all solutions in the population. Then the width of each grid in the mth objective (d_m) can be defined as:

$$d_m = \left(ub_m - lb_m\right) / div \tag{4}$$

Through the above grid division, the objective space can be divided into div^M hyper-grids, and the grid location (GL) of solution $\mathbf{x}$ can be determined by:

$$\mathrm{GL}_m(\mathbf{x}) = \left\lfloor \left(f_m(\mathbf{x}) - lb_m\right) / d_m \right\rfloor \tag{5}$$

where $\lfloor \cdot \rfloor$ represents the floor function. In [56], grid dominance was proposed, as:

Grid Dominance: In a population P, solution $\mathbf{x} \in P$ is said to grid-dominate another solution $\mathbf{y} \in P$ iff $\forall\, i \in \{1, \ldots, M\}$ $\mathrm{GL}_i(\mathbf{x}) \le \mathrm{GL}_i(\mathbf{y})$ and $\exists\, j \in \{1, \ldots, M\}$ $\mathrm{GL}_j(\mathbf{x}) < \mathrm{GL}_j(\mathbf{y})$.

Through grid dominance, solutions in the population can be sorted in a strict partial order. Moreover, unlike Pareto dominance, grid dominance does not require exact fitness values but only discrete grid locations.

C. Social Learning PSO

Social learning PSO (SLPSO) [59] is a PSO [60][61][62] variant that has shown good performance for large-scale single-objective optimization with high-dimension. As this paper also focuses on high-dimensional EMOPs, the SLPSO is adopted as the base optimizer. The main idea of SLPSO is to introduce a social learning mechanism into PSO to help the particle update. In the canonical SLPSO, all the particles in the current population are sorted based on their fitness values in descending order. Each particle dimension is then updated based on one better particle and the mean decision vector of the whole population. For a particle $\mathbf{x}$ at the gth generation in SLPSO, an offspring particle $\mathbf{y}$ is generated as:

$$y_d = \begin{cases} x_d, & \text{if } rand() > Pr \\ x_d + v_d, & \text{otherwise} \end{cases} \tag{6}$$

$$v_d = r_1 \cdot (z_d^k - x_d) + \varepsilon \cdot r_2 \cdot (\bar{x}_d - x_d) + r_3 \cdot v_d \tag{7}$$

where $d \in \{1, \ldots, D\}$ is the dimension of decision variables; Pr is the learning probability; $\mathbf{v} = (v_1, \ldots, v_D)$ is the velocity vector; r_1, r_2, and r_3 are three random numbers between 0 and 1; z_d^k is the dth dimension decision variable of selected better particle, k is the index of better particle and is reselected for each dimension; ε is a parameter called social influence factor; $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_D)$ is the mean decision vector of all solutions at the gth generation. When all offspring particles have been generated, each particle $\mathbf{x}$ will be updated by its corresponding offspring particle $\mathbf{y}$ to the next (g + 1)th generation.

III. The Proposed GCS-PSO Algorithm

A. General Framework

Fig. 4 shows the diagram of GCS-PSO, which includes two parts: surrogate-assisted optimization (left green dotted box) and grid-based classification surrogate management (right blue dotted box). The surrogate-assisted optimization aims to search for more promising solutions, while the GCS management is responsible for when and how GCS is used to replace the real FE. After each loop of the optimization process, a DB-assisted (DBA) infill criterion is adopted to select potential promising solutions from the current population for real FE. These real evaluated solutions are then moved into the DB and help update the GCS.

The pseudo-code of the proposed GCS-PSO is presented in **Algorithm 1**. Firstly, the Latin hypercube sampling (LHS) [63] method is adopted to generate $11 \times D - 1$ solutions (Line 1). These solutions are evaluated by real FE and regarded as the initial DB. After that, the algorithm goes into a loop until the maximum number of FEs ($FE_{\max}$) is reached (Lines 2-12).

In each loop, all solutions in the DB are first assigned into different hyper-grids via the method introduced in Section II-B, and their grid locations are calculated (Line 4). A GCS is built and trained according to these solutions and their corresponding grid locations (Line 5). Moreover, an archive is used to preserve non-grid-dominated solutions in the DB since these solutions could be optimal. In this paper, we call the set of these non-grid-dominated solutions as the grid front (GF) (Line 6). Then N solutions are selected from the DB as the initial population for surrogate-assisted SLPSO (Line 7). After the surrogate-assisted optimization process (Line 8), μ solutions are selected from the final population via the DBA infill criterion (Line 9). These solutions are added to the DB after real FE (Line 10). At the end of the algorithm, all Pareto non-dominated solutions in the DB are output.

Algorithm 1 The framework of GCS-PSO

Input: N (population size), D (dimensions of decision space), div (number of grid divisions), $FE_{\max}$ (maximum number of FEs), μ (number of solutions selected for real FE), $g_{\max}$ (number of generations for surrogate-assisted optimization);
Begin
1: DB = generate solutions by LHS method; // DB initialization
2: FE = |DB|;
3: **While** $FE < FE_{\max}$ **Do**
4: GL(DB) = calculate the grid location (div, DB); //Sec. II-B
5: GCS = train grid classification surrogate via {DB, GL(DB)}; //Sec. III-B
6: GF = non-grid-dominated solutions in DB;
7: $P = N$ solutions selected from DB;
8: P = Surrogate-assisted_SLPSO (P, GF, GCS, $g_{\max}$); //**Algorithm 2**
9: S = DBA_infill (DB, P, μ); //**Algorithm 5**
10: Add S into DB;
11: $FE = FE + |S|$;
12: **End While**
Output: Pareto non-dominated solutions in DB.
End

Algorithm 2 Surrogate-assisted_SLPSO

Input: P, GF, GCS, $g_{\max}$;
Begin
1: $g = 0$;
2: **While** $g \le g_{\max}$ **Do**
3: $\bar{\mathbf{x}}$ = mean decision vector of all solutions in P;
4: **For Each** $\mathbf{x} \in P$ **Do**
5: $\mathbf{y}$ = generate offspring via SLPSO($\mathbf{x}, \bar{\mathbf{x}}$, GF); // Eq. (6) and Eq. (10)
6: GL($\mathbf{y}$) = predicted by GCS via Eq. (9);
7: Calculate GR($\mathbf{x}$) and GR($\mathbf{y}$) via Eq. (11);
8: NCI($\mathbf{x}$) = NCI_calculation (GL($\mathbf{x}$), P); //**Algorithm 3**
9: Temporarily replace $\mathbf{x}$ with $\mathbf{y}$ in P;
10: NCI($\mathbf{y}$) = NCI_calculation (GL($\mathbf{y}$), P); //**Algorithm 3**
11: **If** GR($\mathbf{x}$) > GR($\mathbf{y}$) or (GR($\mathbf{x}$) = GR($\mathbf{y}$) & NCI($\mathbf{x}$) > NCI($\mathbf{y}$)) **Then**
12: Update $\mathbf{x}$ by $\mathbf{y}$;
13: **End If**
14: **End For**
15: GF = Grid_front_update (P, GF); //**Algorithm 4**
16: $g = g + 1$;
17: **End While**
Output: P.
End

In the following, the details of the GCS management, the surrogate-assisted SLPSO, and the DBA infill criterion are presented.

B. GCS Management

The solutions collected in the DB initially contain only their objective values. In GCS management, we first calculate their grid locations. Then we use these solutions and their corresponding grid locations to train a GCS. For an M-objective EMOP, the GCS consists of M classification models:

$$\text{GCS} = \{gcs_1, \ldots, gcs_M\} \tag{8}$$

in which the mth model gcs_m approximates the grid location in the mth objective.

Without loss of generality, gcs_m for the mth objective is taken as an example to describe the process of training. After grid division, the mth coordinate axis is converted into div discrete grids. Solutions are then assigned into d_m ($0 < d_m \le div$) out of the div grids based on their values on f_m. Note that only d_m grids instead of div grids are utilized since some grids may not be assigned with any solution. A more detailed analysis of the parameter div can be seen in Section IV-D-4).

Then each utilized grid can be treated as a class consisting of solutions that have been assigned to it. Those solutions in it are labeled by its grid index, i.e., GL_m. Thus, the training data {DB, GL_m(DB)} is actually a classification dataset with d_m classes for training a multi-class classification model gcs_m. The d_m is an inherent parameter of the dataset and does not require manual involvement.

After training, GCS is used to predict the grid locations of new candidate solutions without real FE. For a candidate solution $\mathbf{x}$, its grid location is predicted as:

$$\text{GL}(\mathbf{x}) = \text{GCS}(\mathbf{x}) \tag{9}$$

In detail, when a solution $\mathbf{x}$ to be predicted is fed to the GCS, gcs_m in GCS will provide a class $\text{GL}_m(\mathbf{x})$ on the mth objective, which is the index of grid $\mathbf{x}$ likely to be. The predicted classes from all gcs_m together make up the predicted grid location for $\mathbf{x}$.

C. Surrogate-assisted SLPSO

GCS-PSO adopts an SLPSO variant as the base optimizer for optimization. The pseudo-code of surrogate-assisted SLPSO is presented in **Algorithm 2**.

In the canonical SLPSO, each solution (i.e., particle) is updated based on its better solutions and the mean position vector of the whole population. In this paper, we consider the solutions in GF as better solutions since they are non-grid-dominated. For a solution $\mathbf{x} = (x, \ldots, x_D)$ at gth generation, an offspring $\mathbf{y} = (y_1, \ldots, y_D)$ can be generated by Eq. (6). It is worth noting that the velocity vector update (i.e., Eq. (7)) at every generation is affected by its previous generation in canonical SLPSO. However, Yang et al. [64] pointed out that such a velocity vector update method is harmful since the prediction error may remain in these historical records. Therefore, a new generation-independent velocity vector update method is adopted in GCS-PSO, as:

$$v_d = r_1 \cdot (z_d^k - x_d) + r_2 \cdot (\bar{x}_d - x_d) \tag{10}$$

where z_d^k is the dth dimension decision variable of the better solution randomly selected from GF, $k \in \{1, \ldots, |GF|\}$ is the index of the selected solution and is reselected for each dimension.

For each offspring $\mathbf{y}$, its grid location is predicted by the GCS. Then we decide whether to update $\mathbf{x}$ with $\mathbf{y}$ based on their grid locations. The update decision refers to the two basic criteria of multiobjective optimization: convergence and diversity. Two indicators, namely grid ranking (GR) and neighbors crowding index (NCI), are adopted to determine the performance relation between $\mathbf{x}$ and $\mathbf{y}$ in terms of convergence and diversity.

GR: GR is proposed in [56] to estimate the convergence of solutions by their grid locations. For a solution $\mathbf{x}$, its GR value can be calculated as:

$$\text{GR}(\mathbf{x}) = \sum_{m=1}^{M} \text{GL}_m(\mathbf{x}) \tag{11}$$

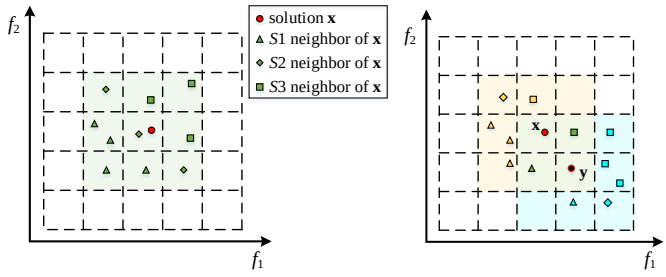
A smaller GR value means a better convergence performance. During the update, GR works as the primary indicator for maintaining convergence.

Algorithm 3 NCI_calculation

Input: GL(**x**), *P*;
Begin
1: Find neighbors of **x** from *P* via Eq. (12);
2: *S*1, *S*2, and *S*3 = neighbor division through grid dominance;
3: NCI(**x**) calculation via Eq. (13).
Output: NCI(**x**).
End

Algorithm 4 Grid_front_update

Input: *P*, *GF*;
Begin
1: **For Each** **x** ∈ *P* **Do**
2: *Arc* = solutions in *GF* that are grid-dominated by **x**;
3: *GF* = *GF* \ *Arc*;
4: **If** ∀ ***s*** ∈ *GF* are non-grid-dominated with **x** **Then**
5: *GF* = *GF* ∪ {**x**};
6: **End If**
7: **End For**
Output: updated *GF*.
End



(a) (b)

Fig. 5. Example of neighbors in a 2-objective space.

NCI: NCI estimates the convergence and diversity of a solution **x** through the grid location relation between **x** and its neighbors. Before introducing NCI, we explain the concept of neighbors.

For solution **x**, we use the grid difference (GridD) to quantify its crowding relation with any other solution **s**. The GridD between **x** and **s** can be calculated by:

$$\mathrm{GridD}(\mathbf{x},\mathbf{s}) = \sum_{m=1}^{M} |\,\mathrm{GL}_m(\mathbf{x}) - \mathrm{GL}_m(\mathbf{s})\,| \tag{12}$$

Any solution **s** in *P* is regarded as the neighbor of **x** iff GridD(**x**, **s**) ≤ *M*. The more neighbors a solution has, the worse its diversity is. Moreover, these neighbors can be further divided into three different types:

*S*1: neighbors that grid-dominates **x**;
*S*2: neighbors that are grid-dominated by **x**;
*S*3: neighbors that are non-grid-dominated with **x**.

Fig. 5(a) gives an example in a 2-objective space, where the grid region with a light green background is the neighbor region of **x**; the triangles, rhombus, and quadrates are the *S*1, *S*2, and *S*3 neighbors of **x,** respectively.

These different types of neighbors have different qualitative effects on **x**. Here we use an example in a 2-objective space to illustrate it, as shown in Fig. 5(b). Considering two solutions **x** and **y**, the triangles, rhombus, and quadrates filled with yellow are *S*1, *S*2, and *S*3 neighbors of **x**, respectively; those filled with blue are *S*1, *S*2, and *S*3 neighbors of **y**, respectively; and the triangles and quadrates filled with green are *S*1 and *S*3 neighbors of both **x** and **y**. Moreover, **x** and **y** are *S*2 neighbors mutually. As shown in Fig. 5(b), **x** and **y** have the same number of neighbors, but **x** has more *S*1 neighbors, and **y** has more *S*3 neighbors. More *S*1 neighbors mean worse convergence. Conversely, more *S*3 neighbors indicate better convergence in the local region. Therefore, the NCI of **x** herein is defined as:

$$\begin{aligned}\mathrm{NCI}(\mathbf{x}) = {} & \frac{|S1|+1}{|S3|+1}\cdot\sum_{\mathbf{s}\in S1}(M+1-\mathrm{GridD}(\mathbf{x},\mathbf{s})) \\ & +\sum_{\mathbf{s}\in S2}(M+1-\mathrm{GridD}(\mathbf{x},\mathbf{s})) \\ & +\frac{|S3|+1}{|S1|+1}\cdot\sum_{\mathbf{s}\in S3}(M+1-\mathrm{GridD}(\mathbf{x},\mathbf{s}))\end{aligned} \tag{13}$$

A smaller NCI value represents better convergence and diversity. **Algorithm 3** gives the pseudo-code of the NCI calculation process. As an assistant of GR, NCI can only come into play when GR cannot determine the relation between solutions. Therefore, NCI works as an enhancement indicator in GCS-PSO.

Based on the two indicators mentioned above, **x** would be updated by **y** if:

(1) GR(**x**) > GR(**y**), or
(2) GR(**x**) = GR(**y**) and NCI(**x**) > NCI(**y**).

At the end of each generation, the *GF* needs to be updated to ensure that the solutions in it are grid-optimal. The detail of the grid front update is given in **Algorithm 4**. For each solution **x** in the current population, we first remove all the solutions that are grid-dominated by **x** from *GF*. Then **x** is added into *GF* once it is non-grid-dominated with all solutions in *GF*.

Algorithm 5 DBA_infill

Input: DB, *P*, μ (number of solutions selected for real FE);
Begin
1: Remove solutions that have been evaluated by real FE from *P*;
2: **If** $|P| \le \mu$ **Then**
3: Real FE for solutions in *P*;
4: **Else**
5: $\{\mathbf{r}_1, \ldots, \mathbf{r}_\mu\}$ = select μ reference points from DB; /** Step 1 **/
/** Step 2 **/
6: **For Each** **x** ∈ *P* **Do**
7: $k = \arg\min_{i=1,\ldots,\mu} dis(\mathbf{x}, \mathbf{r}_i)$;
8: Move **x** into cluster *k*;
9: **End For**
/** Step3 **/
10: **For Each** cluster *i* ∈ {1, …, μ} **Do**
11: **If** cluster *i* has assigned solutions **Then**
12: **x** = solution with smallest GR value in cluster *i*;
13: **Else**
14: **x** = arg min *dis*(**x**, $\mathbf{r}_i$), **x** ∈ neighbor cluster;
15: **End If**
16: Real FE for **x**;
17: **End For**
18: **End If**
Output: real evaluated solutions.
End

D. DBA Infill Criterion

Limited by $FE_{\max}$, only a few potential promising solutions with good convergence and diversity can be selected for real FE via an infill criterion. Considering that the DB holds the optimal

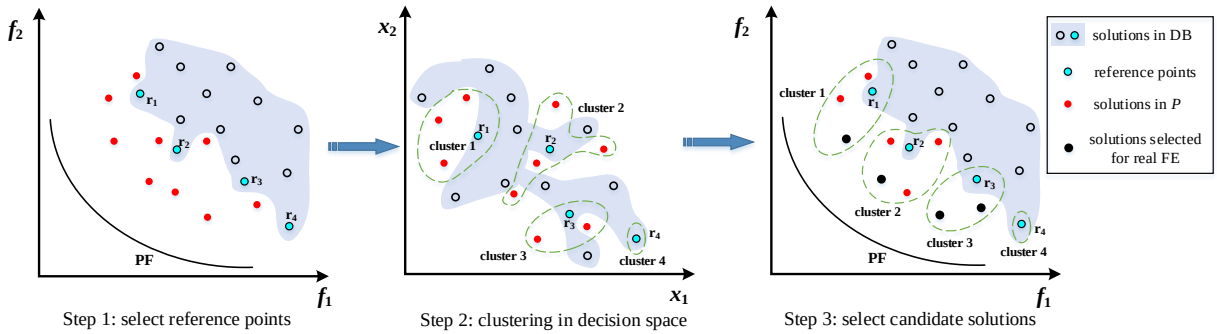


Fig. 6. Illustration of DBA infill criterion in a 2-objective space with 2 decision variables.

solutions so far, they can serve as guides for the infill criterion. Herein, we propose the DBA infill criterion to select μ candidate solutions for real FE and add them into the DB to support the GCS update.

In the DBA infill criterion, the solutions that have been evaluated by real FE are removed from P first. Then all remaining solutions are added to the DB after FE once the number is less than μ. Otherwise, μ candidate solutions will be selected from P in the following three steps, as shown in Fig. 6.

Step 1: μ reference points ($\mathbf{r}_1$, …, $\mathbf{r}_\mu$) are selected from the DB by considering their convergence and diversity to serve as the centroids of every cluster next. The environmental selection method in NSGA-II [5] is adopted to select these reference points herein, i.e., μ solutions with the largest crowding distances at the non-dominated front.

Step 2: Every solution in P finds the nearest reference point in the decision space and assigns itself to the cluster where the nearest reference point is. After that, P will be classified into μ clusters.

Step 3: For each cluster $i \in \{1, \ldots, \mu\}$, a candidate solution with good convergence will be selected for real evaluation. Herein, the solution with the smallest GR value will be selected (e.g., clusters 1, 2, and 3 in Fig. 6). However, there may be a situation in which no solutions (except for the reference point) were assigned to a cluster. In this situation, the selection will be performed in its non-empty neighbor cluster, in which the solution nearest to $\mathbf{r}_i$ is selected (e.g., cluster 4 in Fig. 6). The pseudo-code of our proposed infill selection method is given in **Algorithm 5**.

IV. Experimental Studies

A. Parameter Settings

The experiments adopt four different classification models for building GCS in the GCS-PSO. These models include the KNN, naïve Bayes classifier (NBC) [65], C-support vector classification (CSVC) [66], and random forests (RF) [67].

KNN: If most of the K nearest neighbors of a candidate solution in the feature space (decision space herein) belong to a class, then this candidate solution also belongs to that class and has the properties of neighbors in that class. KNN is a straightforward classification model since it only concerns a tiny number of adjacent samples.

NBC: NBC is a simple probabilistic classifier based on the application of Bayes' theorem under the assumption of *conditional independence* between features. For a given candidate solution, NBC calculates the probability that it belongs to each class conditional on the occurrence of the candidate solution. Then, the candidate solution is considered to belong to the class with the largest probability.

TABLE I
Parameters of Classification Models Used in the Experiments

Model name	Parameters
KNN	$K = 5$
NBC	distribution = 'Gaussian kernel smoothing density'
CSVC	kernel = 'polynomial'
RF	features = $0.8\times D$, samples = $0.8\times$|DB|, trees = 5

CSVC: CSVC is a kind of SVM for classification in which the penalty parameter c can adjust the generalization ability of the classifier. The larger c is, the greater the punishment for misclassification is, and the smaller deviation but larger variance the obtained classifier has. Herein, the default value $c = 1$ is used.

RF: RF is an ensemble classifier that consists of multiple classification trees. Note that the training samples of each tree are only part of the whole *DB*, and the unordered sampling method with replacement is adopted in this paper. At each node, the features (i.e., decision variables) used in the split criterion are selected from a random feature subset rather than the complete feature set. For a candidate solution, the output by an RF is determined by the mode of the output from all classification trees.

Some specific parameters about them are shown in Table I. Apart from that, GCS-PSO is compared with seven state-of-the-art SAEAs, including three regression-based SAEAs (K-RVEA [26], HeE-MOEA [35], and EDN-ARMOEA [40]) and four classification-based SAEAs (CSA-MOEA [48], CSEA [47], MCEA/D [49], and REMO [55]). Each algorithm is conducted on three-objective DTLZ1-DTLZ5 [68] and WFG1-WFG9 [69] benchmark problems with different dimensions of decision variables D = 20, 40, 60, and 80. When the dimension of decision variables is larger than 50, we can regard the EMOP as high-dimensional. For each test instance, the LHS is used to randomly generate $11\times D - 1$ real evaluated solutions as initialized *DB* to simulate the data collection process. Note that the data are collected offline and we assume it is available in real-world applications. Nevertheless, even if the collected data is insufficient, the localized data generation method to generate synthetic data can

be used to alleviate data shortage [70]. Moreover, compared with the regression-based surrogates, the training of our GCS does not require a large amount of data. This is also one of the advantages of our proposed algorithm. The maximum number of FEs ($FE_{\max}$) is set to $(11 \times D - 1) + 120$. Thus, the amount of FEs that can be used in the optimization process is limited to the same as 120 for each algorithm to ensure the fairness of the comparison.

For each algorithm, the population size N is set to 50. In K-RVEA, HeE-MOEA, EDN-ARMOEA, and GCS-PSO, the number of generations ($g_{\max}$) for surrogate-assisted optimization before the infill criterion is set as 20; and the number of selected solutions (μ) by infill criterion is set to 5. The number of grid divisions (div) in GCS-PSO is set to 30. As for the SLPSO used in GCS-PSO, the learning probability (Pr) in it is set to 0.8. Except for these, other parameters of all compared algorithms are set according to their original papers for a fair comparison. All algorithms are implemented in Matlab and run 20 times independently on an identical computer with Inter(R) Core (TM) i5-11400 2.60 GHz and 16GB of RAM. The generalized distance (GD) [71] is used to evaluate the performance of the obtained solution set in convergence, and the inverted GD (IGD) [72] is adopted to evaluate the performance both in convergence and diversity. Note that G in GD and IGD stands for "generalized" may be more suitable than "generational" because it means the distance between two sets of solutions rather than the "direct" distance between two points [73]. Wilcoxon's rank-sum test at a 5% significance level is conducted for the significance test. The symbols of '+', '–', and '=' represent that the algorithm in the first column is significantly better than, worse than, or similar to the corresponding compared SAEA, respectively.

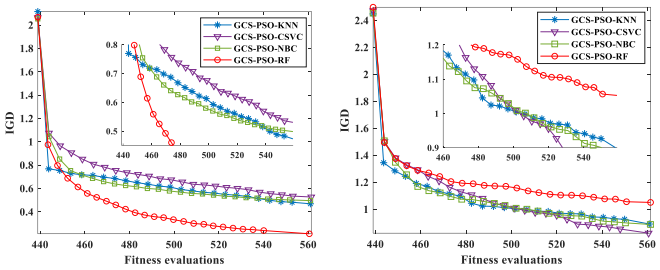
B. Experiments on Different Classification Models

To investigate the impact of different classification models on the performance of GCS-PSO, we design four GCS-PSO algorithms with different GCSs denoted as GCS-PSO-KNN, GCS-PSO-CSVC, GCS-PSO-NBC, and GCS-PSO-RF. The statistical results in terms of IGD and GD on DTLZ and WFG problems are shown in Table S.I and S.II in the supplementary material, where the best result on each instance is marked in **boldface**. Table II gives the summary results, in which the significance test is conducted between GCS-PSO-RF and the other three GCS-PSO algorithms.

Through the results in Table II, it can be seen that the performance of GCS-PSO is greatly affected by the type of classification model used in GCS. RF is more suitable for GCS-PSO than the other three models since GCS-PSO-RF outperforms other GCS-PSO algorithms on most problems. This is because, along with the optimization process, the new real-evaluated solutions that are added to the DB will gradually be closer to the true PF. After grid division, the corresponding data in the DB will become more unbalanced since more data will belong to classes with small GL indexes. Fortunately, RF is more suitable to deal with this kind of unbalanced data, and the GCS built by RF can be more refined than those built by the other three classification models. However, this does not necessarily mean that RF is the best choice as GCS in GCS-PSO. For example, GCS-PSO-RF performs very badly on DTLZ1 and DTLZ4 problems compared with the other three GCS-PSO algorithms. This result indicates that the performance of a GCS depends on the characteristics of the optimized problem. Certainly, except for RF, the other three models perform roughly the same. To compare the different models in more detail, the mean IGD curves of the four GCS-PSO algorithms on DTL2 and DTLZ4 (D = 40) are plotted in Fig. 7. We can also see that the performance of the three models except RF in GCS-PSO is similar, and is significantly different from RF including better or worse. The overall results in Table II show that RF is more suitable for GCS since GCS-PSO-RF is superior to the other three GCS-PSO algorithms in most instances. In subsequent experiments, RF is used for building GCS in GCS-PSO unless otherwise specified.

TABLE II
SIGNIFICANCE TEST BETWEEN GCS-PSO-RF AND OTHER GCS-PSO ALGORITHMS, SUMMARIZED AS '+/-/='

Problem	Metric	GCS-PSO-KNN	GCS-PSO-CSVC	GCS-PSO-NBC
DTLZ	IGD	8/6/6	8/8/4	8/3/9
	GD	8/9/3	8/7/5	8/5/7
WFG	IGD	23/1/12	22/2/12	11/3/22
	GD	16/9/11	17/9/10	9/7/20
Total	IGD	31/7/18	30/10/16	19/6/31
	GD	24/18/14	25/16/15	17/12/27



(a) DTLZ2 (b) DTLZ4

Fig. 7. The mean IGD curves of GCS-PSO with different GCSs on DTLZ2 and DTLZ4 (D = 40).

C. Comparison with Other SAEAs

1) Statistical Results Analysis

In this subsection, we conduct a comparison experiment between GCS-PSO-RF and seven other SAEAs to verify the performance of GCS-PSO in dealing with EMOPs. At the same time, we also replace the base optimizer in GCS-PSO-RF with differential evolution (DE) [74]-[77] to verify the universality of our algorithm framework, and the GCS-assisted algorithm using DE/current-to-best/1 mutation operator is called GCS-DE-RF. Notice that K-RVEA did not experiment on problems with D = 80 due to the long running time. Table III presents the statistical results in terms of IGD values on DTLZ and WFG problems, where the best result on each instance is marked in **boldface**. Note that there are two symbols after each IGD result, where the first and second denote the significance test versus GCS-PSO-RF and GCS-DE-RF, respectively.

It can be seen that the two GCS-assisted algorithms are better than others on most problems. GCS-PSO-RF outperforms K-RVEA on 30 out of 42 instances, and outperforms

TABLE III
MEAN AND STANDARD DEVIATION IGD VALUES OBTAINED BY NINE SAEAS ON DTLZ AND WFG PROBLEMS

Problem	*D*	GCS-PSO-RF	GCS-DE-RF	K-RVEA	HeE-MOEA	EDN-ARMOEA	CSA-MOEA	CSEA	MCEA/D	REMO
DTLZ1	20	3.601e+2 (4.57e+1)	3.597e+2 (3.29e+1) =	3.174e+2 (4.82e+1) - -	3.701e+2 (3.49e+1) = =	3.538e+2 (3.79e+1) = =	3.119e+2 (3.87e+1) - -	2.836e+2 (3.43e+1) - -	2.341e+2 (5.36e+1) - -	**1.989e+2 (4.02e+1) - -**
	40	9.045e+2 (4.11e+1)	8.828e+2 (5.46e+1) =	8.361e+2 (5.60e+1) - -	9.139e+2 (6.27e+1) = =	8.858e+2 (4.75e+1) = =	8.394e+2 (7.61e+1) - -	7.907e+2 (6.64e+1) - -	7.067e+2 (1.10e+2) - -	**4.305e+2 (5.06e+1) - -**
	60	1.430e+3 (7.70e+1)	1.436e+3 (7.72e+1) =	1.368e+3 (7.47e+1) - -	1.451e+3 (7.14e+1) = =	1.458e+3 (6.45e+1) = =	1.436e+3 (5.98e+1) = =	1.275e+3 (7.09e+1) - -	**1.114e+3 (1.36e+2) - -**	1.260e+3 (9.63e+1) - -
	80	2.000e+3 (9.79e+1)	2.005e+3 (8.04e+1) =	Na	2.002e+3 (8.52e+1) = =	1.990e+3 (6.61e+1) = =	1.990e+3 (7.77e+1) = =	1.807e+3 (1.34e+2) - -	**1.695e+3 (1.95e+2) - -**	1.749e+3 (1.62e+2) - -
DTLZ2	20	**1.422e-1 (1.98e-2)**	1.996e-1 (4.00e-2) +	7.127e-1 (6.64e-2) + +	3.507e-1 (3.09e-2) + +	6.335e-1 (8.37e-2) + +	6.173e-1 (8.82e-2) + +	6.313e-1 (5.74e-2) + +	4.330e-1 (3.89e-2) + +	4.189e-1 (8.03e-2) + +
	40	**2.128e-1 (3.30e-2)**	3.533e-1 (6.65e-2) +	2.013e+0 (1.14e-1) + +	7.163e-1 (7.99e-2) + +	2.048e+0 (7.35e-2) + +	1.605e+0 (1.59e-1) + +	1.632e+0 (2.25e-1) + +	6.465e-1 (9.40e-2) + +	1.499e+0 (2.00e-1) + +
	60	**3.299e-1 (5.82e-2)**	4.972e-1 (7.85e-2) +	3.271e+0 (2.27e-1) + +	1.219e+0 (1.04e-1) + +	3.326e+0 (1.47e-1) + +	2.766e+0 (2.23e-1) + +	2.734e+0 (2.73e-1) + +	8.370e-1 (7.88e-2) + +	2.765e+0 (2.82e-1) + +
	80	**3.628e-1 (5.96e-2)**	6.770e-1 (1.32e-1) +	Na	1.768e+0 (1.81e-1) + +	4.703e+0 (2.32e-1) + +	4.106e+0 (2.18e-1) + +	4.076e+0 (2.54e-1) + +	9.942e-1 (1.52e-1) + +	3.792e+0 (1.95e-1) + +
DTLZ3	20	1.049e+3 (1.10e+2)	1.001e+3 (8.11e+1) =	8.777e+2 (1.03e+2) - -	1.041e+3 (1.17e+2) = =	1.071e+3 (1.02e+2) = +	8.858e+2 (1.49e+2) - -	7.991e+2 (1.21e+2) - -	**5.263e+2 (9.33e+1) - -**	5.374e+2 (9.40e+1) - -
	40	2.695e+3 (1.34e+2)	2.685e+3 (1.68e+2) =	2.543e+3 (2.10e+2) - -	2.693e+3 (1.71e+2) = =	2.832e+3 (1.39e+2) + +	2.502e+3 (2.17e+2) - -	2.269e+3 (1.65e+2) - -	1.604e+3 (3.81e+2) - -	**1.305e+3 (1.48e+2) - -**
	60	4.441e+3 (1.97e+2)	4.493e+3 (2.24e+2) =	4.180e+3 (2.99e+2) - -	4.566e+3 (1.54e+2) + =	4.496e+3 (2.18e+2) = =	4.434e+3 (2.27e+2) = =	3.999e+3 (2.52e+2) - -	**2.571e+3 (4.58e+2) - -**	3.972e+3 (2.90e+2) - -
	80	6.267e+3 (1.73e+2)	6.147e+3 (2.74e+2) =	Na	6.407e+3 (2.00e+2) + +	6.417e+3 (1.23e+2) + +	6.229e+3 (1.91e+2) = =	5.817e+3 (3.38e+2) - -	**4.173e+3 (8.56e+2) - -**	5.807e+3 (2.77e+2) - -
DTLZ4	20	8.920e-1 (1.27e-1)	9.389e-1 (8.95e-2) =	1.007e+0 (1.41e-1) + =	1.035e+0 (2.37e-2) + +	9.958e-1 (1.15e-1) + =	6.711e-1 (1.22e-1) - -	6.679e-1 (1.12e-1) - -	9.521e-1 (4.14e-2) = =	**4.611e-1 (1.05e-1) - -**
	40	1.052e+0 (1.12e-1)	1.149e+0 (6.60e-2) +	2.108e+0 (2.00e-1) + +	1.276e+0 (5.34e-2) + +	2.228e+0 (1.76e-1) + +	1.645e+0 (2.62e-1) + +	1.543e+0 (2.07e-1) + +	1.067e+0 (3.28e-2) = -	**8.134e-1 (1.42e-1) - -**
	60	1.258e+0 (1.14e-1)	1.362e+0 (1.45e-1) +	3.297e+0 (2.62e-1) + +	1.667e+0 (1.08e-1) + +	3.649e+0 (1.99e-1) + +	2.772e+0 (2.77e-1) + +	2.610e+0 (2.00e-1) + +	**1.162e+0 (5.45e-2) - -**	2.579e+0 (2.32e-1) + +
	80	1.442e+0 (1.44e-1)	1.632e+0 (1.73e-1) +	Na	2.142e+0 (1.86e-1) + +	4.923e+0 (1.82e-1) + +	4.1360e+0 (3.41e-1) + +	3.684e+0 (2.87e-1) + +	**1.311e+0 (1.34e-1) - -**	3.982e+0 (3.18e-1) + +
DTLZ5	20	**7.440e-2 (2.23e-2)**	9.912e-2 (2.72e-2) +	5.668e-1 (1.23e-1) + +	2.631e-1 (3.42e-2) + +	4.941e-1 (7.91e-2) + +	6.673e-1 (6.87e-2) + +	5.615e-1 (9.91e-2) + +	2.617e-1 (4.06e-2) + +	4.163e-1 (8.44e-2) + +
	40	**1.550e-1 (4.31e-2)**	2.500e-1 (5.32e-2) +	1.899e+0 (1.22e-1) + +	6.099e-1 (4.03e-2) + +	2.006e+0 (1.14e-1) + +	1.880e+0 (1.46e-1) + +	1.489e+0 (1.84e-1) + +	4.888e-1 (7.41e-2) + +	1.508e+0 (1.77e-1) + +
	60	**2.372e-1 (4.83e-2)**	3.723e-1 (5.97e-2) +	3.184e+0 (1.83e-1) + +	1.101e+0 (1.28e-1) + +	3.246e+0 (1.54e-1) + +	2.978e+0 (2.82e-1) + +	2.630e+0 (2.29e-1) + +	6.703e-1 (1.36e-1) + +	2.631e+0 (2.58e-1) + +
	80	**3.162e-1 (4.73e-2)**	5.539e-1 (1.31e-1) +	Na	1.640e+0 (1.57e-1) + +	4.610e+0 (2.08e-1) + +	4.371e+0 (2.57e-1) + +	3.851e+0 (2.39e-1) + +	8.262e-1 (1.22e-1) + +	3.859e+0 (2.76e-1) + +
WFG1	20	2.213e+0 (6.46e-2)	2.185e+0 (7.34e-2) =	1.848e+0 (1.57e-1) - -	2.283e+0 (4.59e-2) + +	2.086e+0 (5.44e-2) - -	2.085e+0 (1.23e-1) - -	1.840e+0 (1.33e-1) - -	2.276e+0 (4.58e-2) + +	**1.649e+0 (1.30e-1) - -**
	40	2.197e+0 (6.19e-2)	2.166e+0 (6.54e-2) =	1.790e+0 (8.65e-2) - -	2.153e+0 (6.71e-2) - =	2.070e+0 (6.22e-2) - -	2.061e+0 (1.23e-1) - -	1.886e+0 (9.39e-2) - -	2.241e+0 (2.87e-2) + +	**1.542e+0 (4.47e-2) - -**
	60	2.182e+0 (3.97e-2)	2.142e+0 (6.55e-2) =	**1.792e+0 (1.06e-1) - -**	2.112e+0 (7.05e-2) - =	2.064e+0 (6.66e-2) - -	2.052e+0 (1.30e-1) - -	1.839e+0 (1.33e-1) - -	2.204e+0 (5.95e-2) = +	1.797e+0 (6.10e-2) - -
	80	2.175e+0 (4.57e-2)	2.132e+0 (5.05e-2) =	Na	2.085e+0 (6.59e-2) - -	2.046e+0 (6.72e-2) - -	2.070e+0 (8.40e-2) - -	1.842e+0 (9.67e-2) - -	2.195e+0 (4.64e-2) = +	**1.819e+0 (1.20e-1) - -**
WFG2	20	4.640e-1 (2.78e-2)	**4.616e-1 (3.22e-2) =**	7.107e-1 (3.11e-2) + +	7.637e-1 (5.15e-2) + +	8.529e-1 (3.79e-2) + +	6.936e-1 (3.91e-2) + +	6.823e-1 (4.18e-2) + +	7.440e-1 (6.65e-2) + +	6.894e-1 (1.26e-1) + +
	40	4.795e-1 (3.09e-2)	**4.610e-1 (2.06e-2) -**	7.970e-1 (2.67e-2) + +	7.091e-1 (5.11e-2) + +	7.933e-1 (1.60e-2) + +	7.438e-1 (3.00e-2) + +	7.190e-1 (2.14e-2) + +	7.541e-1 (5.29e-2) + +	6.231e-1 (7.94e-2) + +
	60	5.132e-1 (4.74e-2)	**4.714e-1 (1.63e-2) -**	8.015e-1 (2.40e-2) + +	6.998e-1 (5.29e-2) + +	8.227e-1 (1.99e-2) + +	7.469e-1 (1.54e-2) + +	7.381e-1 (2.24e-2) + +	7.295e-1 (4.01e-2) + +	7.437e-1 (2.64e-2) + +
	80	5.239e-1 (3.79e-2)	**4.785e-1 (1.63e-2) -**	Na	7.066e-1 (3.48e-2) + +	7.997e-1 (1.30e-2) + +	7.635e-1 (1.83e-2) + +	7.376e-1 (1.50e-2) + +	7.380e-1 (3.39e-2) + +	7.495e-1 (1.69e-2) + +
WFG3	20	**3.821e-1 (4.10e-2)**	4.088e-1 (3.27e-2) +	6.935e-1 (2.42e-2) + +	4.943e-1 (2.17e-2) + +	6.911e-1 (2.64e-2) + +	6.972e-1 (2.47e-2) + +	6.764e-1 (2.91e-2) + +	5.732e-1 (4.08e-2) + +	6.157e-1 (4.03e-2) + +
	40	**4.387e-1 (3.31e-2)**	4.458e-1 (3.41e-2) =	7.541e-1 (1.32e-2) + +	5.433e-1 (1.79e-2) + +	7.578e-1 (1.29e-2) + +	7.581e-1 (1.18e-2) + +	7.361e-1 (1.89e-2) + +	6.177e-1 (2.87e-2) + +	6.191e-1 (4.85e-2) + +
	60	4.694e-1 (3.75e-2)	**4.634e-1 (3.01e-2) =**	7.766e-1 (8.25e-3) + +	5.754e-1 (1.94e-2) + +	7.757e-1 (1.14e-2) + +	7.739e-1 (1.13e-2) + +	7.607e-1 (1.70e-2) + +	6.399e-1 (3.12e-2) + +	7.612e-1 (8.24e-3) + +
	80	**4.827e-1 (2.83e-2)**	4.837e-1 (2.50e-2) =	Na	5.977e-1 (2.36e-2) + +	7.900e-1 (6.04e-3) + +	7.881e-1 (5.63e-3) + +	7.682e-1 (1.24e-2) + +	6.463e-1 (2.38e-2) + +	7.710e-1 (1.34e-2) + +
WFG4	20	4.908e-1 (5.21e-2)	**4.645e-1 (4.05e-2) =**	5.258e-1 (1.40e-2) + +	6.182e-1 (3.79e-2) + +	5.369e-1 (1.73e-2) + +	5.147e-1 (2.63e-2) = +	5.211e-1 (1.81e-2) + +	5.482e-1 (2.61e-2) + +	4.695e-1 (2.35e-2) = =
	40	4.690e-1 (4.10e-2)	4.572e-1 (3.36e-2) =	5.274e-1 (9.64e-3) + +	5.511e-1 (3.33e-2) + +	5.280e-1 (7.50e-3) + +	5.213e-1 (1.34e-2) + +	5.104e-1 (1.41e-2) + +	5.300e-1 (1.45e-2) + +	**4.530e-1 (2.49e-2) = =**
	60	4.479e-1 (2.87e-2)	**4.448e-1 (2.65e-2) =**	5.301e-1 (9.69e-3) + +	5.256e-1 (3.50e-2) + +	5.595e-1 (1.99e-2) + +	5.232e-1 (1.58e-2) + +	5.182e-1 (1.25e-2) + +	5.196e-1 (1.13e-2) + +	5.153e-1 (1.59e-2) + +
	80	**4.348e-1 (3.09e-2)**	4.407e-1 (2.39e-2) =	Na	5.050e-1 (1.11e-2) + +	5.223e-1 (3.63e-3) + +	5.201e-1 (1.10e-2) + +	5.156e-1 (1.15e-2) + +	5.170e-1 (1.18e-2) + +	5.179e-1 (1.45e-2) + +
WFG5	20	7.628e-1 (1.20e-2)	7.588e-1 (1.18e-2) =	**5.616e-1 (4.06e-2) - -**	7.619e-1 (1.35e-2) = =	6.853e-1 (1.11e-2) - -	7.330e-1 (1.64e-2) - -	6.482e-1 (2.98e-2) - -	6.086e-1 (3.02e-2) - -	6.017e-1 (3.69e-2) - -
	40	7.549e-1 (7.94e-3)	7.563e-1 (7.68e-3) =	6.650e-1 (1.60e-2) - -	7.544e-1 (5.45e-3) = =	7.220e-1 (7.80e-3) - -	7.360e-1 (8.95e-3) - -	6.890e-1 (1.62e-2) - -	6.534e-1 (2.60e-2) - -	**5.969e-1 (3.78e-2) - -**
	60	7.514e-1 (4.15e-3)	7.486e-1 (6.72e-3) =	6.923e-1 (1.72e-2) - -	7.513e-1 (6.42e-3) = =	7.350e-1 (5.86e-3) - -	7.389e-1 (6.45e-3) - -	7.045e-1 (1.15e-2) - -	**6.705e-1 (1.62e-2) - -**	7.118e-1 (1.08e-2) - -
	80	7.503e-1 (3.29e-3)	7.493e-1 (5.96e-3) =	Na	7.478e-1 (4.76e-3) = =	7.361e-1 (5.11e-3) - -	7.396e-1 (6.47e-3) - -	7.229e-1 (9.15e-3) - -	**6.789e-1 (2.95e-2) - -**	7.169e-1 (6.53e-3) - -
WFG6	20	6.239e-1 (3.58e-2)	**6.159e-1 (4.72e-2) =**	8.069e-1 (2.90e-2) + +	7.943e-1 (2.67e-2) + +	8.638e-1 (1.46e-2) + +	8.382e-1 (2.92e-2) + +	8.189e-1 (3.03e-2) + +	8.238e-1 (5.44e-2) + +	8.183e-1 (3.43e-2) + +
	40	6.733e-1 (5.73e-2)	**6.290e-1 (3.66e-2) -**	8.677e-1 (1.66e-2) + +	8.104e-1 (2.00e-2) + +	8.999e-1 (7.29e-3) + +	8.725e-1 (1.29e-2) + +	8.567e-1 (2.40e-2) + +	8.668e-1 (4.80e-2) + +	8.664e-1 (1.52e-2) + +
	60	7.067e-1 (5.15e-2)	**6.399e-1 (3.83e-2) -**	8.914e-1 (1.40e-2) + +	8.337e-1 (1.15e-2) + +	9.049e-1 (5.14e-3) + +	8.981e-1 (8.61e-3) + +	8.741e-1 (1.16e-2) + +	8.632e-1 (2.42e-2) + +	8.849e-1 (1.80e-2) + +
	80	6.898e-1 (5.59e-2)	**6.511e-1 (3.28e-2) =**	Na	8.345e-1 (1.88e-2) + +	9.130e-1 (3.75e-3) + +	9.027e-1 (9.32e-3) + +	8.902e-1 (1.19e-2) + +	8.690e-1 (3.20e-2) + +	8.923e-1 (1.12e-2) + +
WFG7	20	**4.784e-1 (2.02e-2)**	4.905e-1 (2.04e-2) +	6.880e-1 (1.47e-2) + +	6.071e-1 (1.51e-2) + +	6.604e-1 (9.59e-3) + +	6.399e-1 (2.18e-2) + +	6.538e-1 (2.56e-2) + +	6.290e-1 (2.96e-2) + +	6.302e-1 (3.66e-2) + +
	40	**4.774e-1 (2.05e-2)**	4.880e-1 (1.59e-2) +	6.938e-1 (6.52e-3) + +	6.312e-1 (1.30e-2) + +	6.798e-1 (6.69e-3) + +	6.621e-1 (1.51e-2) + +	6.668e-1 (9.92e-3) + +	6.278e-1 (2.08e-2) + +	6.550e-1 (1.38e-2) + +
	60	**4.891e-1 (2.21e-2)**	4.914e-1 (1.73e-2) =	6.911e-1 (5.11e-3) + +	6.378e-1 (1.30e-2) + +	6.788e-1 (4.60e-3) + +	6.749e-1 (7.56e-3) + +	6.665e-1 (1.03e-2) + +	6.269e-1 (1.37e-2) + +	6.705e-1 (1.39e-2) + +
	80	**4.748e-1 (1.75e-2)**	4.961e-1 (1.78e-2) +	Na	6.474e-1 (1.10e-2) + +	6.842e-1 (3.56e-3) + +	6.752e-1 (6.93e-3) + +	6.729e-1 (1.20e-2) + +	6.308e-1 (1.12e-2) + +	6.712e-1 (9.11e-3) + +
WFG8	20	6.268e-1 (3.66e-2)	**6.069e-1 (1.92e-2) =**	7.064e-1 (2.71e-2) + +	7.701e-1 (2.83e-2) + +	7.441e-1 (1.48e-2) + +	7.766e-1 (2.78e-2) + +	7.834e-1 (2.71e-2) + +	7.937e-1 (2.77e-2) + +	7.228e-1 (3.38e-2) + +
	40	5.798e-1 (2.40e-2)	**5.662e-1 (2.56e-2) =**	7.249e-1 (2.09e-2) + +	7.052e-1 (2.32e-2) + +	7.598e-1 (1.18e-2) + +	7.539e-1 (1.83e-2) + +	7.527e-1 (1.35e-2) + +	7.583e-1 (2.36e-2) + +	7.413e-1 (1.74e-2) + +
	60	5.743e-1 (3.63e-2)	**5.513e-1 (2.22e-2) =**	7.243e-1 (1.53e-2) + +	6.903e-1 (2.04e-2) + +	7.308e-1 (8.46e-3) + +	7.320e-1 (1.18e-2) + +	7.355e-1 (1.59e-2) + +	7.530e-1 (1.82e-2) + +	7.288e-1 (1.13e-2) + +
	80	5.709e-1 (2.48e-2)	**5.505e-1 (2.28e-2) -**	Na	6.750e-1 (1.49e-2) + +	7.424e-1 (7.26e-3) + +	7.365e-1 (8.92e-3) + +	7.294e-1 (1.60e-2) + +	7.353e-1 (1.37e-2) + +	7.342e-1 (1.56e-2) + +
WFG9	20	**6.371e-1 (1.03e-1)**	6.628e-1 (6.94e-2) =	8.581e-1 (4.20e-2) + +	7.723e-1 (5.10e-2) + +	9.010e-1 (3.29e-2) + +	8.018e-1 (6.56e-2) + +	8.283e-1 (6.72e-2) + +	7.488e-1 (4.57e-2) + +	8.254e-1 (7.49e-2) + +
	40	**6.617e-1 (7.51e-2)**	6.656e-1 (8.10e-2) =	9.161e-1 (2.14e-2) + +	8.485e-1 (4.03e-2) + +	9.398e-1 (2.20e-2) + +	8.807e-1 (4.82e-2) + +	8.730e-1 (4.63e-2) + +	7.820e-1 (4.42e-2) + +	8.960e-1 (3.06e-2) + +
	60	**6.573e-1 (9.88e-2)**	6.656e-1 (3.21e-2) =	9.380e-1 (1.69e-2) + +	9.022e-1 (3.36e-2) + +	9.515e-1 (1.61e-2) + +	9.231e-1 (1.74e-2) + +	9.114e-1 (2.92e-2) + +	8.113e-1 (5.54e-2) + +	9.131e-1 (2.51e-2) + +
	80	**6.031e-1 (9.24e-2)**	6.196e-1 (6.93e-2) =	Na	9.279e-1 (1.86e-2) + +	9.580e-1 (1.35e-2) + +	9.285e-1 (1.70e-2) + +	9.128e-1 (4.32e-2) + +	8.280e-1 (5.70e-2) + +	9.229e-1 (1.78e-2) + +
+ / - / =	DTLZ	GCS-PSO-RF vs.	11/0/9	9/6/0	14/0/6	14/0/6	11/5/4	11/9/0	8/10/2	10/10/0
		GCS-DE-RF vs.	Na	8/6/1	13/0/7	14/0/6	11/5/4	11/9/0	8/11/1	10/10/0
	WFG	GCS-PSO-RF vs.	4/6/26	21/6/0	29/3/4	28/8/0	27/8/1	28/8/0	30/4/2	26/8/2
		GCS-DE-RF vs.	Na	21/6/0	29/1/6	28/8/0	28/8/0	28/8/0	32/4/0	26/8/2
	Total	GCS-PSO-RF vs.	15/6/35	30/12/0	43/3/10	42/8/6	38/13/5	39/17/0	38/14/4	36/18/2
		GCS-DE-RF vs.	Na	29/12/1	42/1/13	42/8/6	39/13/4	39/17/0	40/15/1	36/18/2

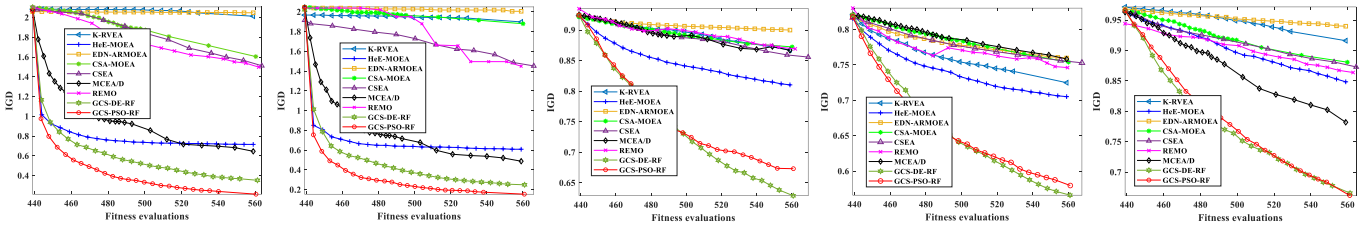
* from K-RVEA to REMO, the first and second symbols denote the significance test results versus GCS-PSO-RF and GCS-DE-RF, respectively

HeE-MOEA, EDN-ARMOEA, CSA-MOEA, CSEA, MCEA/D, and REMO on 43, 42, 38, 39, 38, and 36 out of 56 instances respectively; and GCS-DE-RF outperforms K-RVEA on 29 instances, and outperforms the others on 42, 42, 39, 39, 40, and 36 instances. In total, GCS-PSO-RF and GCS-DE-RF together get 35 best IGD results, followed by MCEA/D and REMO (which get 10 times). It is worth noting that the performance of the two GCS-assisted algorithms is not satisfactory on some problems like DTLZ1, DTLZ3, WFG1, and WFG5, which challenge the convergence ability of tested algorithms. Nevertheless, they still perform as well as, or even better than, some SAEAs like HeE-MOEA and EDN-ARMOEA on these problems.

Due to the space limitation, the statistical results in terms of GD values are shown in Table S.III in the supplementary material, which is summarized in Table IV. Different from the results in Table III, the GD results of EDN-ARMOEA, CSA-MOEA, and REMO are comparable to those of the two GCS-assisted algorithms. These results laterally confirm their lack of diversity maintenance, thus leading to a sharp performance decline on the IGD metric when compared with our proposed GCS-assisted algorithms. It is worth noting that CSA-MOEA and REMO are the representative classification-based SAEAs in *rank classification* and *relation classification*, respectively. Their performance discrepancy between IGD and GD metrics further confirms our previous statement, that these existing rank and relation classification-based surrogates lack the performance prediction for solutions in diversity. When comparing the two GCS-assisted algorithms, it is found that GCS-PSO-RF performs much better than

TABLE IV
GD SIGNIFICANCE TEST AMONG NINE SAEAS ON DTLZ AND WFG PROBLEMS, SUMMARIZED AS '+/-/='

Problem	Algorithm	GCS-DE-RF	K-RVEA	HeE-MOEA	EDN-ARMOEA	CSA-MOEA	CSEA	MCEA/D	REMO
DTLZ	GCS-PSO-RF vs.	7/0/13	9/1/5	9/4/7	12/8/0	8/0/12	15/0/5	16/4/0	12/6/2
	GCS-DE-RF vs.	Na	9/1/5	9/4/7	12/8/0	10/3/7	15/0/5	16/3/1	12/6/2
WFG	GCS-PSO-RF vs.	11/3/22	13/10/4	20/8/8	13/20/3	12/17/7	17/12/7	24/6/6	13/13/10
	GCS-DE-RF vs.	Na	13/11/3	20/7/9	12/21/3	11/19/6	17/13/6	22/8/6	13/16/7
Total	GCS-PSO-RF vs.	18/3/35	22/11/9	29/12/15	25/28/3	20/17/19	32/12/12	40/10/6	25/19/12
	GCS-DE-RF vs.	Na	22/12/8	29/11/16	24/29/3	21/22/13	32/13/11	38/11/7	25/22/9



(a) DTLZ2 (b) DTLZ5 (c) WFG6 (d) WFG8 (e) WFG9

Fig. 8. The mean IGD curves of eight SAEAs on different DTLZ and WFG problems (D = 40).

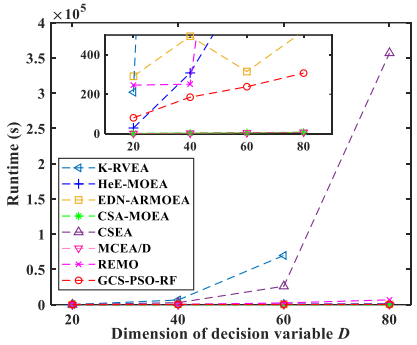


Fig. 9. Runtime consumed by eight SAEAs on DTLZ2 over D.

GCS-DE-RF, and this may own to our newly-designed velocity update strategy (i.e., Eq. (10)) in SLPSO to make it more suitable for expensive optimization.

To make the comparison more intuitive, we plot the IGD curves of these SAEAs on DTLZ2, DTLZ5, WFG6, WFG8, and WFG9, as shown in Fig. 8. It can be seen that the performance of the three classification-based SAEAs (CSA-MOEA, CSEA, and REMO) is similar. However, the performance of the three regression-based SAEAs is quite different, especially HeE-MOEA. The IGD curves of HeE-MOEA are always the third-best or fourth-best among all SAEAs, although there is still a massive gap with GCS-PSO-RF or GCS-DE-RF. However, the performance of the other two regression-based SAEAs is less than satisfactory, even worse than these classification-based SAEAs. This indicates that compared with classification surrogates, it is not necessarily better to use regression surrogates to approximate real FE. In addition, it is worth noting that the IGD curve of MCEA/D performs very well on DTLZ2 and DTLZ5, but is as bad as the other classification-based SAEAs on WFG problems. This is due to the decomposition MOEA framework adopted by MCEA/D, which can facilitate performance on problems with regular PF shapes like DTLZ. GCS-PSO-RF and GCS-DE-RF are the top two algorithms with fast convergence speed, although their performance differs slightly on different problems.

To further visually compare these algorithms' performance, the obtained non-dominated solutions from 20 runs with the median IGD values on different problems are shown in Figs. S1 to S5 in the supplementary material. These figures show that the solutions obtained by GCS-PSO-RF are closer to the PF than those obtained by other SAEAs on the same instance. More specifically, only the solutions obtained by GCS-PSO-RF can approximate the true PF on DTLZ2 and DTLZ5, while the solutions obtained by others are scattered sporadically in the objective space. This shows that compared with other SAEAs, GCS-PSO can search for better-converged solutions with the assistance of GCS. The above experimental results and analysis indicate that GCS-PSO is competitive in solving EMOPs.

2) Runtime Analysis

In this subsection, we study the runtimes of all SAEAs. DTLZ2 is adopted as the example here, where the runtime curves over the dimension of decision variables are shown in Fig. 9. From Fig. 9 we can see that the runtimes of K-RVEA and CSEA go up dramatically versus D. It is not surprising that this result appears in K-RVEA, because the larger D is, the more training data samples required for an applicable Kriging-based surrogate. In addition, the surrogate in CSEA is an FNN instead of traditional classification models. Its structural complexity (e.g., the number of hidden neurons) is also positively correlated with D, dramatically changing the training time. However, the runtime of EDN-ARMOEA, in which the surrogate is also an FNN, does not fluctuate much. This should be thanks to the dropout technology [78] used in it and the fixed number of hidden neurons. The runtimes of other SAEAs also keep increasing trend versus D, although not dramatically. Overall, the runtime of GCS-PSO-RF is always competitive, although is larger than that of CSA-MOEA. By comprehensively analyzing the statistical results and the runtimes among these SAEAs, we can see that GCS-PSO is very competitive in both performance and computational cost.

3) Experiments on Many-objective Optimization Problems

In this subsection, we conduct comparison experiments on many-objective DTLZ problems to further explore the potential of GCS-PSO-RF in handling more complex EMOPs. Table S. IV and S.V in the supplementary material show the statistical results in terms of IGD and GD with 5-, 8-, and 10-objective

TABLE V
SIGNIFICANCE TEST BETWEEN GCS-PSO-RF AND OTHER SAEAS ON MANY-OBJECTIVE DTLZ PROBLEMS (D = 20), SUMMARIZED AS '+/-/='

Metric	K-RVEA	HeE-MOEA	EDN-ARMOEA	CSA-MOEA	CSEA	MCEA/D	REMO
IGD	9/4/2	10/1/4	11/0/4	10/0/5	9/6/0	9/6/0	7/6/2
GD	6/7/2	5/3/6	6/7/2	8/7/0	5/5/5	11/1/3	6/6/3

TABLE VI
SIGNIFICANCE TEST BETWEEN GCS-PSO-RF AND GCS-NOGF, SUMMARIZED AS '+/-/='

Metric	DTLZ	WFG	Total
IGD	8/1/11	12/1/23	20/2/34
GD	6/5/9	12/5/19	17/10/28

instances, respectively. Due to the limited space, only the summarized results are given in Table V.

From these results, other compared SAEAs perform slightly worse than REMO and GCS-PSO-RF on IGD or GD. In particular, GCS-PSO-RF is significantly better than others on IGD, which furthermore confirms the competitiveness of our algorithm in convergence and diversity maintenance when dealing with expensive many-objective optimization problems.

D. Ablation Experiments

1) Effectiveness of Grid Front

The update of the grid front ensures that the solutions stored in it are always non-grid-dominated, which is essential for SLPSO to search for promising offspring in the optimization process. Herein, a GCS-PSO-RF variant without grid front update (denoted as GCS-noGF) is designed, and comparison experiments between GCS-PSO-RF are conducted to investigate the effectiveness of the grid front update strategy. The statistical results in terms of IGD and GD are given in Table S.VI and S.VII in the supplementary material, which is summarized in Table VI. These results show that GCS-PSO-RF performs superior or similarly to GCS-noGF on almost all instances in terms of IGD. As for GD, GCS-PSO-RF still gets better or equivalent results on most instances. The results on the two metrics confirm that GCS-PSO-RF performs much better than GCS-noGF, especially in terms of diversity. This indicates that the grid front update strategy can improve the performance of GCS-PSO in most problems, and it plays a remarkable role in diversity maintenance.

2) Effectiveness of Two Indicators

The two indicators in GCS-PSO play an important role in the update process. In this subsection, we conduct ablation experiments to analyze the effectiveness of the two indicators. Two GCS-PSO-RF variants are designed here, denoted by GCS-noGR and GCS-noNCI. The GR is not used in GCS-noGR, and the NCI is not used in GCS-noNCI. In other words, during the update, solution $\mathbf{x}$ only can be updated to $\mathbf{y}$ iff NCI($\mathbf{x}$) > NCI($\mathbf{y}$) in GCS-noGR, and iff GR($\mathbf{x}$) > GR($\mathbf{y}$) in GCS-noNCI. The statistical results in terms of IGD and GD are given in Table S.VIII and S.IX in the supplementary material. Table VII shows the summary results of them. Through the results, we can see that GCS-PSO-RF outperforms GCS-noGR on most instances. This indicates that GR, as the primary indicator in GCS-PSO, plays a significant role in convergence maintenance. In addition, GCS-PSO-RF performs similarly to GCS-noNCI on most instances. Compared with GR, NCI is an additional indicator that promotes diversity and distinguishes relations between solutions when GR is out of work. However, this does not mean NCI is useless since the IGD and GD values of GCS-PSO-RF are generally better than those of GCS-noNCI on WFG problems. This is because the WFG problems are asymmetric, and NCI can better promote diversity on such problems. This confirms that NCI is beneficial for the performance of GCS-PSO.

TABLE VII
SIGNIFICANCE TEST BETWEEN GCS-PSO-RF, GCS-NOGR, AND GCS-NONCI, SUMMARIZED AS '+/-/='

Problem	Metric	GCS-noGR	GCS-noNCI
DTLZ	IGD	10/2/8	1/1/18
	GD	9/6/5	2/1/17
WFG	IGD	28/0/8	1/0/35
	GD	35/10/11	5/2/49
Total	IGD	38/2/16	2/1/53
	GD	35/10/11	5/2/49

To make the effectiveness of the two indicators more intuitive, the non-dominated solutions obtained by these three algorithms with median IGD value are plotted in Fig. S6 in the supplementary material. From these figures, we can see that: (1) the results obtained by GCS-noGR are the worst of the three algorithms, neither in terms of convergence nor diversity; (2) the results by GCS-PSO-RF are generally better than those by GCS-noNCI in terms of diversity. Most notably, GCS-PSO-RF is also significantly better than GCS-noNCI in terms of both convergence and diversity on some problems. It indicates that NCI can promote both convergence and diversity.

Through the above ablation experimental results, we can conclude that GR and NCI indicators are both indispensable to GCS-PSO. As the primary indicator for convergence maintenance, GR is crucial to the performance of GCS-PSO. In addition, the NCI can further promote diversity and decide about the update when GR cannot work. It improves the performance of GCS-PSO, although not very significant.

3) Effectiveness of DBA Infill Criterion

In this subsection, we investigate the effectiveness of the proposed DBA infill criterion in GCS-PSO. Herein, we design two GCS-PSO-RF variants with two different random infill criteria, denoted as GCS-rand1 and GCS-rand2.

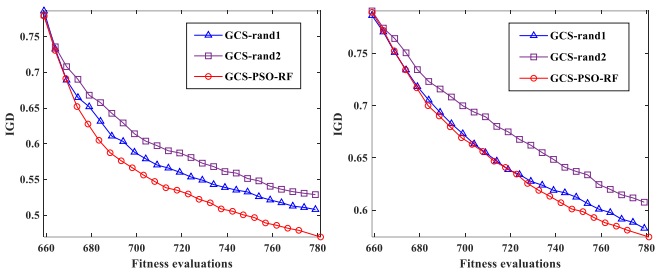
GCS-rand1: During the infill selection, μ candidate solutions are randomly selected from the population.

GCS-rand2: During the infill selection, all solutions in the population are clustered into μ subpopulations in the decision space by the k-means clustering [79] method. Then one candidate solution is randomly selected from each subpopulation.

The two GCS-PSO-RF variants are compared with GCS-PSO-RF on DTLZ and WFG problems, where the statistical results in terms of IGD and GD are shown in Table S.X and S.XI in the supplementary material. Table VIII gives the summary results of them. It can be seen that the performance of GCS-PSO-RF is superior or similar to the other two variants on all instances. This shows that the DBA can select more potential promising candidate solutions compared with random

TABLE VIII
SIGNIFICANCE TEST BETWEEN GCS-PSO-RF AND TWO VARIANTS WITH DIFFERENT INFILL CRITERIA, SUMMARIZED AS '+/-/='

Problem	Metric	GCS-rand1	GCS-rand2
DTLZ	IGD	8/0/12	8/0/12
	GD	8/0/12	9/0/11
WFG	IGD	5/0/31	21/0/15
	GD	8/0/28	21/1/14
Total	IGD	13/0/43	29/0/27
	GD	16/0/40	30/1/25



(a) WFG3 (b) WFG8

Fig. 10. The mean IGD curves of GCS-PSO-RF and its variants on WFG3 and WFG8 (D = 60).

selection. Excluding GCS-PSO-RF, it seems that GCS-rand1 performs much better than GCS-rand2 on WFG problems. Therefore, we take WFG3 and WFG8 with D = 60 as examples and compare the IGD curves. As shown in Fig. 10, the IGD curve of GCS-PSO-RF is always the best, while GCS-rand1 ranks second and GCS-rand2 is the worst. This result indicates that trying to maintain the diversity of selected candidate solutions by clustering method in infill selection may be ineffective or even counterproductive. Unlike the unsupervised clustering method, the DBA takes the optimal solutions in the DB as references to ensure the nearby candidate solutions are similar in diversity. Therefore, we only need to consider selecting candidate solutions with good GR values near each reference point, and their dual potential in both convergence and diversity can be ensured.

4) Analysis of Parameter div

The grid technique can discrete the continuous objective space, in which the *div* is a manual-set parameter that controls the degree of this discretization. The smaller the *div*, the rougher the discretization, and the more solutions will be assigned to the same grid, thus making their grid locations lack effective position discrimination ability. On the contrary, a large *div* will cause too refined discretization, and more training burden will be put on each multi-class classification model in GCS because of the larger number of classes in the training dataset. It is worth noting that in extreme cases, the grid division would lose its meaning in discretization if the *div* approaches infinity. Thus a proper *div* value is necessary for GCS-PSO-RF.

Herein we conduct GCS-PSO-RF with different *div* values to investigate the sensitivity of GCS-PSO to the parameter *div*. GCS-PSO-RF with *div* = 10, 20, 30, 40, 50, 60, and 70 are tested on DTLZ2, DTLZ5, WFG1, WFG3, WFG5, WFG6 and WFG8 problems with D = 20, 40, 60, and 80. The results of mean IGD and runtime are shown in Table S.XII in the supplementary material. On the one hand, the computational cost of GCS-PSO-RF is positively correlated with the *div* value, i.e., the smaller the value of *div*, the lower the computational cost of GCS-PSO-RF. On the other hand, GCS-PSO-RF seems to favor smaller *div* values for DTLZ problems and larger *div* values for WFG problems in terms of performance. However, it is worth noting that the use of different *div* values does not significantly affect the performance of GCS-PSO-RF. GCS-PSO-RF can still perform well on WFG problems without the most appropriate *div* value. Considering both the computational cost and performance, we set the *div* to 30 in this paper. Note that this setting is based on our experience, and there may be better *div* settings for other problems.

TABLE IX
MEAN IGD VALUES OF DIFFERENT ALGORITHMS ON CSDOP

Problem	NSGA-II	CSA-MOEA	MCEA/D	GCS-PSO-CSVC	GCS-PSO-RF
IGD	9.1521	4.3096	4.6673	2.0539	**1.8518**
FE_{max}	30000	1000			

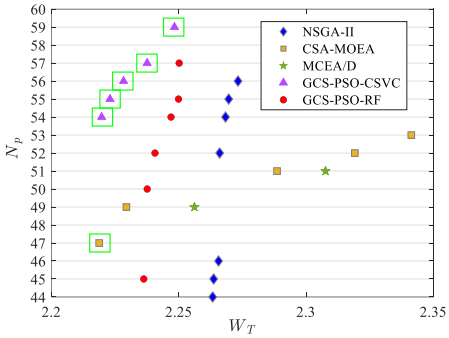


Fig. 11. Non-dominated solutions obtained by NSGA-II, CSA-MOEA, MCEA/D, GCS-PSO-CSVC, and GCS-PSO-RF on CSDOP after 20 runs.

E. Performance on Real-world Car Structure Design Optimization Problem

In this subsection, we investigate the performance of the proposed GCS-PSO on the real-world high-dimensional car structure design optimization problem (CSDOP). The CSDOP is modeled from the actual car structure design optimization results by Mazda, and it consists of 222 decision variables and 42 constraints. More details can be found in [80]. CSDOP aims to minimize the total weight of three different cars (SUV, LV, and SV) and maximize the number of common gauge parts, which can be formulated as:

$$\min \begin{cases} W_T = W_{SUV} + W_{LV} + W_{SV} \\ -N_p \end{cases} \tag{14}$$

where W_{SUV}, W_{LV}, W_{SV} are the weights of SUV, LV, and SV, respectively; W_T is the total weight of three cars; N_p is the number of common gauge parts. For the convenience of experimental design, all constraints are not considered in this paper, and the decision variables are treated as continuous variables.

Considering the dimension of decision variables, CSDOP may challenge some SAEAs in terms of computational cost. Therefore, the experiments are only conducted on CSA-MOEA, MCEA/D, GCS-PSO-CSVC, and GCS-PSO-RF. The popula-

tion size is set to 100. Each algorithm runs 20 times, and the Pareto non-dominated solutions are output. For a fair comparison, the initial DB for surrogate training of each independent experiment is the same, and its size is set to 900. The $FE_{\max}$ here is set to 1000. All other parameters remain unchanged. In addition, NSGA-II is conducted on CSDOP with $FE_{\max} = 30000$, and compared with the three SAEAs to verify the superiority of SAEAs in solving CSDOP.

Since the true PF of CSDOP is unknown, all Pareto non-dominated solutions from all experiments are taken as the approximate PF for IGD calculation. The mean IGD values are shown in Table IX. We can see that all SAEAs can get smaller IGD values than NSGA-II. This result further confirms the effectiveness of SAEAs in solving real-world EMOP since they can find better solutions within only one-thirtieth of the number of FEs compared to NSGA-II. In addition, the two GCS-PSO algorithms get better IGD results than CSA-MOEA. Then we take the Pareto non-dominated solutions in 20 runs as the output result of each algorithm and plot them in the same figure, as shown in Fig. 11. All solutions in the green box represent the optimal solution set for all algorithms. As can be seen, six optimal solutions are screened out from the comprehensive results of all algorithms, of which GCS-PSO-CSVC contributes five. It shows that on CSDOP, GCS-PSO-CSVC can obtain better solutions with smaller total weight and more common gauge parts than the other three algorithms. Apart from that, the solution set obtained by GCS-PSO-RF performs better than CSA-MOEA and NSGA-II, although slightly worse than that of GCS-PSO-CSVC. This result confirms the competitiveness of GCS-PSO in dealing with real-world EMOPs.

V. Conclusion

In this paper, we propose a new classification way named *grid classification* by discretizing the objective space into hyper-grids for classification-based SAEAs. Via *grid classification*, the lightweight GCS is built and the GCS-PSO is proposed for solving EMOPs. During the optimization process, the GCS can replace the expensive real FE by predicting the grid locations of the candidate solutions. Then these predicted grid locations are further transformed into two indicators for helping the solution comparison and update. The proposed DBA infill criterion can select candidate solutions with a higher probability of good convergence and diversity to update the GCS. To the best of our knowledge, this GCS-PSO is the first attempt at grid classification-based SAEA.

In the experiments, we use different classification models to construct GCS, and the experimental results on DTLZ and WFG show that RF is very suitable for GCS construction. Therefore, RF is chosen for constructing GCS in GCS-PSO, and the comparison of experimental results with other SAEAs shows the competitiveness of GCS-PSO in solving EMOPs with different dimensions of decision variables. Moreover, the study on real-world car structure design optimization confirms the effectiveness of GCS-PSO in dealing with high-dimensional real-world EMOPs.

References

[1] Z. -G. Chen, Z. -H. Zhan, S. Kwong, and J. Zhang, "Evolutionary computation for intelligent transportation in smart cities: A survey," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 83–102, 2022.

[2] L. M. Antonio and C. A. C. Coello, "Coevolutionary multiobjective evolutionary algorithms: Survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 851–865, 2018.

[3] Z. -H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, vol. 55, pp. 59–110, 2022.

[4] S. -C. Liu, Z. -H. Zhan, K. C. Tan, and J. Zhang, "A multi-objective framework for many-objective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13654–13668, 2022.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[6] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[7] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[8] Z. -H. Zhan, J. Li, J. Cao, J. Zhang, H. S. -H. Chung, and Y. -H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, 2013.

[9] S. -C. Liu *et al.*, "Many-objective job-shop scheduling: A multiple populations for multiple objectives-based genetic algorithm approach," *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1460–1474, 2023.

[10] Q. -T. Yang, Z. -H. Zhan, S. Kwong, and J. Zhang, "Multiple populations for multiple objectives framework with bias sorting for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 5, pp. 1340–1354, 2023.

[11] J. -Y. Li *et al.*, "A multipopulation multiobjective ant colony system considering travel and prevention costs for vehicle routing in COVID-19-like epidemics," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25062–25076, 2022.

[12] X. -F. Liu *et al.*, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, 2019.

[13] G. Dan, T. Chai, J. Ding, and Y. Jin, "Small data driven evolutionary multi-objective optimization of fused magnesium furnaces," in *Proc. Symp. Series Comput. Intell.*, 2016, pp. 1–8.

[14] S. -H. Wu, Z. -H Zhan, and J. Zhang, "SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 478–491, 2021.

[15] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by Gaussian random field meta-models," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, 2006.

[16] X. Song, Y. Zhang, D. Gong, H. Liu, and W. Zhang, "Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 595–609, 2023.

[17] J. -Q. Yang *et al.*, "Bi-directional feature fixation-based particle swarm optimization for large-scale feature selection," *IEEE Trans. Big Data*, vol. 9, no. 3, pp. 1004–1017, 2023.

[18] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multi-objective structural optimization and beyond," *IEEE Comput. Intell. Mag.*, vol. 4, no. 3, pp. 62–76, 2009.

[19] X. -F. Liu, Z. -H Zhan, and J. Zhang, "Resource-aware distributed differential evolution for training expensive neural-network-based controller in power electronic circuit," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6286–6296, 2022.

[20] J. -Y. Li, Z. -H Zhan, J. Xu, S. Kwong, and J. Zhang, "Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameter optimization in convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 2338–2352, 2023.

[21] Y. -Q. Wang, J. -Y. Li, C. -H. Chen, J. Zhang, and Z. -H Zhan, "Scale adaptive fitness evaluation-based particle swarm optimization for hyperparameter and architecture optimization in neural networks and deep learning," *CAAI Trans. Intell. Technol.*, vol. 8, no. 3, pp. 849–862, 2023.

[22] S. Z. Martinez and C. A. C. Coello, "MOEA/D assisted by RBF net-

works for expensive multiobjective optimization problems," in *Proc. Genet. Evol. Compu. Conf.*, 2013, pp. 1405–1412.

[23] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. De Freitas, "Taking the human out of loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[25] Q. Zhang, W. Liu, and E. Tsang, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, 2010.

[26] T. Chugh, Y. Jin, K. Miettinen, and J. Hakanen, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, 2018.

[27] Z. Song, H. Wang, C. He, and Y. Jin, "A Kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1013–1027, 2021.

[28] Q. Lin, X. Wu, L. Ma, J. Li, M. Gong, and C. A. C. Coello, "An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 631–645, 2022.

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

[30] J. Luo, Y. Dong, Z. Zhu, W. Cao, and X. Li, "Expensive multiobjective optimization based on information transfer surrogate," *IEEE Trans. Syst., Man, Cybern.*: *Syst.*, vol. 53, no. 3, pp. 1684–1696, 2023.

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

[32] J. Tang, H. Wang, and X. Lin, "Surrogate-assisted multi-objective optimization via knee-oriented Pareto front estimation," *Swarm Evol. Comput.*, 2023, DOI: 10.1016/j.swevo.2023.101252.

[33] J. Xu, Y. Jin, and W. Du, "A federated data-driven evolutionary algorithm for expensive multi-/many-objective optimization," *Compl. Intell. Syst.*, vol. 7, pp. 3093–3109, 2021.

[34] C. Yang, J. Ding, Y. Jin, and T. Chai, "Offline data-driven multiobjective optimization: Knowledge transfer between surrogates and generation of final solutions," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 409–423, 2020.

[35] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, 2018.

[36] A. Habib *et al.*, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 1000–1014, 2019.

[37] J. -Y. Li, Z. -H. Zhan, and J. Zhang, "Evolutionary computation for expensive optimization: A survey," *Mach. Intell. Res.*, vol. 19, no. 1, pp. 3–23, 2022.

[38] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 442–458, 2019.

[39] S. R. -Gonzalez and V. Nieuwenhuyse, "A survey on kriging-based infill algorithms for multiobjective simulation optimization," *Comput. Oper. Res.*, vol. 116, 2020.

[40] 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. Syst., Man, Cybern.*: *Syst.*, vol. 52, no. 4, pp. 2084–2097, 2021.

[41] T. Runarsson, "Ordinal regression in evolutionary computation," in *Proc. Para. Prob. Solv. Nature*, 2006, pp. 1048–1057.

[42] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *Proc. Cong. Evol. Comput.*, 2015, pp. 2883–2890.

[43] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[44] J. Laaksonen and E. Oja, "Classification with learning k-nearest neighbors," in *Proc. Neur. Net.*, 1996, pp. 1480–1483.

[45] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, 2008.

[46] J. Zhang, A. Zhou, K. Tang, and G. Zhang, "Preselection via classification: A case study on evolutionary multiobjective optimization," *Inf. Sci.*, vol. 465, pp. 388–403, 2018.

[47] 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. Evol. Comput.*, vol. 23, no. 1, pp. 74–88, 2018.

[48] J. Li *et al.*, "A classification surrogate-assisted multi-objective evolutionary algorithm for expensive optimization," *Knowl-Based Syst.*, vol. 242, 2022, DOI: 10.1016/j.knosys.2022.108416.

[49] T. Sonoda and M. Nakata, "Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multi-objective problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 6, pp. 1581–1595, 2022.

[50] J. Zhang, L. He, and H. Ishibuchi, "Dual fuzzy classifier-based evolutionary algorithm for expensive multiobjective optimization," *IEEE Trans. Evol. Comput.*, 2022, DOI: 10.1109/TEVC.2022.3195668.

[51] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (GANs)," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3129–3142, 2021.

[52] H. Hao, A. Zhou, and H. Zhang. "An approximated domination relationship based on binary classifiers for evolutionary multiobjective optimization," in *Proc. Cong. Evol. Comput.*, 2021, pp. 2427–2434.

[53] G. Guo, W. Li, B. Yang, W. Li, and C. Yin, "Predicting pareto dominance in multi-objective optimization using pattern recognition," in *Proc. Intell. Sys. Des. Engi. App.*, 2012, pp. 456–459.

[54] Y. Yuan and W. Banzhaf, "Expensive multi-objective evolutionary optimization assisted by dominance prediction," *IEEE Trans. Evol. Comput.*, vol. 26, no. 1, pp, 159–173, 2021.

[55] H. Hao, A. Zhou, H. Qian, and H. Zhang, "Expensive multiobjective optimization by relation learning and prediction," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1157–1170, 2022.

[56] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, 2013.

[57] J. Cheng, G. G. Yen, and G. Zhang, "A grid-based adaptive multi-objective differential evolution algorithm," *Inf. Sci.*, vol. 367, pp. 890–908, 2016.

[58] P. Wang, B. Xue, M. Zhang, and J. Liang, "A grid-dominance based multi-objective algorithm for feature selection in classification," in *Proc. Cong. Evol. Comput.*, 2021, pp. 2053–2060.

[59] R. Cheng and Y. C. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Inf. Sci.*, vol. 291, pp. 43–60, 2015.

[60] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *in Proc. IEEE Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.

[61] J. -R. Jian, Z. -G. Chen, Z. -H. Zhan, and J. Zhang, "Region encoding helps evolutionary computation evolve faster: A new solution encoding scheme in particle swarm for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 779–793, 2021.

[62] Z. -H. Zhan, J. -Y. Li, S. Kwong, and J. Zhang, "Learning-aid evolution for optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 6, pp. 1794–1808, 2023.

[63] 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," *Techno.*, vol. 42, no. 1, pp. 55–61, 2000.

[64] Q. -T. Yang, Z. -H. Zhan, Y. Li, and J. Zhang, "Social learning particle swarm optimization with two-surrogate collaboration for offline data-driven multiobjective optimization", in *Proc. Genet. Evol. Compu. Conf.*, 2022, pp. 49–57.

[65] I. Rish, "An empirical study of the naive Bayes classifier," *Artif. Intell.*, vol. 3, no. 22, pp. 41–46, 2001.

[66] L. Jiang and R. Yao, "Modelling personal thermal sensations using c-support vector classification algorithm," *Build. Environ.*, vol. 99, pp. 98–106, 2016.

[67] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[68] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Proc. Evol. Multi-Criterion Optim.*, 2005, pp. 105–145.

[69] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.

[70] J. -Y. Li, Z. -H. Zhan, C. Wang, H. Jin, and J. Zhang, "Boosting data-driven evolutionary algorithm with localized data generation," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 923-937, 2020.

[71] V. Van, A. David, and B. L. Gary, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evol. Comput.*, vol. 8, no. pp. 125–147, 2000.

[72] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fon-

seca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, 2003.

[73] J. -Y. Li, Z. -H. Zhan, Y. Li, and J. Zhang, "Multiple tasks for multiple objectives: A new multiobjective optimization method via multitask optimization," *IEEE Trans. Evol. Comput.*, 2023, DOI: 10.1109/TEVC.2023.3294307.

[74] R. M. Storn and K. V. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[75] J. -Y. Li, Z. -H. Zhan, K. C. Tan, and J. Zhang, "A meta-knowledge transfer-based differential evolution for multitask optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 719–734, 2022.

[76] Z. -H. Zhan, Z. -J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4633–4647, 2020.

[77] Y. Jiang, Z. -H. Zhan, K. C. Tan, and J. Zhang, "Knowledge learning for evolutionary computation," *IEEE Trans. Evol. Comput.*, 2023, DOI: 10.1109/TEVC.2023.3278132.

[78] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[79] A. Likas, N. Vlassis, and J. J. Verbeek, "The global *k*-means clustering algorithm," *Pattern Recogn.*, vol. 36, no. 2, pp. 451–461, 2003.

[80] T. Kohira, H. Kemmotsu, O. Akira, and T. Tatsukawa, "Proposal of benchmark problem based on real-world car structure design optimization," in *Proc. Genet. Evol. Compu. Conf. Compa.*, 2018, pp. 183–184.

Qi-Te Yang (Student Member, IEEE) received the B.S. degree from Dalian University, Dalian, China, in 2017, and the M.S. degree from Xiangtan University, Xiangtan, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His research interests mainly include computational intelligence, data-driven optimization, multiobjective optimization, and their applications in real-world problems. He has been invited as a reviewer of the *IEEE Transactions on Evolutionary Computation* and the *IEEE Transactions on Systems, Man and Cybernetics: Systems*.

Zhi-Hui Zhan (Fellow, IEEE) received the Bachelor's degree and the Ph. D. degree in Computer Science from the Sun Yat-Sen University, Guangzhou China, in 2007 and 2013, respectively.

He is currently a Changjiang Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China and with the College of Artificial Intelligence, Nankai University, Tianjin, China. His current research interests include evolutionary computation, swarm intelligence, and their applications in real-world problems.

Prof. Zhan is an IEEE Fellow. He was a recipient of the IEEE Computational Intelligence Society (CIS) Outstanding Early Career Award in 2021, the Outstanding Youth Science Foundation from National Natural Science Foundations of China (NSFC) in 2018, and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. He is listed as the World's Top 2% Scientist for both Career-Long Impact and Year Impact in Artificial Intelligence, and is listed as the Elsevier Highly Cited Chinese Researcher in Computer Science. He is currently an Associate Editor of the *IEEE Transactions on Evolutionary Computation*, the *IEEE Transactions on Systems, Man and Cybernetics: Systems*, and the *IEEE Transactions on Artificial Intelligence*.

Xiao-Fang Liu (Member, IEEE) received the B.S. degree and Ph.D degree in computer science from Sun Yat-Sen University, Guangzhou, China, in 2015 and 2020, respectively.

She is currently a lecturer with College of Artificial Intelligence, Nankai University. Her current research interests include artificial intelligence, evolutionary computation, swarm intelligence, and their applications in design and optimization such as cloud computing resources scheduling, multirobot systems.

Jian-Yu Li (Member, IEEE) received the Bachelor's degree and the Ph. D. degree in Computer Science and Technology from the South China University of Technology, China, in 2018 and 2022, respectively.

His research interests mainly include computational intelligence, data-driven optimization, machine learning including deep learning, and their applications in real-world problems, and in environments of distributed computing and big data. He has been invited as a reviewer of the *IEEE Transactions on Evolutionary Computation* and the *IEEE Transactions on Systems, Man and Cybernetics: Systems*.

Jun Zhang (Fellow, IEEE) received the PhD degree in Electrical Engineering from the City University of Hong Kong in 2002.

He is a professor with the School of Electrical and Engineering, Hanyang University ERICA, Ansan 15588, South Korea, and he holds a Distinguished Professorship in Nankai University. Prof. Zhang's research contributions span over 300 peer-reviewed publications, of which more than 180 appear in IEEE Transactions. His research interests include Computational Intelligence, cloud computing, big data mining, and power electronic circuits.

Professor Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and was appointed as a Cheung Kong Chair Professor in 2013 by the Ministry of Education, China. Presently, Prof. Zhang serves as an associate editor for both the *IEEE Transactions on Artificial Intelligence* and the *IEEE Transactions on Cybernetics*.