A Classification-Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization

Linqiang Pan, Cheng He, Ye Tian, Handing Wang, *Member, IEEE*, Xingyi Zhang, and Yaochu Jin, *Fellow, IEEE*

Abstract—Surrogate-assisted evolutionary algorithms (SAEAs) have been developed mainly for solving expensive optimization problems where only a small number of real fitness evaluations are allowed. Most existing SAEAs are designed for solving low-dimensional single or multiobjective optimization problems, which are not well suited for many-objective optimization. This paper proposes a surrogate-assisted many-objective evolutionary algorithm that uses an artificial neural network to predict the dominance relationship between candidate solutions and reference solutions instead of approximating the objective values separately. The uncertainty information in prediction is taken into account together with the dominance relationship to select promising solutions to be evaluated using the real objective functions. Our simulation results demonstrate that the proposed algorithm outperforms the state-of-the-art evolutionary algorithms on a set of many-objective optimization test problems.

Index Terms—Classification, expensive many-objective optimization, Pareto dominance, surrogate-assisted evolutionary optimization.

L. Pan is with the Key Laboratory of Image Information Processing and Intelligent Control of Education Ministry of China, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the School of Electric and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China (e-mail: lqpanhust@gmail.com).

C. He was with the Department of Computer Science, University of Surrey, Surrey GU2 7XH, U.K. He is now with the Key Laboratory of Image Information Processing and Intelligent Control of Education Ministry of China, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: chenghehust@gmail.com).

Y. Tian was with the Department of Computer Science, University of Surrey, Surrey GU2 7XH, U.K. He is now with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: field910921@gmail.com).

H. Wang is with the Department of Computer Science, University of Surrey, Surrey GU2 7XH, U.K. (e-mail: wanghanding.patch@gmail.com).

X. Zhang is with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: xyzhanghust@gmail.com)

Y. Jin is with the Department of Computer Science, University of Surrey, Surrey GU2 7XH, U.K., and also with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: yaochu.jin@surrey.ac.uk).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2018.2802784

I. INTRODUCTION

MULTIOBJECTIVE optimization problems (MOPs) are commonly seen in the real world, e.g., electrical engineering [1], industrial scheduling [2], and robotics [3]. These problems aim to simultaneously optimize more than two often conflicting objectives, which can be mathematically formulated as follows:

$$\text{minimize } F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$$
$$\text{subject to } x \in X \tag{1}$$

where m is the number of objectives and x is the decision vector [4]. A large number of multiobjective evolutionary algorithms (MOEAs) were proposed in the past decades, e.g., the region-based selection algorithm (PESA-II) [5], the improved strength Pareto evolutionary algorithm [6], the elitist nondominated sorting genetic algorithm (NSGA-II) [7], the improved indicator-based evolutionary algorithm [8], and the MOEA based on decomposition (MOEA/D) [9]. These algorithms have been shown to be effective in solving MOPs with two or three objectives [10].

The performance of MOEAs, in particularly those based on dominance relation comparisons, dramatically degenerates on MOPs with more than three objectives, also known as many-objective optimization problems (MaOPs) [11]. The performance deterioration can be largely attributed to the fact that the rate of nondominated solutions in a limited population increases exponentially as the number of objectives increases, making traditional Pareto-based MOEAs fail to distinguish these solutions [12], [13].

Plenty of research efforts have been dedicated to designing MOEAs for many-objective optimization and their applications [14], [15], e.g., the evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach (NSGA-III) [11], the hypervolume (HV)-based evolutionary algorithm [16], the knee-point driven evolutionary algorithm [17], the improved two-archive algorithm for many-objective optimization (Two_Arch2) [18], the

reference vector-based evolutionary algorithm (RVEA) [13], and the region division-based many-objective optimization evolutionary algorithm [19]. Note that most existing MOEAs for solving MaOPs typically require tens of thousands of fitness evaluations (FEs).

One grand challenge in solving many real-world MaOPs is that one single FE is computationally and/or financially very expensive, since it requires time-consuming computer simulations or physical experiments [20], [21], e.g., in aerodynamic design optimization [22], drug design [23], or flowshop scheduling problems in [24]. Take a ten-job and five-machine flowshop scheduling problem as an example [24], it will take over 200 days for a conventional MOEA if a total of 10 000 FEs is used, which is impractical.

One popular approach to expensive optimization problems is to introduce computationally efficient surrogates for approximating the expensive FEs [25], [26]. Various types of surrogates are commonly used in expensive optimization, including polynomial response surface methodology [27], radial basis function [28], Gaussian process model, also known as Kriging model [29], or sometimes as efficient global optimization (EGO), artificial neural networks [30], and support vector machines [31]. A variety of surrogate-assisted evolutionary algorithms (SAEAs) were proposed to handle single-objective optimization using classification or regression-based fitness approximation, e.g., the neural network assisted evolution strategy [32], the feasibility structure modeling assisted memetic algorithm [33], the classification-assisted memetic algorithm [34], and the surrogate-assisted cooperative particle swarm optimization [35]. Furthermore, many SAEAs for expensive multiobjective optimization were proposed in the past decades, e.g., the generalized surrogate-assisted multiobjective memetic algorithm (GS-MOMA) [36], the weighted aggregation-based multiobjective optimization assisted by EGO (ParEGO) [37], the EGO assisted MOEA/D (MOEA/D-EGO) [21], the Pareto rank learning MOEA [38], and the Kriging assisted RVEA (K-RVEA) [39], for solving MaOPs. These SAEAs have been shown promising in reducing the number of FEs for expensive multiobjective optimization [40].

Although numerous SAEAs have been proposed for solving expensive optimization problems, four main challenges remain in designing effective surrogate-assisted optimization algorithms. First, choice of the surrogate model is not straightforward as there are many different types of surrogates but there is no simple rule for determining which type should be chosen [41]. Second, it is nontrivial to determine what should be predicted by the surrogate [42]. Third, it is hard to decide in which component of the MOEA the surrogate should be applied [43], [44]. Last but not the least, surrogate management, i.e., to decide which solution within the current population should be re-evaluated using the expensive FE method, is critical for the performance of SAEAs.

As suggested in [39], building surrogates for MaOPs will incur increasing computational costs as the number of objectives increases, if the surrogates are used for approximating the objective functions. Thus, it is very desirable to build surrogates that can directly predict the dominance relationship

of compared solutions [45], [46], or the ranking of the solutions [47], [48]. Inspired by the above ideas, this paper proposes a classification-based surrogate-assisted evolutionary algorithm (CSEA) for expensive many-objective optimization. Different to the majority of the aforementioned work, the surrogate in CSEA aims to learn the dominance relationship between the candidate solutions and a set of selected reference solutions, instead of the dominance relationship between the candidate solutions. In addition, the degree of uncertainty in prediction is estimated, which plays an important role in surrogate management. The main new contributions of this paper can be summarized as follows.

1) Different to conventional SAEAs using surrogates to approximate the fitness or objective functions, or using surrogates to learn the dominance relationship between the candidate solutions, the surrogate in the proposed CSEA predicts the dominance relationship between the candidate solutions and the selected reference solutions. Its main advantage is that only one single surrogate is needed compared to those SAEAs using surrogates for fitness prediction, and the dominance relationship is more predictable compared to those using surrogates for predicting the dominance relationships between the candidate solutions.

2) A degree of reliability is estimated using a validation data set, which divides the objective space into three regions of uncertainty. The uncertainty information is then used in model management for selecting solutions to be evaluated using the expensive FE method.

The rest of this paper is organized as follows. In Section II, an introduction to surrogate-assisted optimization and radial projection is given. The details of the proposed CSEA for expensive many-objective optimization are described in Section III. Experimental settings and comparisons of CSEA with the state-of-the-art methods on the benchmark problems are presented in Section IV. The conclusions and future work are drawn in Section V.

II. BACKGROUNDS

In this section, we first provide some backgrounds of surrogate-assisted evolutionary optimization, together with a description of the artificial neural network model used in this paper. We then present the details of the radial projection-based grid division, which is used for selecting reference points for separating solutions in candidate solutions into two classes for training the surrogate.

A. Surrogate-Assisted Optimization

The surrogate model in surrogate-assisted optimization typically aims to approximate the objective function or a fitness function of a candidate solution x

$$\widehat{f}(x) = f^*(x) + \xi(x) \tag{2}$$

where f^* is the true value of the objective or fitness value of the solution, $\widehat{f}$ is the approximated value, and ξ is the error function, which reflects the degree of "uncertainty" of the approximation of the surrogate model [36]). The main idea of surrogate-assisted optimization is to replace expensive

FEs with the computationally cheap surrogate, thereby reducing the computation time for solving expensive optimization problems [21].

In recent years, a plenty of multiobjective SAEAs have been shown to be very effective in reducing the needed number of FEs in solving MOPs [20], [49], [50], referring to [41] for a recent review. However, few multiobjective SAEAs have been particularly designed for solving MaOPs. Most recently, a Kriging-based surrogate-assisted reference vector guided evolutionary algorithm was proposed for expensive many-objective optimization [39]. In K-RVEA, the model management strategy focuses on the balance between diversity and convergence by making use of the uncertainty information in the approximated objective values, on the basis of the angel-based penalized distance proposed the original RVEA [13]. K-RVEA was demonstrated to be competitive compared to a few state-of-the-art SAEAs for expensive many-objective optimization.

In multiobjective or many-objective SAEAs, the surrogate can be used to approximate different functions apart from the objective function to accelerate the evolution. Roughly speaking, existing SAEAs can be classified into two different categories according to the target function of the surrogate. In the first category, the fitness function is approximated using single or multiple surrogates. Note that, in this category, the fitness function can be an objective function, or an aggregation function of all objective functions, or even a performance indicator. For example, in ParEGO [37], a single Kriging model is built to approximate an aggregation function at each generation, where the aggregation function is constructed with a weight vector randomly selected from a set of uniform weight vectors. In SMS-EGO, the fitness of an individual is defined as the contribution of this individual to the HV score of the population, and a Kriging model is built to approximate the function of the HV [51]. In K-RVEA, a separate surrogate model is employed for approximating m objective functions, while in MOEA/D-EGO one Kriging model is built for the objective function of each subproblem [21].

In the second category of SAEAs, the surrogate serves as a classifier [48], [52] that divides the candidate solutions into good or bad solutions, e.g., dominated or nondominated solutions. So far, relatively less work has been published that uses surrogate as a classifier, e.g., the classification and Pareto domination-based MOEA (CPS-MOEA) [46] and the MOEA based on decomposition and preselection (MOEA/DP) [53]. In CPS-MOEA, the population is divided into two equal groups, one positive group and one negative group, according to the nondominated sorting. Then a classification and regression tree (or k-nearest neighbor, KNN) is applied to predict the categories of the newly generated offspring for reducing the number of FEs, which is shown to be effective in comparison with the regularity modeling multiobjective evolutionary algorithm [54]. A classification- and regression-assisted differential evolution (DE) algorithm, named CRADE, is proposed for expensive single-objective optimization in [47]. During the environmental selection in CRADE, offspring solutions worse than their parents are discarded according to the classification
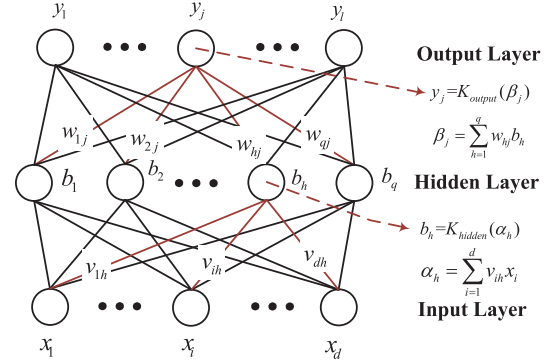

Fig. 1. Three-layer FNN, where the number of neurons in the input, hidden, and output layer is d, q, and l, respectively, with K_{hidden} and K_{output} being the activation functions of the hidden and output neurons.

surrogate, and the fitness of the rest offspring solutions are predicted by the regression surrogate. In [34], classification was used to assist a memetic algorithm in choosing individuals to be refined for solving optimization problems with single equality constraint. In this algorithm, a support vector machine was used to determine whether a solution is close to the feasible region and whether local refinement should be carried out. A significant reduction of computation time was achieved.

B. Feedforward Neural Network

Feedforward neural networks (FNNs) are one type of artificial neural networks in which connections between the neurons do not form a cycle [55], and they have been widely applied in SAEAs [56]. A three-layer FNN involves an input layer, a hidden layer and an output layer, where each layer consists of several neurons, as shown in Fig. 1, where the number of neurons in the input, hidden and output layers is d, q, and l, respectively.

The structure and weights of the FNN are important for its approximation performance. For a d-dimensional input x, its weighted sums of the input nodes α are fanned into the hidden layer. Then the outputs of the hidden neurons b are also weighted by the output weights and summed up to be the final output y, where y is a l-dimensional vector. $K_{\text{hidden}}(\cdot)$ is known as the activation function of the hidden neurons [57], which is typically the sigmoid function or the hyperbolic tangent function, and $K_{\text{output}}(\cdot)$ is the activation function of the output neurons, which is a sigmoid or linear function.

In the FNN shown in Fig. 1, a supervised learning algorithm can be used to train the weight matrices v and w, and the error back-propagation (BP) algorithm is one most popular supervised learning algorithm [58]. In this paper, the Levenberg–Marquardt BP algorithm is adopted to train the FNN surrogate as it has been shown to be one of the fastest BP algorithm [59]. Once the FNN is trained, a prediction result can be obtained when a new input is given to the trained FNN.

C. Radial Projection-Based Selection

In the proposed CSEA, the radial projection-based selection strategy [60] is employed to select a set of reference solutions to construct the classification boundary. RSEA is also adopted

---

Algorithm 1 RadialSelection(P, K)

Output:

P_R (reference population).

1: $[Y, G] \leftarrow \text{RadialGrid}(P, K)$
2: $C \leftarrow 0$
3: $Con(P) \leftarrow || \frac{P - \min P}{\max P - \min P} ||$
4: $P_R \leftarrow \arg_{P_R \in P} \min Con(P)$
5: $Crowd(P_R) \leftarrow 1$
6: **while** $|P_R| < K$ **do**
7: $\quad Q \leftarrow \arg \min Crowd(P)$
8: $\quad Fit(Q, P_R) \leftarrow 0.1 \cdot m \cdot Con(Q) - \min ||Y_Q - Y_{P_R}||$
9: $\quad P_q \leftarrow \arg_{q \in K} \min Fit(Q, P_R)$
10: $\quad P_R \leftarrow P_R \cup \{P_q\}$
11: $\quad P \leftarrow P \backslash \{P_q\}$
12: $\quad Crowd(q) \leftarrow Crowd(q) + 1$
13: **end**

Algorithm 2 RadialGrid(P, K)

Output:

Y (coordinate in radial space), G (rectangle labels).

1: $P_N \leftarrow \frac{P - \min P}{\max P - \min P}$
2: $\theta_i \leftarrow 2\pi(i-1)/m$ /*$i = 1, 2, \ldots, m$*/
3: $W_1 \leftarrow (\cos(\theta_1), \ldots, \cos(\theta_m))$,
$\quad W_2 \leftarrow (\sin(\theta_1), \ldots, \sin(\theta_m))$
4: $Z \leftarrow (P_N\mathbf{1})^{-1}$
5: $Y \leftarrow (P_N W_1 Z, P_N W_2 Z)$
6: $n \leftarrow \lfloor \sqrt{K} \rfloor$
7: $B_l \leftarrow \min Y, B_u \leftarrow \max Y$
8: $G \leftarrow \lfloor n(Y - B_l)/(B_u - B_l) \rfloor$
/*Calculate the label of each projected point*/

---

for selecting individuals in the environmental selection. The details of the radial projection-based selection are presented in Algorithm 1.

First, the radial projection is used to map a set of normalized m-dimensional objective vectors as 2-D points in the radial space, and the region occupied by these points are divided into a sequence of grids (Algorithm 1, line 2). A detailed description of these operations are presented in Algorithm 2, where W is the projection matrix, K the number of required solutions, and $\mathbf{1}$ an $m \times 1$ unit matrix with θ_i being equal to $2\pi(i-1)/m$. Note that the number of required solutions K can be the number of reference solutions or the population size for environmental selection.

Subsequently, K solutions are selected one by one using a combined criterion for convergence and diversity, referring to line 9 in Algorithm 1. In the radial projection-based selection, function Con is used to measure the convergence property of a solution by its Euclidean distance to the ideal point in the objective space, and function $Crowd$ is used to calculate the degree of diversity of a solution using the number of selected solutions in the same grid in the radial space.

III. PROPOSED ALGORITHM

In this paper, a CSEA is proposed for expensive many-objective optimization. A diagram describing the main
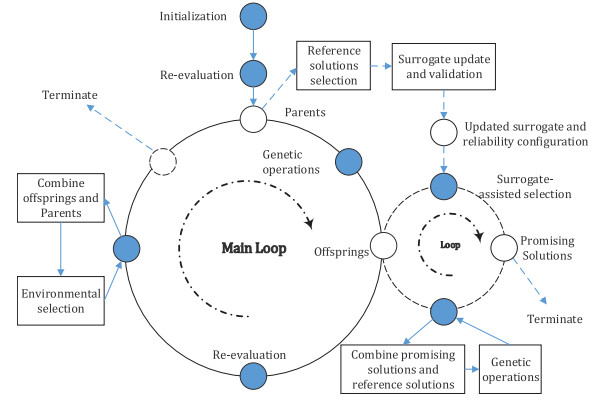

Fig. 2. Diagram of the proposed CSEA consisting of two loops. In the main loop, the solutions are selected according to the real objective function, while in the second loop, solutions are selected according to the surrogate.

framework of the proposed CSEA is given in Fig. 2. From the figure, we can see that CSEA consists of a main loop representing the evolutionary process using the real objective functions, and a second loop in which solutions are selected using the classification surrogate.

The pseudo code of CSEA is presented in Algorithm 3, which can be divided into six main steps as follows.

1) *Initialization (Lines 1–4):* An initial population P with $11d - 1$ solutions is generated using Latin hypercube sampling [61], where d is the number of decision variables. An FNN with H hidden neurons is initialized using randomly generated weights and the activation function is the sigmoid function. In the initialization, the number of solutions to be evaluated using the expensive objective function t equals $11d - 1$, and these solutions are copied to archive *Arc*.

2) *Selection of Reference Solutions (Line 6):* A set of reference solutions P_R is selected from those evaluated by the expensive fitness function using the radial projection-based selection method. The reference solutions are used to construct the classification boundary.

3) *Surrogate Update and Validation (Lines 7–10):* The solutions in the archive are categorized into two classes according to classification boundary. Then these solutions are divided into a training data set (75%) and a test data set (25%). The training set is used for training the FNN and the test set is for cross-validation to estimate the reliability of the classification surrogate.

4) *Surrogate-Assisted Selection (Line 11):* Reproduction operators like crossover and mutation are applied on the parent solutions to create offspring solutions. Promising solutions from the offspring are selected according to the classification results and reliability of classification obtained in step 3.

5) *Environmental Selection (Line 13):* Population P is combined with offspring population Q, then environmental selection is performed to select N solutions from the combined population to be the parent individuals of the next generation (as given in Algorithm 1 with K being set to N).

Algorithm 3 Framework of CSEA

Input:

N (population size), K (number of reference solutions), H (number of hidden neurons), t_{max} (maximum number of FEs), g_{max} (maximum number of surrogate predictions before being updated).

Output:

P (final population).

1: $P \leftarrow$ Initialize the population with $11d-1$ solutions using Latin hypercube sampling method
2: $t \leftarrow 11d - 1$ /* d is the number of decision variables */
3: $net \leftarrow$ Initialize the neural network with H hidden neurons /*m is the number of objectives*/
4: $Arc \leftarrow P$
5: **while** $t \leq t_{max}$ **do**
6:    $P_R \leftarrow$ UpdateReference(P, K)
7:    $C \leftarrow$ Classify(P_R, Arc)
8:    $rr \leftarrow$ The rate of category II solutions in C
9:    $tr \leftarrow \min\{rr, 1 - rr\}$
10:   $[D_{train}, D_{test}] \leftarrow$ DataPartition(Arc, C)
11:   $net \leftarrow$ Train(net, D_{train}, T)
12:   $[p_1, p_2] \leftarrow$ Validation(net, D_{test})
13:   $Q \leftarrow$ SASelection($P, P_R, p_1, p_2, g_{max}, tr$)
14:   $Arc \leftarrow Arc \cup Q$
15:   $P \leftarrow$ RadialSelection($P \cup Q, N$)
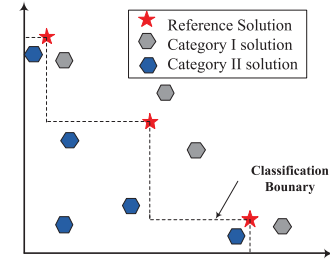16: $t \leftarrow t + |Q|$
17: **end**

Fig. 3. Illustration of the proposed classification criterion using a set of reference solutions. A classification boundary is formed by three reference solutions (denoted by stars), which classifies solutions into category I (filled hexagons) and category II solutions (denoted by unfilled hexagons).

Algorithm 4 Classify(R, P)

Output:

C (categories of solutions in P).

1: **for** $i \leftarrow 1 : |P|$ **do**
2:   $c \leftarrow true$
3:   **for** $j \leftarrow 1 : |R|$ **do**
4:     $c \leftarrow c || (P_i$ Dominate $R_j)$
5:   **end**
6:   **if** c is $false$ **then**
7:     $C_i \leftarrow 0$ /*Category I solution*/
8:   **else**
9:     $C_i \leftarrow 1$ /*Category II solution*/
10:   **end**
11: **end**

6) Repeat steps 2–5 until the maximum number of FEs is reached.

In CSEA, the classification criterion and the surrogate management strategy are crucial for the success of the algorithm. We will discuss them in greater detail in the following sections.

A. Classification Criterion

In CSEA, a classification criterion is proposed to distinguish good solutions from poor ones for selecting promising solutions. This classification criterion is crucial for CSEA as the classification accuracy heavily depends on the classification criterion. In multiobjective optimization, a basic criterion for comparing two solutions is the Pareto dominance relationship [7], [62], [63], which categorizes solutions into dominated solutions and nondominated solutions. Based on this idea, a set of reference solutions are chosen to construct the Pareto dominance boundary, which will divide all solutions into two different categories. An illustrative example is provided in Fig. 3, where three reference points are used to classify nine solutions into two different categories. Solutions on the right side of the nondominance boundary are category I solutions and those on the left side are category II solutions. Algorithm 4 presents the details of the classification criterion in pseudo code.

B. Selection of Reference Solutions

To choose a set of reference solutions to form the classification boundary, a radial space division-based selection strategy,

as described in Algorithm 1, is adopted the proposed CSEA. In this selection strategy, all solutions evaluated using the expensive fitness function are first projected into a 2-D radial space.

The reference solutions play a key role in the performance of CSEA. Fig. 4 provides an example showing the impact of the number of reference solutions on the performance of the proposed algorithm in terms of convergence and diversity balancing. In the figure, there are three possible classification boundaries, $B1$–$B3$, where $B1$ is formed by one reference solution only, $B2$ by seven reference solutions, and $B3$ by three reference solutions. If $B1$ is adopted as the classification boundary, then three solutions are classified as good category solutions, but the diversity of these good category solutions is poor, which may lead to the loss of diversity in the population. By contrast, if $B2$ is used as the boundary, all the six candidate solutions are classified into the same category and some poorly converged solutions like s_2 will be considered as a good solution and convergence may be slowed down if it is selected as a promising solution. However, if $B3$ is adopted as the classification boundary, solutions s_1, s_3, and s_5 are classified as good solutions. We can see that these good solutions are able to achieve a proper balance between convergence and diversity, which can enhance the performance of the proposed algorithm.

In summary, a small number of reference solutions will result in a small number of category II solutions that have
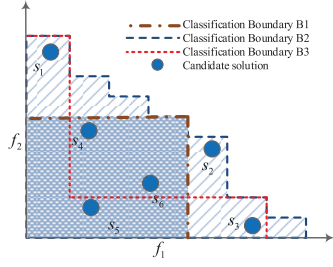
Fig. 4. Example illustrating the impact of the number of reference solutions on the performance of the proposed algorithm, where the classification boundaries B1–B3 are constructed by one, seven, and two reference solutions, respectively.

Algorithm 5 DataPartition(P, C)

Output:

D_{train} (training set), D_{test} (test set).

1: $D_0 \leftarrow$ Find the category I solutions in P
2: $D_1 \leftarrow$ Find the category II solutions in P
3: $i_0 \leftarrow$ Randomly select $\lfloor 3/4|D_0| \rfloor$ solutions from D_0
4: $i_1 \leftarrow$ Randomly select $\lfloor 3/4|D_1| \rfloor$ solutions from D_1
5: $D_{train} \leftarrow \{P_{i_0}, C_{i_0}\}$
6: $D_{test} \leftarrow \{P_{i_1}, C_{i_1}\}$

---

better convergence performance but are poor in terms of diversity. By contrast, a large number of reference solutions will result in a large number of category II solutions which are poorly converged but have good diversity. The choice of the number of reference solutions provides a way of balancing the convergence and diversity of the selected category II solutions.

The relationship between the number of objectives and the number of reference solutions K is also considered. For problems with three to four objectives, CSEA should emphasize on diversity maintenance, so a relative big K is chosen. Instead, the convergence enhancement is crucial for CSEA on problems with more than four objectives due to the high rate of nondominated solutions in the population, and CSEA with a small K could classify most nondominated solutions as category I solutions and ensure the convergence of the category II solutions.

To empirically investigate the impact of the number of reference solutions K on the performance of CSEA for solving problems with different numbers of objectives, CSEA with different settings of parameter K is tested on DTLZ1, DTLZ2, DTLZ5, and DTLZ7 with different objectives. Fig. 5 shows the inverted generational distance (IGD) results achieved by CSEA with 2, 4, 6, 8, 10, 12, and 14 reference solutions on 3-, 4-, 6-, 8-, and 10-objective DTLZ1, respectively. Note that, these four test problems have different properties. DTLZ1 is a multimodal problem with a regular Pareto optimal front, which is difficult for conventional many-objective evolutionary algorithms (MaOEAs) to converge to the PF; DTLZ2 is a relatively simple test problem, which is used to test the diversity maintenance in MaOEA; DTLZ5 has a degenerated PF, which is difficult for MaOEAs with predefined reference information; and DTLZ7 has a discontinuous PF, which is difficult for most MaOEAs to achieve a set of uniformly distributed solutions.

For DTLZ1 with different number of objectives, the IGD values fluctuate slightly as the number of reference solutions increases, and CSEA can achieve its best performance around $K = 6$. As for DTLZ2, the IGD values decrease as the number of reference solutions increases before K reaches 6, but the IGD values rebound when K further increases, indicating that CSEA achieves the best performance when $K = 6$. The change of IGD values on DTLZ5 is similar to that on DTLZ1, and CSEA achieves the best performance when $K = 4$. The IGD values on DTLZ7 remain nearly constant as K changes, nevertheless we can still observe that CSEA with K being around

8 can achieve its best performance. All in all, CSEA with $K = 6$ can achieve stable performance in terms of both convergence enhancement and diversity maintenance on these four test problems. Accordingly, the number of reference solutions will be fixed to six for all the test instances.

C. Surrogate Management

In the proposed CSEA, the surrogate management method, as shown in Algorithm 3, consists of the following four parts.

1) *Surrogate initialization* which initializes the parameters and structure of the FNN (line 3).
2) *Surrogate update* which updates the weights of the FNN with the training data set (line 11).
3) *Surrogate validation* where the trained network is verified by cross-validation using the test data and the test errors are calculated (line 12).
4) *Surrogate-assisted selection* which aims to select promising solutions to be evaluated using the expensive fitness function. The selection is based on the predicted performance as well as the reliability (lines 12 and 13).

Note that before surrogate management is completed, all solutions evaluated by the expensive fitness function are divided into training and test sets. This data partition procedure aims to alleviate the negative impact of possible categories imbalance in the data and improve the classification performance of the FNN. The solutions are randomly divided into two sets where |training set| : |test set| is set to 3:1 in this paper, and the percentages of category I solutions in both sets are the same. The details of the data partition procedure is presented in Algorithm 5.

1) Surrogate Initialization: Three main components of the FNN, including the network structure, the weights and the activation function, need to be initialized. In this paper, a 3-layer FNN is constructed with d-dimensional input, 1-D output and H hidden layer neurons, where d is the number of decision variables. All weights are initialized with a random number in [0, 1]. Finally, the sigmoid function is used as the activation function in the hidden and output layer neurons [64]

$$S(x) = \frac{1}{1 + e^{-\lambda x}} \qquad (3)$$

where x is the input variable and parameter λ is set to one in this paper.

2) Surrogate Update: The training set data obtained by Algorithm 5 is used for updating the weights of the FNN. As previously mentioned, Levenberg–Marquardt BP method is adopted in this paper. The details of the Levenberg–Marquardt
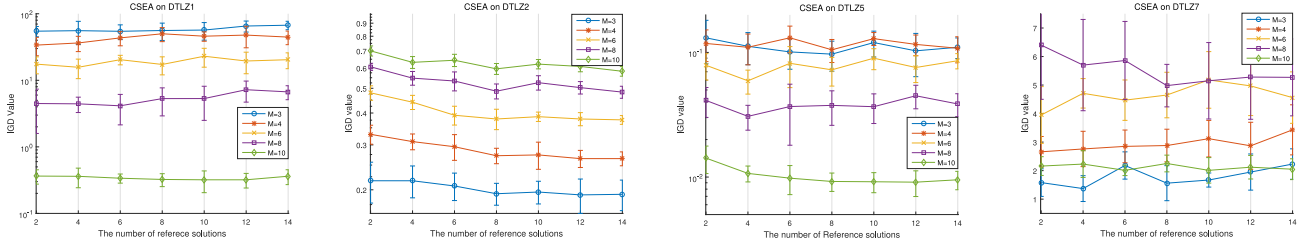
Fig. 5.   IGD results achieved by CSEA With 2, 4, 6, 8, 10, 12, and 14 reference solutions on 3-, 4-, 6-, 8-, and 10-objective DTLZ1, DTLZ2, DTLZ5, and DTLZ7, respectively.

BP method can be found in [65]. In training the FNN update, the maximum number of training epochs T is set to 500 for all the test instances. In addition, an archive is used to store all solutions evaluated using the expensive fitness function.

3) Surrogate Validation: Cross validation is performed for calculating the error on the test data as a measure for estimating the prediction uncertainty of the FNN. It is important that the predicted classification is reliable as wrong predictions will mislead the evolutionary search. Instead of calculating the error on the entire test data, the error of the category I solutions (denoted as p_1) and that of the category II solutions (denoted as p_2) in the test set (denoted as D_{test}) are calculated separately. Suppose Q_c is the set of category c solutions, and their predicted categories are $C_{p_1}, C_{p_2}, \ldots, C_{p_{|Q|}}$. The mean absolute error (MAE) of the prediction is defined as follows:

$$\text{MAE} = \frac{\sum_{i=1}^{|Q_c|} \text{abs}(c - C_{p_i})}{|Q_c|} \tag{4}$$

where $|Q_c|$ denotes the number of solutions in Q_c and abs(∗) denotes the absolute value of ∗. MAE is adopted as the test error of category c solutions. Then the values of p_1 and p_2 can be worked out using (4), respectively. These two values indicate the reliability of the surrogate classifier, and the use of these two errors will be further discussed in the following section.

4) Surrogate-Assisted Solution Selection: The errors on the test data, p_1 and p_2 are jointly used to estimate the reliability of the prediction of the FNN. Fig. 6 presents the reliability configuration formed by the test errors based on the relationship between the uncertainty of the FNN and its test errors. Once point (p_1, p_2) is located in a region of the reliability configuration, the uncertainty of the FNN can be estimated.

In reliability configuration, region R_1 represents the region in which all category II solutions are reliably predicted and therefore the FNN is able to select the right category II solutions to be evaluated using the expensive fitness function. Actually, this region consists of two parts.
1) For $p_2 <$ tr, the classifier is able to predict category II solutions.
2) For $p_1 <$ tr AND $p_2 < (1-$tr$)$, the prediction of category I solutions is reliable and the predicted category I solutions can be discarded, hence the boundary of p_2 can be relaxed to sample some promising category II solutions to benefit from the uncertainty. In this case, if a poor classifier that always predicts solutions to belong to category II, some category I solutions may be selected. This
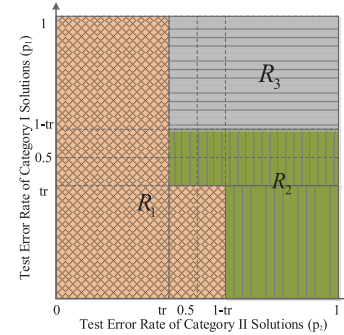

Fig. 6.   Reliability configuration of the FNN constructed by the test errors p_1 and p_2 for estimating the uncertainty of the prediction.

problem might be mitigated by using a stricter threshold (line 11 in Algorithm 6) to select some very certain category II solutions.

Note that a large R_1 does not necessarily mean that a large number of candidate solutions will be evaluated using the expensive objective functions. Meanwhile, the predicted category I solutions will not be evaluated using the expensive objective functions, which helps save a lot of real function evaluations.

Region R_2 represents the region in which the FNN can correctly predict neither category I solutions nor category II solution. Hence, none of these predicted solutions will be evaluated using the real objective function, implying that the FNN fails to provide useful information for predicting the dominance relationship of solutions.

Region R_3 represents the region in which category I solutions will be very likely to be predicted as category II solutions. Therefore, solutions predicted to be category I in R_3 should be chosen to be evaluated using the expensive real fitness function, as they are very likely category II solutions.

The reliability configuration is formed using a threshold far from 0.5, ensuring that the classification decision is reliable enough. A rescaling method [66] is adopted to determine the threshold value so that the influence of different classes on the learning process is reduced. Further discussion about the threshold will be given in the following section. In this paper, the threshold tr is set to $0.5 \times \min \{rr, 1 - rr\}$, where rr is given in (6).

A surrogate-assisted selection strategy is applied in CSEA to select potentially well converged solutions (category II solutions) from the offspring solutions, as presented in Algorithm 6

Algorithm 6 SASelection($P, P_R, p_1, p_2, g_{max}$, tr)

Output:

    Q (offsprings).

1: $Q \leftarrow$ Variation($P \cup P_R$)
2: $L \leftarrow$ Prediction(net, Q)
3: $i \leftarrow 0$
4: **if** $p_2 < tr$ OR ($p_1 < tr$ AND $p_2 < 1 - tr$) **then**
5:     **while** $i < g_{max}$ **do**
6:         $H \leftarrow$ Select $|P_R|$ solutions from Q with the largest L values
7:         $Q \leftarrow$ Variation($H \cup P_R$)
8:         $L \leftarrow$ Prediction(net, Q)
9:         $i \leftarrow i + |Q|$
10:     **end**
11:     $Q \leftarrow$ Select solutions from Q with $L > 0.9$
12: **else if** $p_1 > 1 - tr$ AND $p_2 > tr$ **then**
13:     **while** $i < g_{max}$ **do**
14:         $H \leftarrow$ Select $|P_R|$ solutions from Q with the smallest L values
15:         $Q \leftarrow$ Variation($H \cup P_R$)
16:         $L \leftarrow$ Prediction(net, Q)
17:         $i \leftarrow i + |Q|$
18:     **end**
19:     $Q \leftarrow$ Select solutions from Q with $L < 0.1$
20: **else**
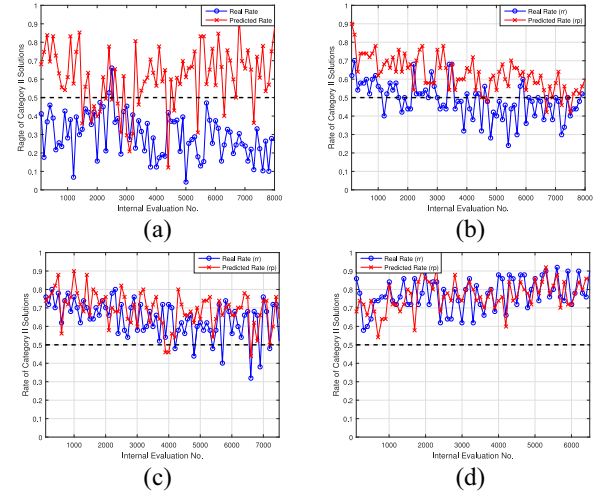21: $Q \leftarrow \emptyset$
22: **end**

Fig. 7. Variations of the rate of category II solutions in the current population predicted by the FNN in CSEAs with and without data partition rp and that of the real rate rr on DTLZ1 with different numbers of objectives. (a) CSEA without data partition on three-objective DTLZ1. (b) CSEA with data partition on three-objective DTLZ1. (c) CSEA with data partition on five-objective DTLZ1. (d) CSEA with data partition on ten-objective DTLZ1

according to the reliability configuration. In this strategy, a set of offspring solutions is first generated by reproduction operators, e.g., the simulated binary crossover [67] and polynomial mutation [68]. Then the location of the test errors p_1 and p_2 on the reliability configuration is determined. If a solution is located in region R_1, the solutions predicted to be category II are used to generate offspring solutions until the termination condition is satisfied; if the solution is located in region R_3, the solutions predicted to be category I are used to generate offspring solutions until the termination condition is satisfied; otherwise, no solution will be selected. During the surrogate-assisted selection strategy, a maximum number of g_{max} solutions are predicted by the trained FNN, and this number indicates the frequency that the surrogate has been used before the surrogate is updated. Note that only a few candidate solutions are selected to be evaluated using the expensive fitness function.

D. Surrogate Accuracy Analysis

In this section, the impact of the imbalance of different categories in the training data set on the performance of the proposed CSEA and the accuracy of the FNN will be investigated. First, we introduce a new metric for assessing the accuracy of the FNN. Assuming that the set of candidate solutions is Q (as shown in Algorithm 6 step 2), and the predicted categories of solutions in Q are $\{Cp_1, Cp_2, \ldots, Cp_{|Q|}\}$ with their true categories (classified according to the objective values calculated using the expensive objective functions)

are $\{Cr_1, Cr_2, \ldots, Cr_{|Q|}\}$. The predicted rate of category II solutions rp is defined as

$$rp = \frac{\sum_{i=1}^{|Q|} (Cp_i \text{ is category II})}{|Q|}. \tag{5}$$

The real rate of category II solutions rr is defined as

$$rr = \frac{\sum_{i=1}^{|Q|} (Cr_i \text{ is category II})}{|Q|}. \tag{6}$$

Note that a smaller difference between rp and rr indicates a better accuracy of the FNN.

The CSEA without the data partition is tested on DTLZ1 with three objectives, and the variations of rp and rr are plotted in Fig. 7(a). Then the variations of rp obtained by FNN with the data partition procedure and that of the real rate rr are plotted in Fig. 7(b). It can be observed from these two figures that the CSEA without data partition has failed to predict the categories of the most candidate solutions as the rate of category II solutions obtained by FNN is quite different from the real rate. However, CSEA with the data partition procedure performs much better, and the FNN has achieved good prediction results, where the variations of rp obtained by the FNN and that of rr are similar.

The rp obtained by CSEA with data partition is compared with the corresponding rr on DTLZ1 with five and ten objectives, respectively, which aims to investigate the influence of class imbalance in the data during the evolution on the prediction accuracy, as the number of objectives increases. The variations of rp and rr on DTLZ1 with five and ten objectives are shown in Fig. 7(c) and (d), respectively. In these figures, we can see that rp and rr are over 50%, and these two rates increase as the number of objectives increases, since the rate of nondominated solutions increases as the number of objectives increases and more nondominated solutions are predicted as category II solutions. Meanwhile, the tendency of

TABLE I
RESULTS OF CSEA- AND CSEA ON NINE TEST INSTANCES.
THE BEST RESULT IN EACH ROW IS HIGHLIGHTED

Problem	Obj.	CSEA$^-$	CSEA
DTLZ1	3	9.71e+1(2.30e+1)−	4.36e+1(8.82e+0)
	6	2.58e+1(1.11e+1)−	1.32e+1(4.43e+0)
	10	3.87e-1(7.15e-2)−	2.86e-1(4.37e-2)
DTLZ3	3	2.47e+2(5.91e+1)−	1.09e+2(2.73e+1)
	6	1.05e+2(2.60e+1)−	4.78e+1(1.38e+1)
	10	1.26e+0(2.49e-1)−	1.00e+0(9.63e-2)
DTLZ5	3	2.21e-1(3.21e-2)−	8.07e-2(2.22e-2)
	6	1.13e-1(1.90e-2)−	6.53e-2(2.20e-2)
	10	8.36e-2(2.96e-2)−	1.00e-2(9.22e-4)

the variation of rp matches the variation of rr well, and the difference between them becomes smaller and smaller as the number of objectives increases, which is due to the fact that the prediction accuracy increases as the number of objective increases. In summary, the class imbalance in data becomes worse as the number of objectives increases (rr increases), and the FNN with data partition can achieve a satisfactory prediction accuracy also on MaOPs.

E. Effectiveness of the Surrogate

To investigate the effectiveness of the surrogate in the proposed CSEA, we compare the CSEA with its variant without using the surrogate, denoted by CSEA$^-$, on DTLZ1, DTLZ3, and DTLZ5 with 3, 6, and 10 objectives, respectively. Note that CSEA$^-$ is similar to CSEA except that the second loop in Fig. 2 using the surrogate is removed. The experimental results are presented in Table I.

It can be observed from the table that CSEA significantly outperforms CSEA$^-$ on all the test instances, confirming the effectiveness and importance of using the surrogate in CSEA. More results are presented in the supplementary materials.

IV. EMPIRICAL STUDIES

In this section, we examine the performance of CSEA by empirically comparing it with a few state-of-the-art algorithms, namely NSGA-III [11], ParEGO [37], CPS-MOEA [46], MOEA/D-EGO [21], and K-RVEA [39] (all the compared algorithms are implemented in PlatEMO [69]). Note that ParEGO, MOEA/D-EGO, and K-RVEA are SAEAs for solving many-objective optimization using Kriging models to approximate the objective values and the aggregation function values, respectively, while CPS-MOEA [46] is also a classification-based SAEA using k-nearest neighbor method to predict the quality of a solution. NSGA-III is not an SAEA *per se*, but is also adopted for comparison to illustrate the competitiveness of the SAEAs in reducing the number of FEs for expensive many-objective optimization. This set of the experiments are conducted on 35 test instances taken from test suite DTLZ [70] with 3, 4, 6, 8, and 10 objectives, respectively. We also compared the algorithms on the WFG test suite [71], MaF test suite [72], and a car cab design problem [11]. The comparative results are presented in the supplementary materials.

In the experiments, the number of decision variables is set to ten (or an integer closest to ten for WFG2 and WFG3). The Wilcoxon rank sum test is adopted to compare the results achieved by CSEA and other algorithms under comparison at a significance level of 0.05 over 30 independent runs. Symbols "+" and "−" denote that the compared algorithm performs significantly better and significantly worse than CSEA according to the Wilcoxon rank sum test, while "≈" indicates there is no statistically significant difference between the two compared algorithms.

A. Performance Indicator

For comparisons, the IGD is adopted for evaluating the performance of the compared algorithms. IGD is believed to be able to account for both convergence and diversity of the nondominated solutions, and a smaller IGD value indicates better performance of the MOEA. Since IGD requires a reference set, which should be evenly distributed on the Pareto optimal front of test problems. For all test problems, the closest integer to 5000 is used as the number of reference points for IGD calculation, as an exact number of 5000 is impossible to be set for all test instance. In this paper, the HV [73] is not used to assess the performance of the compared algorithms as the obtained solutions of expensive MaOPs are not yet well converged due to the smaller number of FEs allowed, often leading to a zero HV. In the supplementary materials, the HV metric is used to assess the performance of different algorithms when the obtained results are well converged.

Suppose that P^* is a set of evenly distributed reference points on the PF and Ω is the set of achieved nondominated solutions, IGD is defined as follows:

$$\text{IGD}(P^*, \Omega) = \frac{\sum_{x \in P^*} \text{dis}(x, \Omega)}{|P^*|} \qquad (7)$$

where $\text{dis}(x, \Omega)$ is the minimum Euclidean distance between x and points in Ω and $|P^*|$ the number of elements in P^*.

B. Experimental Settings

For a fair comparison, we adopt the recommended parameter settings for the compared algorithms that have achieved the best performance reported in the literature and the details are given below.

1) Reproduction Operators: In this paper, simulated binary crossover [67] and polynomial mutation [68] are adopted in the compared algorithms for offspring generation except for CPS-MOEA. The distribution index of crossover is set to $n_c = 20$ and the distribution index of mutation is set to $n_m = 20$, as recommended in [74]. The crossover probability p_c is set to 1.0 and the mutation probability p_m is set to $1/d$, where d is the number of decision variables. In CPS-MOEA and MOEA/D-EGO, DE operators [75] are used for offspring generation, where the control parameters are set to $F = 0.5$, $p_m = 1/d$, and $\eta = 20$ as recommended in [46].

2) Population Size: The population size of CPS-MOEA, ParEGO and CSEA is set to 50. For NSGA-III, MOEA/D-EGO and K-RVEA, a two-layer reference vector generation strategy is applied to generate reference (weight) vectors on

Obj.	Parameter(p_1, p_2)	Population size (N)
3	(8,0)	45
4	(4,0)	35
6	(2,2)	42
8	(2,1)	44
10	(2,0)	55

both outer boundaries and inside layers of the Pareto optimal fronts, where p_1 and p_2 are parameters controlling the numbers of weight vectors along the boundary of the Pareto front and inside it, respectively [11]. The settings of these two parameters for generating a population size nearest to 50 for different test instances are list in Table II. Note that the population size is set to an extremely small number as only hundreds of FEs are allowed and a smaller population size will make it possible to run a relative larger number of generations.

3) Termination Condition: For all the test instances, the maximum number of FEs is adopted as the termination condition, which is set to 300 as the problems involved are assumed to be computationally expensive.

4) Specific Parameter Setting in Each Algorithm: For ParEGO, the number of weight vectors is set to 11 for bi-objective problems and 15 for tri-objective problems, and the maximum number of surrogate-assisted fitness approximation before the surrogate update is set to 200 000 as recommended in [37]. For CPS-MOEA, k-nearest neighbor classification approach is used for solutions classification, where k is set to five as recommended in [46]. For MOEA/D-EGO, the number of surrogate-assisted FEs before updating the models is set to $20 \times (11d - 1)$, and the other parameter settings are the same as in [21].[1] For K-RVEA, parameter δ is set to $0.05N$ with N being the population size, and the number of generations w_{max} before updating the Kriging models is set to 20 as recommended in [39]. All the parameters are the same as recommended in [13]. Regarding the settings of CSEA, the number of surrogate-assisted prediction before updating the models is equal to that in K-RVEA and MOEA/D-EGO, the maximum epochs for training the FNN T is set to 500 and the training is terminated once the change of the weights is smaller than 0.001, the number of hidden neurons H is set to 10, and the number of reference solutions is set to 6 for all the test instances.

C. Results on DTLZ Problems

The statistical results of the IGD values achieved by the six algorithms under comparison over 20 independent runs on DTLZ1 to DTLZ7 are summarized in Table III, where the best results are highlighted. It can be observed that CSEA has

[1]The implementation of MOEA/D-EGO is adapted from the codes available at http://www.cs.cityu.edu.hk/~qzhang/publications.html. However, we replace the DE algorithm with the method in a Kriging toolbox called DACE [76] to optimize the hyperparameters, since it is time-consuming to optimize the hyperparameters in Kriging using DE.

achieved the best results on the 35 test instances (mostly on DTLZ1 to DTLZ5), followed by MOEA/D-EGO (mostly on DTLZ6 and DTLZ7).

DTLZ1 and DTLZ3 are difficult due to their multimodal landscapes, which means that it is difficult to obtain a set of well converged solutions using a small number of FEs. It is obvious that CSEA has achieved the best converged solutions, followed by MOEA/D-EGO, K-RVEA, ParEGO, CPS-MOEA, and NSGA-III. The final nondominated solutions achieved by the compared algorithms on three-objective DTLZ1 in the run associated with the median IGD values are plotted in Fig. 8. It can be concluded that the proposed CSEA is effective in accelerating convergence on expensive many-objective optimization.

The results on DTLZ2 and DTLZ4 show good convergence but poor diversity. It can be seen from the table that, CSEA and MOEA/D-EGO have achieved the best results, and the other four compared algorithms exhibit similar performance. This might be attributed to the radial space division-based strategy in CSEA for updating the reference solutions and the cluster-based selection strategy in MOEA/D-EGO for selecting the candidate solutions for expensive FE.

The PF of DTLZ5 and DTLZ6 are degenerated curves, therefore, it is difficult to achieve a set of diverse and well converged solutions. It can be observed that CSEA together with MOEA/D-EGO have achieved all the best results on these problems. The final nondominated solutions achieved by the compared algorithms on ten-objective DTLZ5 in the run associated with the median IGD values are presented in Fig. 9. It can be seen that CSEA has achieved a set of well converged and evenly distributed solutions, MOEA/D-EGO has achieved a set of well distributed but less converged solutions, K-RVEA has achieved a set of well distributed but not well converged solutions. By contrast, NSGA-III, ParEGO, and CPS-MOEA fail to achieve a set of well converged solutions.

The PF of DTLZ7 is discontinuous, and therefore diversity maintenance is challenging. It can be observed that MOEA/D-EGO performs best on the five test instances, followed by K-RVEA, CPS-MOEA, CSEA, NSGA-III, and ParEGO. Although CSEA has not achieved the best results on these five test instances, it has been shown competitive compared with other five algorithms.

D. Influence of the Number of Hidden Neurons

This part of the empirical studies aims to verify the influence of the number of hidden neurons H on the performance of CSEA in expensive many-objective optimization. Since an FNN is adopted as the surrogate in the proposed CSEA, the complexity of the surrogate and its approximation ability increase as the number of hidden neurons of the FNN increases. So is the risk of overfitting. Hence, an appropriate number of hidden neurons is important for achieving acceptable prediction. In the study, $0.5d$, d, $1.5d$, and $2d$ hidden neurons are tested on DTLZ1 with 3, 5, and 10 objectives, respectively, where d is the number of decision variables. The experimental results are listed in Table IV. It can be observed that CSEA with $2d$ hidden neurons has achieved the best

TABLE III
STATISTIC RESULTS OF THE FIVE COMPARED ALGORITHMS ON 35 TEST INSTANCES. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED

Problem	Obj.	NSGA-III	ParEGO	CPS-MOEA	K-RVE	MOEA/D-EGO	CSEA
DTLZ1	3	7.87e+1(1.49e+1)−	1.08e+2(1.95e+1)−	7.79e+1(1.21e+1)−	1.25e+2(2.35e+1)−	9.64e+1(1.94e+1)−	4.36e+1(8.82e+0)
	4	5.96e+1(1.39e+1)−	8.24e+1(1.97e+1)−	5.68e+1(1.24e+1)−	8.08e+1(2.01e+1)−	6.39e+1(6.63e+0)−	3.18e+1(9.29e+0)
	6	2.97e+1(1.40e+1)−	4.15e+1(8.11e+0)−	3.24e+1(8.59e+0)−	4.45e+1(1.06e+1)−	2.71e+1(7.52e+0)−	1.32e+1(4.32e+0)
	8	9.65e+0(3.71e+0)−	1.36e+1(4.69e+0)−	1.17e+1(2.89e+0)−	1.51e+1(4.33e+0)−	1.05e+1(3.21e+0)−	4.36e+0(2.25e+0)
	10	7.12e-1(4.74e-1)−	5.44e-1(1.75e-1)−	5.16e-1(3.83e-1)−	4.71e-1(1.96e-1)−	3.88e-1(1.13e-1)−	2.86e-1(4.37e-2)
DTLZ2	3	2.76e-1(3.57e-2)−	3.42e-1(2.17e-2)−	2.91e-1(2.97e-2)−	2.92e-1(2.77e-2)−	3.12e-1(2.59e-2)−	1.89e-1(1.13e-2)
	4	3.47e-1(1.64e-2)−	3.86e-1(2.59e-2)−	4.51e-1(4.06e-2)−	3.69e-1(3.41e-2)−	3.48e-1(1.92e-2)−	2.63e-1(1.98e-2)
	6	5.19e-1(2.28e-2)−	4.96e-1(2.49e-2)−	6.34e-1(2.80e-2)−	4.49e-1(5.30e-2)−	4.48e-1(1.30e-2)−	3.93e-1(3.28e-2)
	8	6.33e-1(3.47e-2)−	5.55e-1(2.64e-2)≈	6.70e-1(2.77e-2)−	6.11e-1(2.85e-2)−	5.32e-1(1.30e-2)+	5.47e-1(1.75e-2)
	10	7.44e-1(5.91e-2)−	6.18e-1(2.72e-2)+	6.85e-1(2.77e-2)−	6.71e-1(3.92e-2)−	5.14e-1(1.67e-2)+	6.25e-1(2.47e-2)
DTLZ3	3	2.55e+2(4.34e+1)−	2.63e+2(5.27e+1)−	2.13e+2(4.49e+1)−	3.57e+2(4.36e+1)−	2.06e+2(2.06e+1)−	1.09e+2(2.73e+1)
	4	1.65e+2(6.51e+1)−	1.96e+2(3.84e+1)−	1.59e+2(2.94e+1)−	2.57e+2(5.26e+1)−	1.83e+2(6.18e+1)−	8.69e+1(2.19e+0)
	6	1.02e+2(3.56e+1)−	1.14e+2(2.72e+1)−	1.19e+2(4.23e+1)−	1.28e+2(4.13e+1)−	9.17e+1(1.63e+1)−	4.78e+1(1.38e+0)
	8	2.35e+1(1.33e+1)−	3.55e+1(1.48e+1)−	5.02e+1(1.29e+1)−	3.22e+1(1.98e+1)−	3.63e+1(8.15e+0)−	7.76e+0(4.37e+0)
	10	1.56e+0(1.13e+0)−	1.68e+0(5.61e-1)−	7.29e+0(6.48e+0)−	1.84e+0(5.65e-1)−	1.26e+0(3.36e-1)−	1.01e-1(9.63e-2)
DTLZ4	3	6.44e-1(1.02e-1)−	6.81e-1(7.26e-2)−	5.85e-1(5.94e-2)−	5.26e-1(1.51e-1)−	6.50e-1(2.74e-2)−	3.89e-1(5.76e-2)
	4	7.00e-1(1.68e-1)−	7.24e-1(5.21e-2)−	6.23e-1(2.75e-2)−	6.43e-1(1.08e-1)−	6.58e-1(5.26e-2)−	4.13e-1(5.21e-2)
	6	7.61e-1(1.25e-1)−	6.88e-1(3.33e-2)−	6.59e-1(2.46e-2)−	7.18e-1(8.64e-2)−	6.64e-1(3.21e-2)−	5.48e-1(4.46e-2)
	8	7.32e-1(3.55e-2)−	6.69e-1(1.90e-2)−	6.61e-1(1.82e-2)−	7.43e-1(3.64e-2)−	6.34e-1(1.82e-2)−	6.17e-1(3.19e-2)
	10	7.84e-1(3.31e-2)−	6.12e-1(1.36e-2)+	6.46e-1(7.00e-3)−	7.12e-1(3.77e-2)−	6.50e-1(0.00e+0)−	6.39e-1(1.41e-2)
DTLZ5	3	1.82e-1(3.94e-2)−	2.84e-1(2.20e-2)−	2.11e-1(2.81e-2)−	2.81e-1(4.96e-2)−	2.48e-1(1.92e-2)−	8.07e-2(2.22e-2)
	4	1.91e-1(3.28e-2)−	2.35e-1(2.21e-2)−	2.12e-1(5.22e-2)−	2.21e-1(2.52e-2)−	1.94e-1(8.94e-3)−	9.87e-2(1.73e-2)
	6	1.60e-1(2.87e-2)−	1.52e-1(2.14e-2)−	1.89e-1(2.89e-2)−	1.42e-1(2.59e-2)−	1.44e-1(1.34e-2)−	6.53e-2(2.20e-2)
	8	1.29e-1(3.28e-2)−	7.09e-2(8.92e-3)−	1.25e-1(2.88e-2)−	7.78e-2(1.09e-2)−	7.00e-2(1.22e-2)−	3.27e-2(4.58e-3)
	10	1.59e-1(1.73e-2)−	1.74e-2(1.91e-3)−	5.79e-2(1.20e-2)−	2.00e-2(0.00e+0)−	2.00e-2(0.00e+0)−	1.00e-2(9.22e-4)
DTLZ6	3	5.96e+0(3.22e-1)−	6.33e+0(1.83e-1)−	3.39e+0(8.41e-1)+	6.01e+0(4.48e-1)−	2.43e+0(5.36e-1)+	4.13e+0(8.36e-1)
	4	5.62e+0(4.30e-1)−	5.47e+0(1.50e-1)−	3.21e+0(6.92e-1)−	5.16e+0(5.19e-1)−	1.87e+0(8.81e-1)+	3.01e+0(6.19e-1)
	6	3.85e+0(4.10e-1)−	3.63e+0(2.74e-1)−	2.64e+0(4.76e-1)−	3.60e+0(4.11e-1)−	1.19e+0(2.76e-1)+	2.41e+0(6.28e-1)
	8	2.28e+0(3.40e-1)−	2.02e+0(2.36e-1)−	1.18e+0(4.80e-1)−	1.91e+0(2.51e-1)−	6.72e-1(2.26e-1)+	1.24e+0(4.20e-1)
	10	8.44e-1(2.23e-1)−	6.21e-1(1.11e-1)−	1.82e-1(1.11e-1)−	4.58e-1(2.27e-1)−	1.82e-1(8.20e-2)−	7.53e-2(3.42e-2)
DTLZ7	3	3.97e+0(1.28e+0)−	5.78e+0(8.43e-1)−	4.15e+0(1.46e+0)−	2.23e+0(9.79e-1)−	2.30e+0(8.60e-2)+	1.39e+0(6.83e-1)
	4	5.38e+0(1.21e+0)−	7.79e+0(1.19e+0)−	4.19e+0(1.63e+0)−	2.75e+0(7.48e-1)−	5.62e-1(4.32e-2)+	2.52e+0(4.86e-1)
	6	8.53e+0(2.63e+0)−	8.73e+0(2.18e+0)−	3.19e+0(1.49e+0)+	4.02e+0(1.25e+0)+	8.46e-1(7.67e-2)+	4.45e+0(6.94e-1)
	8	9.39e+0(4.58e+0)−	6.67e+0(2.70e+0)−	2.25e+0(9.94e-1)+	3.83e+0(2.64e+0)+	9.88e-1(4.09e-2)+	4.41e+0(3.15e-1)
	10	9.70e+0(7.95e+0)−	1.68e+0(2.21e-1)+	1.77e+0(2.87e-1)≈	1.59e+0(2.46e+1)+	1.19e+0(3.21e-2)+	1.85e+0(2.29e-1)
+/−/≈		0/35/0	3/31/1	3/31/1	3/32/0	11/24/0	

'+', '−' and '≈' denote that the result is statistically significantly better, worse and comparable to that obtained by CSEA, respectively.
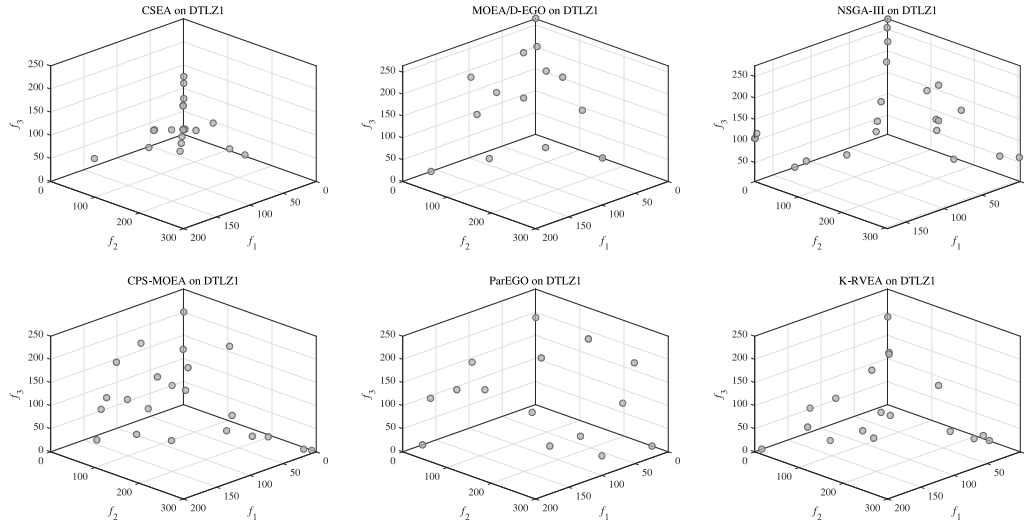

Fig. 8.   Nondominated front obtained by each algorithm on three-objective DTLZ1 in the run associated with the median IGD value.

results on DTLZ1 with three and five objectives, which might be due to the fact that the FNN with a large number of hidden neurons has better approximation capability. On the other hand, CSEA with $0.5d$ hidden neurons has achieved the best result on ten-objective DTLZ1 and the second best on the other two test instances, as a simple surrogate is capable of achieving good approximation result while a complex surrogate may suffer from overfitting. Hence, the number of hidden neurons is set to $0.5d$ as a general setting for all test instances, which can achieve good approximation ability while reducing

the risk of overfitting. This is intuitive as the total number of training data is limited to $11d - 1$.

E. Problems With More Than Ten Decision Variables

Most existing SAEAs have been tested only on optimization problems with up to 30 decision variables [77], and those using Kriging models [21], [39] have typically been tested on problems with up to ten decision variables. This is mainly due to the fact that the computation time for training
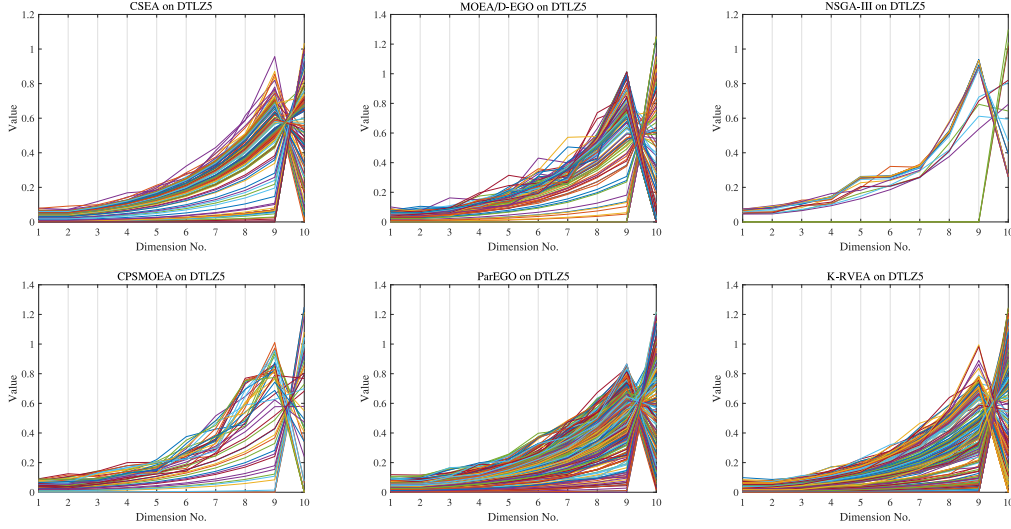
Fig. 9.   Nondominated front obtained by each algorithm on ten-objective DTLZ5 in the run associated with the median IGD value.
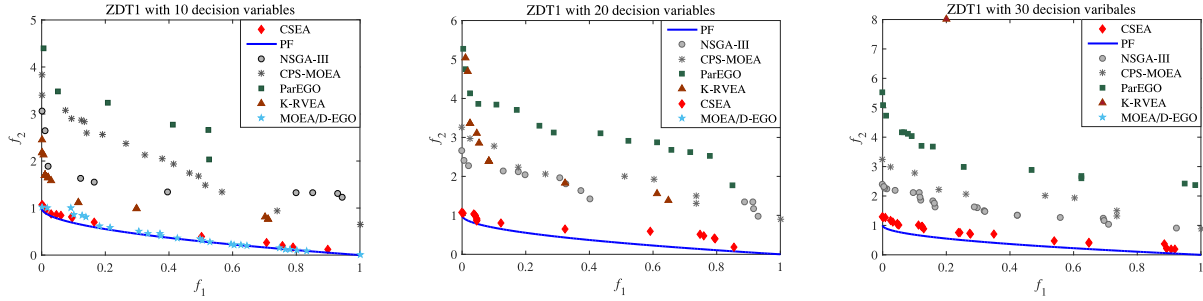

Fig. 10.   Nondominated front obtained by each algorithm on bi-objective ZDT1 with 10, 20, and 30 decision variables in the run associated with the median IGD value, respectively.

TABLE IV
STATISTIC RESULTS OF CSEA WITH DIFFERENT NUMBERS OF HIDDEN LAYER NEURONS ON DTLZ1. THE BEST RESULT IN EACH TEST INSTANCE IS HIGHLIGHTED

H	Obj.		
	3	5	10
$0.5d$	3.82e+1(7.57e+0)	2.11e+1(5.42e+0)	2.31e-1(1.89e-2)
d	4.66e+1(6.40e+0)	2.70e+1(6.37e+0)	2.79e-1(4.35e-2)
$1.5d$	5.26e+1(1.29e+1)	2.15e+1(1.01e+1)	2.75e-1(4.50e-2)
$2d$	3.29e+1(5.20e+0)	2.08+1(3.93e+0)	2.73e-1(3.55e-2)

Kriging models can become prohibitive when the amount of data for training the surrogate increases [78]. As a result, locating the global optimum or global Pareto front of high-dimensional multimodal problems becomes hardly tractable as an accurate approximation of the fitness landscape will usually require a very large number of FEs [25]. Only very recently, some work has been reported on addressing high-dimensional expensive single-objective problems with up to 100 decision variables [35], and large-scale single-objective optimization problems with up to 500 decision variables [79].

To investigate the performance of the proposed CSEA on expensive optimization problems with more than ten decision variables, the performance of CSEA and other four algorithms, namely NSGA-III, CPS-MOEA, ParEGO, MOEA/D-EGO,

and K-RVEA, is examined on ZDT1 [80] with 10, 20, and 30 decision variables.

In this set of experiments, the maximum numbers of FEs are set to 300, 600, and 900 for test instances with 10, 20, and 30 decision variables, respectively. The final nondominated solutions achieved by each algorithm on ZDT1 with 10, 20, and 30 decision variables in the run associated with the median IGD value are shown in Fig. 10. Note that, MOEA/D-EGO has achieved the best results on ZDT1 with ten decision variables, but the nondominated solutions achieved by MOEA/D-EGO on ZDT1 with 20 and 30 decision variables are far from the Pareto optimal front and the solutions achieved by other algorithms, hence these solutions are not included.

It can be observed from these three figures that CSEA has achieved the best performance on these three test instances compared with other five state-of-art algorithms, which further demonstrates the efficiency of CSEA on problems with up to 30 decision variables.

F. Runtime Comparison

For conventional SAEAs using Kriging models, the computation time for training the surrogates varies a lot depending on the number of training samples, which may become prohibitively large if a large number of training samples are used.
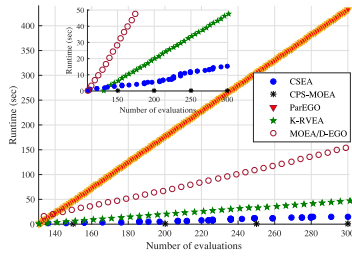
Fig. 11. Runtime over the number of evaluations in CSEA, CPS-MOEA, ParEGO, MOEA/D-EGO, and K-RVEA on three-objective DTLZ2.

For the proposed CSEA, an FNN is adopted as the surrogate, whose training time is much more scalable to the number of training data in contrast to the Kriging models.

To investigate the computational efficiency of CSEA, the runtime of different SAEAs on three-objective DTLZ2 are compared. The results over the number of the evaluations achieved by CSEA, CPS-MOEA, ParEGO, MOEA/D-EGO, and K-RVEA are shown in Fig. 11. Since NSGA-III does not involve any surrogates, NSGA-III is not included in comparing the computation time.

It can be observed that the runtime of CSEA, ParEGO, K-RVEA, and MOEA/D-EGO increases linearly with the number of evaluations, where the runtime of ParEGO and MOEA/D-EGO increases most rapidly. By contrast, the computation time of CSEA increases most slowly. The runtime of CSEA is significantly less than that of K-RVEA, MOEA/D-EGO, and ParEGO, and CPS-MOEA consumes the least runtime. The main reason is that CPS-MOEA adopts KNN to predict the candidate solutions and the computational complexity of KNN is far less than that of Kriging or FNN. In conclusion, CSEA is computationally efficient in comparison with K-RVEA, MOEA/D-EGO, and ParEGO.

V. CONCLUSION

In this paper, we have proposed a classification-based surrogate-assisted multiobjective SAEA, called CSEA, for expensive many-objective optimization. A classification criterion is proposed to divide solutions evaluated using the expensive objective functions into two different categories. Then a surrogate is employed to learn the classification criterion to predict the category of new candidate solutions and select potentially better converged solutions to be evaluated using the expensive objective functions. Assisted by the surrogate, the proposed CSEA is capable of solving expensive MaOPs satisfactorily with only a few hundreds instead of tens of thousands FEs.

An FNN is adopted as the classifier for predicting the possible category of new candidate solutions. Cross-validation is carried out to calculate the test errors for estimating the reliability of the predictions. The test errors are used to construct the reliability configuration for surrogate management taking advantage of the uncertainty of the prediction for enhancing the accuracy of the FNN, which has been demonstrated to play a key role in enhancing the performance of CSEA.

CSEA is compared with NSGA-III, CPS-MOEA, ParEGO, MOEA/D-EGO, and K-RVEA on the widely used test suites

DTLZ and WFG to demonstrate its effectiveness. The effectiveness of CSEA is examined in ZDT problems with up to 30 decision variables, mainly because the computational efficiency of FNNs is more scalable to the increase in the number of training data than that of the Kriging. It is also encouraging that the performance of CSEA is relatively insensitive to its parameters. Therefore, the overall performance of CSEA is highly competitive compared with the state-of-the-art SAEAs for expensive many-objective optimization.

This paper demonstrates that surrogates classifying solutions into dominated and nondominated solutions are promising for solving expensive MaOPs. Therefore, it deserves further efforts to develop surrogate management strategies in classification-based SAEAs for enhancing the convergence and promoting diversity. In addition, the effectiveness of other machine learning models for classification-based surrogates remains to be investigated. Finally, the proposed method will be extended to solve even higher dimensional expensive optimization problems.

ACKNOWLEDGMENT

REFERENCES

[1] F. Rivas-Dávalos and M. R. Irving, "An approach based on the strength Pareto evolutionary algorithm 2 for power distribution system planning," in *Evolutionary Multi-Criterion Optimization*. Heidelberg, Germany: Springer, 2005, pp. 707–720.

[2] L. Jia, Y. Wang, and L. Fan, "Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm," *Integr. Comput.-Aided Eng.*, vol. 21, no. 1, pp. 77–90, 2014.

[3] E. J. S. Pires, P. B. de Moura Oliveira, and J. A. T. Machado, "Multi-objective genetic manipulator trajectory planner," in *Proc. Workshops Appl. Evol. Comput.*, 2004, pp. 219–229.

[4] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.

[5] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multi-objective optimization," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, 2001, pp. 283–290.

[6] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. 5th Conf. Evol. Methods Design Optim. Control Appl. Ind. Problems*, 2001, pp. 95–100.

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

[8] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2004, pp. 832–842.

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

[10] A. Zhou *et al.*, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[11] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[12] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 2419–2426.

[13] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[14] R. J. Lygoe, M. Cary, and P. J. Fleming, "A real-world application of a many-objective optimisation complexity reduction process," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2013, pp. 641–655.

[15] J. B. Kollat, P. M. Reed, and R. Maxwell, "Many-objective groundwater monitoring network design using bias-aware ensemble Kalman filtering, evolutionary optimization, and visual analytics," *Water Resour. Res.*, vol. 47, no. 2, pp. 155–170, 2011.

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

[17] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, Dec. 2015.

[18] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, Aug. 2015.

[19] L. Pan, C. He, T. Ye, Y. Su, and X. Zhang, "A region division based diversity maintaining approach for many-objective optimization," *Integr. Comput.-Aided Eng.*, vol. 24, no. 3, pp. 279–296, 2017.

[20] B. Wilson, D. Cappelleri, T. W. Simpson, and M. Frecker, "Efficient Pareto Frontier exploration using surrogate approximations," *Optim. Eng.*, vol. 2, no. 1, pp. 31–50, 2001.

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

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

[23] D. Douguet, "e-LEA3D: A computational-aided drug design Web server," *Nucleic Acids Res.*, vol. 38, May 2010, Art. no. W615-21.

[24] S. Rashidi and P. Ranjitkar, "Bus dwell time modeling using gene expression programming," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 30, no. 6, pp. 478–489, 2015.

[25] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005.

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

[27] G. E. P. Box and N. R. Draper, *Empirical Model-Building and Response Surfaces*, vol. 424. New York, NY, USA: Wiley, 1987.

[28] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," document AD-A196 234, Defense Tech. Inf. Center, Fort Belvoir, VA, USA, 1988.

[29] D. G. Krige, "A statistical approach to some mine valuation and allied problems on the witwatersrand," Ph.D. dissertation, Faculty Eng., Univ. Witwatersrand, Johannesburg, South Africa, 1951.

[30] J. M. Zurada, *Introduction to Artificial Neural Systems*, vol. 8. St. Paul, MN, USA: West Publ. Company, 1992.

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

[32] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 481–494, Oct. 2002.

[33] S. D. Handoko, C. K. Kwoh, and Y.-S. Ong, "Feasibility structure modeling: An effective chaperone for constrained memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 740–758, Oct. 2010.

[34] S. D. Handoko, K. C. Kwoh, and Y. S. Ong, "Classification-assisted memetic algorithms for equality-constrained optimization problems with restricted constraint function mapping," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2011, pp. 391–400.

[35] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Comput.*, vol. 19, no. 6, pp. 1461–1475, 2015.

[36] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.

[37] 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, Feb. 2006.

[38] C.-W. Seah, Y.-S. Ong, I. W. Tsang, and S. Jiang, "Pareto rank learning in multi-objective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Brisbane, QLD, Australia, 2012, pp. 1–8.

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

[40] T. Chugh, K. Sindhya, K. Miettinen, J. Hakanen, and Y. Jin, "On constraint handling in surrogate-assisted evolutionary many-objective optimization," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2016, pp. 214–224.

[41] R. Allmendinger, M. T. M. Emmerich, J. Hakanen, Y. J. Jin, and E. Rigoni, "Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case," *J. Multi-Criteria Decis. Anal.*, vol. 24, nos. 1–2, pp. 5–24, 2017.

[42] M. Pilát and R. Neruda, "An evolutionary strategy for surrogate-based multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Brisbane, QLD, Australia, 2012, pp. 1–7.

[43] W. Gong, A. Zhou, and Z. Cai, "A multioperator search strategy based on cheap surrogate models for evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 746–758, Oct. 2015.

[44] S. Z. Martínez and C. A. C. Coello, "Combining surrogate models and local search for dealing with expensive multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2013, pp. 2572–2579.

[45] I. Loshchilov, M. Schoenauer, and M. Sebag, "A mono surrogate for multiobjective optimization," in *Proc. 12th Annu. Conf. Genet. Evol. Comput.*, 2010, pp. 471–478.

[46] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, 2015, pp. 2883–2890.

[47] X.-F. Lu and K. Tang, "Classification- and regression-assisted differential evolution for computationally expensive problems," *J. Comput. Sci. Technol.*, vol. 27, no. 5, pp. 1024–1034, 2012.

[48] K. S. Bhattacharjee and T. Ray, "A novel constraint handling strategy for expensive optimization problems," in *Proc. 11th World Congr. Struct. Multidiscipl. Optim.*, 2015, pp. 1041–1046.

[49] Y. Yun, H. Nakayama, and M. Arakava, "Generation of Pareto frontiers using support vector machine," in *Proc. MCDM*, Whistler, BC, Canada, Aug. 2004.

[50] I. Loshchilov, M. Schoenauer, and M. Sebag, "Dominance-based Pareto-surrogate for multi-objective optimization," in *Proc. Asia–Pac. Conf. Simulat. Evol. Learn.*, 2010, pp. 230–239.

[51] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-Metric selection," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2008, pp. 784–794.

[52] S. Bandaru, A. H. C. Ng, and K. Deb, "On the performance of classification algorithms for learning Pareto-dominance relations," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, 2014, pp. 1139–1146.

[53] J. Zhang, A. Zhou, and G. Zhang, "A multiobjective evolutionary algorithm based on decomposition and preselection," in *Bio-Inspired Computing-Theories and Applications*. Heidelberg, Germany: Springer, 2015, pp. 631–642.

[54] 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, Feb. 2008.

[55] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometr. Intell. Lab. Syst.*, vol. 39, no. 1, pp. 43–62, 1997.

[56] S. Yan and B. Minsker, "Optimal groundwater remediation design using an adaptive neural network genetic algorithm," *Water Resour. Res.*, vol. 42, no. 5, 2006, Art. no. W05407.

[57] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Netw.*, vol. 6, no. 6, pp. 861–867, 1993.

[58] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[59] S. Sapna, A. Tamilarasi, and M. P. Kumar, "Backpropagation learning algorithm based on Levenberg Marquardt algorithm," in *Proc. Comput. Sci. Inf. Technol. (CS and IT)*, vol. 2, 2012, pp. 393–398.

[60] C. He, Y. Tian, Y. Jin, X. Zhang, and L. Pan, "A radial space division based evolutionary algorithm for many-objective optimization," *Appl. Soft Comput.*, vol. 61, pp. 603–621, Dec. 2017.

[61] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[62] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, Apr. 2015.

[63] Y. Tian, H. Wang, X. Zhang, and Y. Jin, "Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 4, pp. 247–263, 2017.

[64] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *From Natural to Artificial Neural Computation*. Heidelberg, Germany: Springer, 1995, pp. 195–201.

[65] G. Lera and M. Pinzolas, "Neighborhood based Levenberg–Marquardt algorithm for neural network training," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1200–1203, Sep. 2002.

[66] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[67] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[68] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Informat.*, vol. 26, no. 4, pp. 30–45, 1996.

[69] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[70] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multiobjective Optimization* (Advanced Information and Knowledge Processing). London, U.K.: Springer, 2005.

[71] L. B. S. Huband, P. Hingston, 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, Oct. 2006.

[72] R. Cheng *et al.*, "Benchmark functions for the CEC'2017 competition on many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 1, pp. 67–81, 2017.

[73] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.

[74] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. New York, NY, USA: Wiley, 2001.

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

[76] H. B. Nielsen, S. N. Lophaven, and J. Søndergaard, "DACE, a MATLAB Kriging toolbox," Inf. Math. Model., Tech. Univ. Denmark, Lyngby, Denmark, Rep. IMM-TR-2002-12, 2002.

[77] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "Handling computationally expensive multiobjective optimization problems with evolutionary algorithms-a survey," Dept. Math. Inf. Technol., Univ. Jyväskylä, Jyväskylä, Finland, Rep. B 4/2015, 2015.

[78] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[79] C. Sun, J. Ding, J. Zeng, and Y. Jin, "A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems," *Memetic Comput.*, pp. 1–12, Jul. 2016.

[80] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

Ye Tian received the B.Sc. and M.Sc. degrees from Anhui University, Hefei, China, in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology.

His current research interests include multiobjective optimization methods and their application.

Handing Wang (S'10–M'16) received the B.Eng. and Ph.D. degrees from Xidian University, Xi'an, China, in 2010 and 2015, respectively.

She is currently a Research Follow with the Department of Computer Science, University of Surrey, Surrey, U.K. Her current research interests include nature-inspired computation, multiobjective optimization, multiple criteria decision making, surrogate-assisted evolutionary optimization, and real-world problems.

Dr. Wang is a member of the IEEE Computational Intelligence Society.
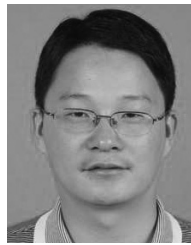
Xingyi Zhang received the B.Sc. degree from Fuyang Normal College, Fuyang, China, in 2003, and the M.Sc. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2009, respectively.

He is currently a Professor with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research interests include unconventional models and algorithms of computation, multiobjective optimization, and membrane computing.

Linqiang Pan received the Ph.D. degree from Nanjing University, Nanjing, China, in 2000.

Since 2004, he has been a Professor with the Huazhong University of Science and Technology, Wuhan, China. His current research interests include membrane computing, DNA computing and related DNA nanotechnology, and systems biology.
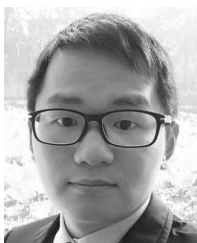
Cheng He received the B.Eng. degree from the Wuhan University of Science and Technology, Wuhan, China, in 2012. He is currently pursuing the Ph.D. degree with the School of Automation, Huazhong University of Science and Technology, Wuhan.

His current research interests include surrogate-assisted evolutionary optimization, multiobjective optimization, and real-world problems.

Yaochu Jin (M'98–SM'02–F'16) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, and the Dr.-Ing. degree from Ruhr-University Bochum, Bochum, Germany.

He is a Professor of Computational Intelligence with the Department of Computer Science, University of Surrey, Surrey, U.K. He is also a Finland Distinguished Professor, Finland, and a Changjiang Distinguished Visiting Professor, China. His current research interests include data-driven surrogate-assisted evolutionary optimization, evolutionary learning, interpretable and secure machine learning, and evolutionary developmental systems.

Dr. Jin was a recipient of the 2018 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award and the 2015 and 2017 *IEEE Computational Intelligence Magazine* Outstanding Paper Award. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and the Co-Editor-in-Chief of *Complex and Intelligent Systems*. He is an IEEE Distinguished Lecturer for the period 2013 to 2015 and 2017 to 2019 and the past Vice President for the Technical Activities of the IEEE Computational Intelligence Society from 2014 to 2015.