

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

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

**Abstract**—Surrogate-assisted evolutionary algorithms have been developed mainly for solving expensive optimization problems where only a small number of real fitness evaluations are allowed. Most existing surrogate-assisted evolutionary algorithms are designed for solving low-dimensional single or multi-objective 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**—Surrogate-assisted evolutionary optimization, Pareto dominance, expensive many-objective optimization, classification

## I. INTRODUCTION

MULTI-objective 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.

$$\begin{aligned} &\text{minimize} && F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ &\text{subject to} && x \in X, \end{aligned} \quad (1)$$

where  $m$  is the number of objectives and  $x$  is the decision vector [4]. A large number of multi-objective 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 (SPEA2) [6], the elitist non-dominated sorting genetic algorithm (NSGA-II) [7], the improved indicator based evolutionary algorithm (IBEA) [8], and the multi-objective evolutionary algorithm 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 non-dominated 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 non-dominated sorting approach (NSGA-III) [11], the hypervolume based evolutionary algorithm (HypE) [16], the knee-point driven evolutionary algorithm (KnEA) [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 (RdEA) [19]. Note that most existing MOEAs for solving MaOPs typically require tens of thousands of fitness evaluations.

One grand challenge in solving many real-world MaOPs is that one single fitness evaluation (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 fitness evaluations [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

L. Pan and C. He are 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. L. Pan is also with the School of Electric and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China. E-mail: {lqpanhust, chenghehust}@gmail.com.

Y. Tian and X. Zhang are with the Key Lab of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China. E-mail: {field910921, xyzhanghust}@gmail.com

H. Wang is with the Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom. Email: wanghanding.patch@gmail.com

Y. Jin is with the Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom. He is also with the State Key Laboratory of Synthetical Automation For Process Industries, Northeastern University, Shenyang, China. Email: yaochu.jin@surrey.ac.uk. (*Corresponding author*)

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 multi-objective optimization were proposed in the past decades, e.g., the generalized surrogate-assisted multi-objective memetic algorithm (GS-MOMA) [36], the weighted aggregation based multi-objective optimization assisted by efficient global optimization (ParEGO) [37], the efficient global optimization 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 multi-objective 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 non-trivial 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 fitness evaluation 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 work 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 the 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 fitness evaluation 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. 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 work. 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  $x$ :

$$\hat{f}(x) = f^*(x) + \xi(x), \quad (2)$$

where  $f^*$  is the true value of the objective or fitness value of the solution,  $\hat{f}$  is the approximated value, and  $\xi$  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 fitness evaluations with the computationally cheap surrogate, thereby reducing the computation time for solving expensive optimization problems [21].

In recent years, a plenty of multi-objective SAEAs have been shown to be very effective in reducing the needed number of FEs in solving MOPs [20], [49], [50], referring to [51] for a recent review. However, few multi-objective 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 reference vector based evolutionary algorithm (RVEA) [13]. K-RVEA was demonstrated to be competitive compared to a few state-of-the-art SAEAs for expensive many-objective optimization.

In multi-objective 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 hypervolume score of the population, and a Kriging model is built to approximate the function of the hypervolume [52]. 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 sub-problem [21].

In the second category of SAEAs, the surrogate serves as a classifier [53], [48] that divides the candidate solutions into good or bad solutions, e.g., dominated or non-dominated 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) [54]. In CPS-MOEA, the population is divided into two equal groups, one positive group and one negative group, according to the non-dominated 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 multi-objective evolutionary algorithm (RM-MEDA) [55]. 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 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 [56], and they have been widely applied in surrogate-assisted evolutionary algorithms [57]. 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  $\alpha$  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.  $K_{hidden}(\cdot)$  is known as the activation function of the hidden neurons [58], which is typically the sigmoid function or the hyperbolic tangent function, and  $K_{output}(\cdot)$  is the activation function of the output neurons, which is a sigmoid or linear function.

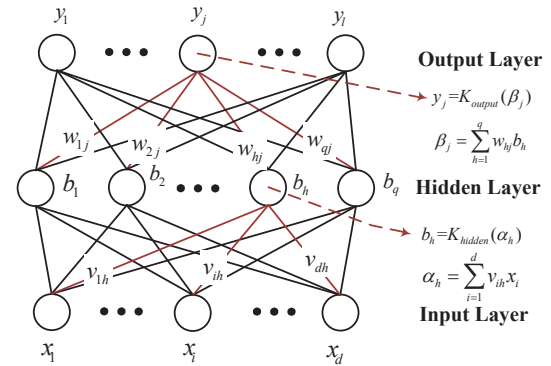


Fig. 1. A 3-layer feedforward neural network, where the number of neurons in the input, hidden and output layer is  $d$ ,  $q$ , and  $l$ , respectively, with  $K_{hidden}$  and  $K_{output}$  being the activation functions of the hidden and output neurons.

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 [59]. In this paper, the Levenberg-Marquardt back-propagation algorithm is adopted to train the FNN surrogate as it has been shown to be one of the fastest BP algorithm [60]. 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 [61] is employed to select a set of reference solutions to construct the classification boundary. RSEA is also adopted 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-dimensional 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  $\theta_i$  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



### Algorithm 1 RadialSelection( $P, K$ )

#### Output:

$P_R$  (reference population).  
1:  $[Y, G] \leftarrow \text{RadialGrid}(P, K)$   
2:  $C \leftarrow 0$   
3:  $Con(P) \leftarrow \left\| \frac{P - \min P}{\max P - \min P} \right\|$   
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:    $Q \leftarrow \arg \min Crowd(P)$   
8:    $Fit(Q, P_R) \leftarrow 0.1 \cdot m \cdot Con(Q) - \min \|Y_Q - Y_{P_R}\|$   
9:    $P_q \leftarrow \arg_{q \in K} \min Fit(Q, P_R)$   
10:    $P_R \leftarrow P_R \cup \{P_q\}$   
11:    $P \leftarrow P \setminus \{P_q\}$   
12:    $Crowd(q) \leftarrow Crowd(q) + 1$   
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:  $\theta_i \leftarrow 2\pi(i-1)/m$   $i = 1, 2, \dots, m$   
3:  $W_1 \leftarrow (\cos(\theta_1), \dots, \cos(\theta_m))$ ,  
    $W_2 \leftarrow (\sin(\theta_1), \dots, \sin(\theta_m))$   
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\*/

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. THE PROPOSED ALGORITHM

In this work, a classification based surrogate-assisted evolutionary algorithm (CSEA) is proposed for expensive many-objective optimization. A diagram describing the main 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** (Line 1 to 4): An initial population  $P$  with  $11d-1$  solutions is generated using Latin hypercube sampling [62], 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

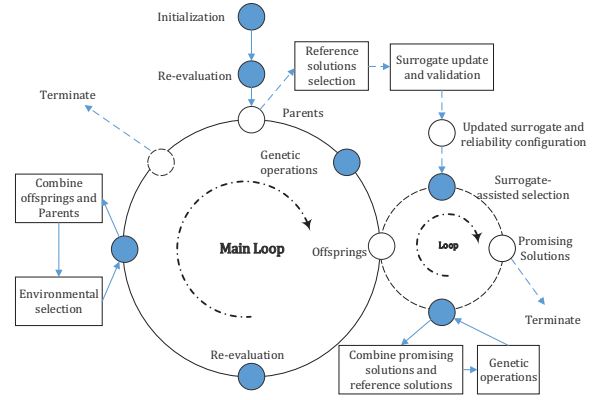


Fig. 2. A diagram of the proposed CSEA consisting of two loops. In the main loop, the solutions are selected according to the real objective function, while in the second loop, solutions are selected according to the surrogate.

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** (Line 7 to 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  $N$ ).
- (6) Repeat Steps (2), (3), (4), and (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 subsections.

#### 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

### Algorithm 3 The 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

```

criterion. In multi-objective optimization, a basic criterion for comparing two solutions is the Pareto dominance relationship [7], [63], [64], which categorizes solutions into dominated solutions and non-dominated 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 non-dominance 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.

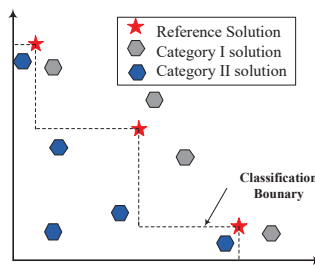


Fig. 3. An illustration of the proposed classification criterion using a set of reference solutions. A classification boundary is formed by three reference solutions (denoted by stars), which classifies solutions into category I (filled hexagons) and category II solutions (denoted by unfilled hexagons).

### 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:      $c \leftarrow c \wedge (P_i \text{ Dominate } R_j)$ 
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

```

### 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-dimensional 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,  $B1$ ,  $B2$ , and  $B3$ , 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  $s_1, s_3, s_5$  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.

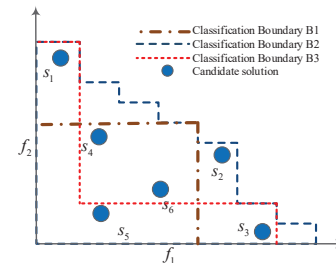


Fig. 4. An example illustrating the impact of the number of reference solutions on the performance of the proposed algorithm, where the classification boundaries  $B1$ ,  $B2$ , and  $B3$  are constructed by one, seven, and two reference solutions, respectively.

In summary, a small number of reference solutions will result in a small number of category II solutions that have 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 non-dominated solutions in the population, and CSEA with a small  $K$  could classify most non-dominated 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, DTLZ7 with different objectives. Fig. 5 shows the 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 multi-modal 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; 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);

### 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:  $D_{test} \leftarrow \{P_{i_1}, C_{i_1}\}$

- 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 (Line 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 work, 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 work, a 3-layer FNN is constructed with  $d$ -dimensional input, one-dimensional 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 [65]:

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

where  $x$  is the input variable and parameter  $\lambda$  is set to one in this work.

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 back-propagation method is adopted in this work. The details of the Levenberg-Marquardt back-propagation method can be found in [66]. 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

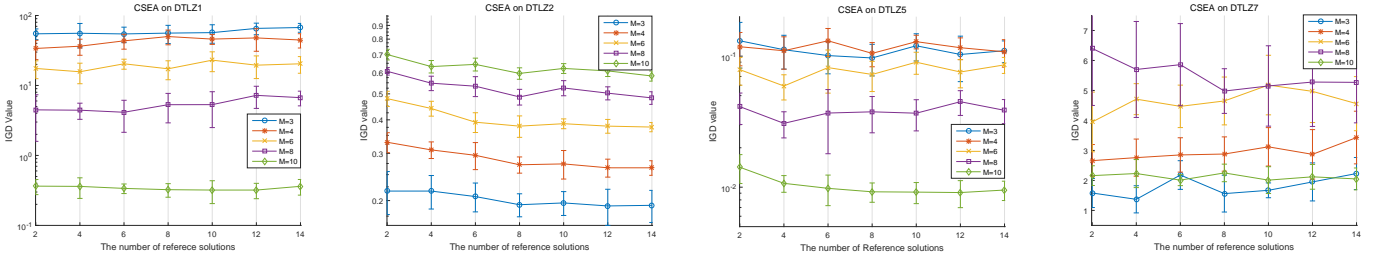


Fig. 5. The IGD results achieved by CSEA with 2,4,6,8,10,12, and 14 reference solutions on 3-,4-,6-,8-, and 10-objective DTLZ1, DTLZ2, DTLZ5, and DTLZ7, respectively.

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  $C_{p_1}, C_{p_2}, \dots, C_{p_{|Q|}}$ . The mean absolute error (MAE) of the prediction is defined as follows:

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

where  $|Q_c|$  denotes the number of solutions in  $Q_c$ ,  $\text{abs}(\ast)$  denotes the absolute value of  $\ast$ . 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 Eq. (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 subsection.

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.

- For  $p_2 < tr$ , the classifier is able to predict category II solutions.
- 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 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 [67] is adopted to determine the threshold value so that the influence of different classes on the learning process is reduced. Further discussions about the threshold will be given in the following subsection. In this paper, the threshold  $tr$  is set to  $0.5 \times \min\{rr, 1 - rr\}$ , where  $rr$  is given in Eq. (6).

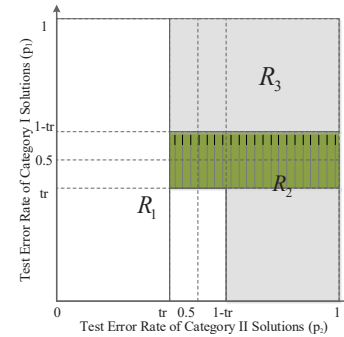


Fig. 6. The reliability configuration of the FNN constructed by the test errors  $p_1$  and  $p_2$  for estimating the uncertainty of the prediction.

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 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 [68] and polynomial mutation [69]. 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



---

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

---

**Output:**

$Q$  (offsprings).

```

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

```

---

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. Firstly, 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  $\{Cp_1, Cp_2, \dots, Cp_{|Q|}\}$  with their true categories (classified according to the objective values calculated using the expensive objective functions) are  $\{Cr_1, Cr_2, \dots, Cr_{|Q|}\}$ . The predicted rate of category II solutions  $rp$  is defined as

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

The real rate of category II solutions  $rr$  is defined as

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

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.

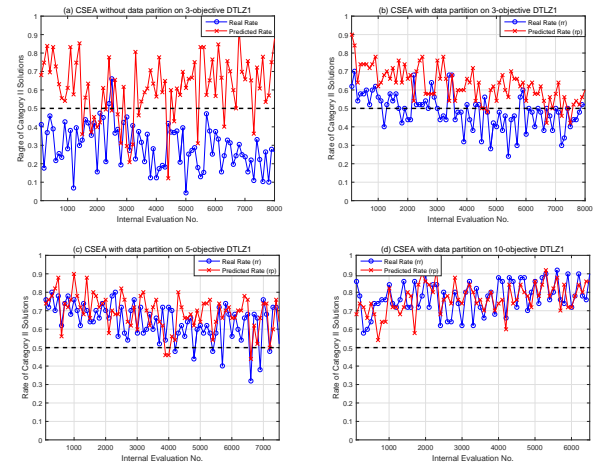


Fig. 7. The variations of the rate of category II solutions in the current population predicted by the FNN in CSEAs with and without data partition  $rp$  and that of the real rate  $rr$  on DTLZ1 with different numbers of objectives.

The  $rp$  obtained by CSEA with data partition is compared with the corresponding  $rr$  on DTLZ1 with 5 and 10 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 5 and 10 objectives are shown in Fig. 7(c) and Fig. 7(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 non-dominated solutions increases as the number of objectives increases and more non-dominated solutions are predicted as category II solutions. Meanwhile, the tendency of 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. The 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, 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.

TABLE I  
THE RESULTS OF  $CSEA^-$  AND CSEA ON 9 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)

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 [70]). 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 [71] with 3, 4, 6, 8, and 10 objectives, respectively. We also compared the algorithms on the WFG test suite [72], MaF test suite [73], 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 ‘ $\approx$ ’ indicates there is no statistically significant difference between the two compared algorithms.

##### A. Performance Indicator

For comparisons, the inverted generational distance (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 non-dominated 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 hypervolume (HV) [74] is not used to assess the performance of the compared algorithms as the obtained solutions of expensive many-objective optimization problems 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  $\Omega$  is the set of achieved non-dominated solutions, IGD is defined as follows:

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

where  $\text{dis}(x, \Omega)$  is the minimum Euclidean distance between  $x$  and points in  $\Omega$  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 work, simulated binary crossover [68] and polynomial mutation [69] 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 [75]. 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, differential evolution (DE) operators [76] 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 both the 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 condi-

TABLE II

SETTING OF THE POPULATION SIZE IN NSGA-III AND K-RVEA, WHERE  $p_1$  AND  $p_2$  ARE PARAMETERS CONTROLLING THE NUMBERS OF WEIGHT POINTS (VECTORS) ALONG THE BOUNDARY OF THE PARETO FRONT AND INSIDE IT, RESPECTIVELY.

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

tion, 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 200000 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 fitness evaluations before updating the models is set to  $20 \times (11d - 1)$ , and the other parameter settings are the same as in [21].<sup>1</sup> For K-RVEA, parameter  $\delta$  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 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 non-dominated solutions achieved by the compared algorithms on 3-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 fitness evaluation.

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 non-dominated solutions achieved by the compared algorithms on 10-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 results on DTLZ1 with 3 and 5 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 10-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  $11d - 1$ .

<sup>1</sup>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 [77] to optimize the hyperparameters, since it is time-consuming to optimize the hyperparameters in kriging using DE.

TABLE III  
THE 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(1.38e+0)–	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	1.60e+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	8.44e+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+1(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(9.84e+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.

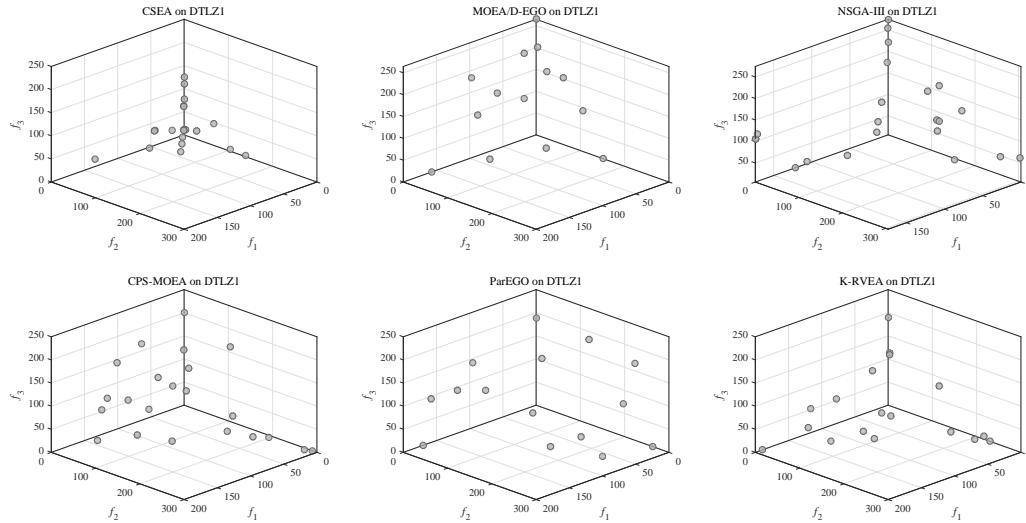


Fig. 8. The non-dominated front obtained by each algorithm on 3-objective DTLZ1 in the run associated with the median IGD value.

TABLE IV  
THE 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)	2.08e+1(3.93e+0)	2.73e+1(3.55e+2)

### E. Problems with More than Ten Decision Variables

Most existing SAEAs have been tested only on optimization problems with up to 30 decision variables [78], 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 Kriging models can become prohibitive when the amount of data for training the surrogate increases [79]. As a result, locating the global optimum or global Pareto front of high-dimensional multi-modal problems becomes hardly tractable as an accurate



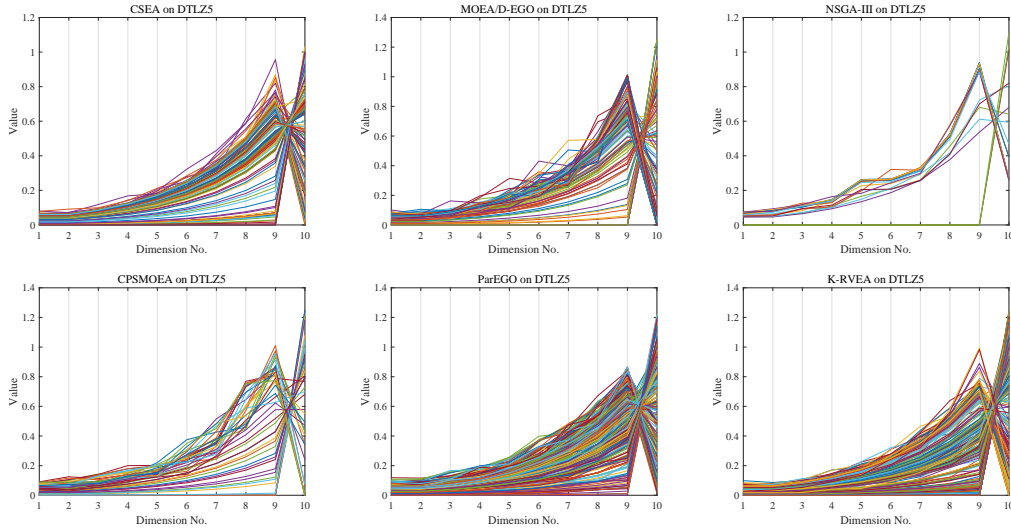


Fig. 9. The non-dominated front obtained by each algorithm on 10-objective DTLZ5 in the run associated with the median IGD value.

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 [80].

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 [81] 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 non-dominated 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 10 decision variables, but the non-dominated 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. 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 3-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 work, we have proposed a classification based surrogate-assisted multi-objective 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 many-objective optimization problems 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



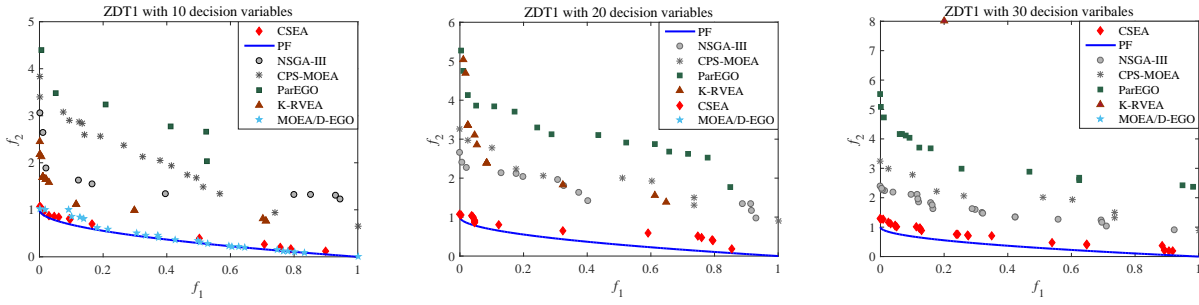


Fig. 10. The non-dominated front obtained by each algorithm on bi-objective ZDT1 with 10, 20 and 30 decision variables in the run associated with the median IGD value, respectively.

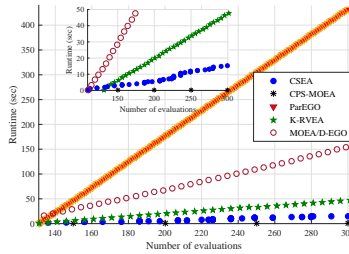


Fig. 11. Runtime over the number of evaluations in CSEA, CPS-MOEA, ParEGO, MOEA/D-EGO, and K-RVEA on 3-objective DTLZ2.

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 SAEs for expensive many-objective optimization.

The present work demonstrates that surrogates classifying solutions into dominated and non-dominated solutions are promising for solving expensive many-objective optimization problems. Therefore, it deserves further efforts to develop surrogate management strategies in classification based surrogate-assisted evolutionary algorithms 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

Part of the work was done when C. He and Y. Tian were visiting the Department of Computer Science, University of Surrey, UK, as visiting PhD students. The work was supported in part by National Natural Science Foundation

of China (61320106005, 61502004, 61502001, 61772214, and 61672033), in part by the Innovation Scientists and Technicians Troop Construction Projects of Henan Province (154200510012), and in part by an EPSRC grant (No. EP/M017869/1). The authors are grateful to Tinkle Chugh's comments on the paper.

#### 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*. 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," *Integrated Computer-Aided Engineering*, 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 *Workshops on Applications of Evolutionary Computation*, 2004, pp. 219–229.
- [4] C. A. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, 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 *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001)*. Citeseer, 2001, pp. 283–290.
- [6] E. Zitzler, M. Laumanns, L. Thiele *et al.*, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Eurogen*, vol. 3242, no. 103, 2001, pp. 95–100.
- [7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [8] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 832–842.
- [9] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [10] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [12] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *IEEE Congress on Evolutionary Computation*. Citeseer, 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 Transactions on Evolutionary Computation*, vol. 20, pp. 773–791, 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*. 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 Resources Research*, vol. 47, no. 2, pp. 155–170, 2011.
- [16] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [17] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.
- [18] H. Wang, L. Jiao, and X. Yao, "Two\_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.
- [19] L. Pan, C. He, T. Ye, Y. Su, and X. Zhang, "A region division based diversity maintaining approach for many-objective optimization," *Integrated Computer-Aided Engineering*, 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," *Optimization and Engineering*, 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 Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.
- [22] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 62–76, 2009.
- [23] D. Douquet, "e-LEA3D: a computational-aided drug design web server," *Nucleic Acids Research*, p. gkq322, 2010.
- [24] S. Rashidi and P. Ranjekar, "Bus dwell time modeling using gene expression programming," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 6, pp. 478–489, 2015.
- [25] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.
- [26] —, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
- [27] G. E. Box and N. R. Draper, *Empirical model-building and response surfaces*. Wiley & New York, 1987, vol. 424.
- [28] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," DTIC Document, Tech. Rep., 1988.
- [29] D. G. Krige, "A statistical approach to some mine valuation and allied problems on the witwatersrand," Ph.D. dissertation, University of Witwatersrand, 1951.
- [30] J. M. Zurada, *Introduction to artificial neural systems*. West publishing company St. Paul, 1992, vol. 8.
- [31] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, 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 Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.
- [33] S. D. Handoko, C. K. Kwoh, and Y.-S. Ong, "Feasibility structure modeling: an effective chaperone for constrained memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 740–758, 2010.
- [34] S. D. Handoko, C. Kwoh, and Y. Ong, "Classification-assisted memetic algorithms for equality-constrained optimization problems with restricted constraint function mapping," in *2011 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2011.
- [35] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Computing*, 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 Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.
- [37] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.
- [38] C.-W. Seah, Y.-S. Ong, I. W. Tsang, and S. Jiang, "Pareto rank learning in multi-objective evolutionary algorithms," in *2012 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2012.
- [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 Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2016.
- [40] T. Chugh, K. Sindhya, K. Miettinen, J. Hakanen, and Y. Jin, "On constraint handling in surrogate-assisted evolutionary many-objective optimization," in *International Conference on Parallel Problem Solving from Nature*. Springer, 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," *Journal of Multi-Criteria Decision Analysis*, vol. 24, no. 1-2, pp. 5–24, 2017.
- [42] M. Pilát and R. Neruda, "An evolutionary strategy for surrogate-based multiobjective optimization," in *2012 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 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 Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 746–758, 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 *2013 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2013, pp. 2572–2579.
- [45] I. Loshchilov, M. Schoenauer, and M. Sebag, "A mono surrogate for multiobjective optimization," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2010, pp. 471–478.
- [46] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 2883–2890.
- [47] X.-F. Lu and K. Tang, "Classification-and regression-assisted differential evolution for computationally expensive problems," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 1024–1034, 2012.
- [48] K. S. Bhattacharjee and T. Ray, "A novel constraint handling strategy for expensive optimization problems," in *11th World Congress on Structural and Multidisciplinary Optimization*, 2015.
- [49] Y. Yun, H. Nakayama, and M. Arakava, "Generation of Pareto frontiers using support vector machine," *MCDM04*, 2004.
- [50] I. Loshchilov, M. Schoenauer, and M. Sebag, "Dominance-based Pareto-surrogate for multi-objective optimization," in *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 2010, pp. 230–239.
- [51] R. Allmendinger, M. Emmerich, J. Hakanen, Y. Jin, and E. Rigoni, "Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case," *Journal of Multi-Criteria Decision Analysis*, vol. 24, no. 1-2, pp. 5–24, 2017.
- [52] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted  $S$ -Metric selection," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2008, pp. 784–794.
- [53] S. Bandaru, A. H. C. Ng, and K. Deb, "On the performance of classification algorithms for learning Pareto-dominance relations," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 1139–1146.
- [54] J. Zhang, A. Zhou, and G. Zhang, "A multiobjective evolutionary algorithm based on decomposition and preselection," in *Bio-Inspired Computing-Theories and Applications*. Springer, 2015, pp. 631–642.
- [55] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.
- [56] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43–62, 1997.
- [57] S. Yan and B. Minsker, "Optimal groundwater remediation design using an adaptive neural network genetic algorithm," *Water Resources Research*, vol. 42, no. 5, 2006.
- [58] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, no. 6, pp. 861–867, 1993.
- [59] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [60] S. Sapna, A. Tamilarasi, and M. P. Kumar, "Backpropagation learning algorithm based on levenberg marquardt algorithm," *Comp Sci Inform Technol (CS and IT)*, vol. 2, pp. 393–398, 2012.
- [61] C. He, Y. Tian, Y. Jin, X. Zhang, and L. Pan, "A radial space division based many-objective optimization evolutionary algorithm," *Applied Soft Computing*, vol. 61, pp. 603–621, 2017.
- [62] 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.

- [63] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, “An efficient approach to non-dominated sorting for evolutionary multi-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2015.
- [64] Y. Tian, H. Wang, X. Zhang, and Y. Jin, “Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization,” *Complex & Intelligent Systems*, vol. 3, pp. 247–263, 2017.
- [65] 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*. Springer, 1995, pp. 195–201.
- [66] G. Lera and M. Pinzolas, “Neighborhood based levenberg-marquardt algorithm for neural network training,” *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1200–1203, 2002.
- [67] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.
- [68] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.
- [69] K. Deb and M. Goyal, “A combined genetic adaptive search (geneas) for engineering design,” *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.
- [70] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “PlatEMO: A MATLAB platform for evolutionary multi-objective optimization,” *IEEE Computational Intelligence Magazine*, vol. 12, pp. 73–87, 2017.
- [71] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable test problems for evolutionary multiobjective optimization*, ser. Advanced Information and Knowledge Processing. Springer London, 2005.
- [72] L. B. S. Huband, P. Hingston and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
- [73] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, “Benchmark functions for the CEC’2017 competition on many-objective optimization,” *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.
- [74] L. While, P. Hingston, L. Barone, and S. Huband, “A faster algorithm for calculating hypervolume,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.
- [75] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.
- [76] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [77] H. B. Nielsen, S. N. Lophaven, and J. Søndergaard, “DACE, a MATLAB Kriging toolbox,” *Informatics and mathematical modelling. Lyngby-Denmark: Technical University of Denmark, DTU*, 2002.
- [78] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, “Handling computationally expensive multiobjective optimization problems with evolutionary algorithms—a survey,” *Reports of the Department of Mathematical Information Technology, Series B, Scientific Computing no. B*, vol. 4, 2015.
- [79] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [80] C. Sun, J. Ding, J. Zeng, and Y. Jin, “A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems,” *Memetic Computing*, pp. 1–12, 2016.
- [81] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.



**Linqiang Pan** received the Ph.D. in 2000 from Nanjing University. Since 2004, he is a Professor at Huazhong University of Science and Technology, Wuhan, China. His research interests include membrane computing, DNA computing and related DNA nanotechnology, and systems biology.



**Cheng He** received the B.Eng. in 2012 from Wuhan University of Science and Technology. Currently, he is a Ph.D. degree candidate in the School of Automation, Huazhong University of Science and Technology. His main research interests include surrogate-assisted evolutionary optimization, multi-objective optimization, and real-world problems.



**Ye Tian** received the B.Sc. in 2012 and the M.Sc. in 2015 both from Anhui University. Currently, he is a Ph.D. degree candidate in the School of Computer Science and Technology, Anhui University. His current research interest is multi-objective 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 fellow with the Department of Computer Science, University of Surrey, Guildford, UK. Dr. Wang is a member of IEEE Computational Intelligence Society. Her research interests include nature-inspired computation, multi-objective optimization, multiple criteria decision making, surrogate-assisted evolutionary optimization, and real-world problems.



**Xingyi Zhang** received the B.Sc. from Fuyang Normal College in 2003, and the M.Sc. in 2006 and Ph.D. in 2009 both from Huazhong University of Science and Technology. Currently, he is a professor in the School of Computer Science and Technology, Anhui University. His main research interests include unconventional models and algorithms of computation, multi-objective optimization and membrane computing.



**Yaochu Jin** (M’98–SM’02–F’16) is a Professor in Computational Intelligence, Department of Computer Science, University of Surrey, Guildford, U.K. He is also a Finland Distinguished Professor, Finland, and a Changjiang Distinguished Visiting Professor, China. His main research interests include data-driven surrogate-assisted evolutionary optimization, evolutionary learning, interpretable and secure machine learning, and evolutionary developmental systems.

Dr Jin is the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Co-Editor-in-Chief of Complex & Intelligent Systems. He is an IEEE Distinguished Lecturer (2013–2015 and 2017–2019) and past Vice President for Technical Activities of the IEEE Computational Intelligence Society (2014–2015). He is the recipient of the 2018 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, and the 2015 and 2017 IEEE Computational Intelligence Magazine Outstanding Paper Award.