A Surrogate-Assisted Many-Objective Evolutionary Algorithm using Multi-Classification and Coevolution for Expensive Optimization Problems

Ruoyu Wang¹, Yuee Zhou¹, Hanning Chen², Lianbo Ma¹, Meng Zheng³

¹Software College, Northeastern University, Shenyang, 110016, China.
²School of Computer Science and Technology, Tiangong University, Tianjin 300387, China.
³Key Laboratory of Robotics and Key Laboratory of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110169, China.

Corresponding authors: Hanning Chen (chenhanning@tiangong.edu.cn) and Lianbo Ma (malb@swc.neu.edu.cn).

This work is supported by National Natural Science Foundation of China Nos. 41772123 and 62022088, Fundamental Research Funds for the Central Universities No.N2117005 and Joint Funds of the Natural Science Foundation of Liaoning Province und Grant 2021-KF-11-01.

ABSTRACT Surrogate-assisted evolutionary algorithms have received a surge of attentions for their promising ability of solving expensive optimization problems. Existing surrogate-assisted evolutionary algorithms usually adopt the regression models and the binary classification models to guide the evolution of the population for solving the multiobjective optimization problems. However, the regression models will make the algorithm to be increasingly computation-expensive as the number of objectives increases, while the use of the binary classification models might suffer from the poor diversity since the diversified information of solutions cannot be reflected in these classification models. For this issue, this paper proposes a surrogate-assisted many-objective evolutionary algorithm using the cooperation of the multi-classification and regression models to improve the search quality while reducing the computational cost. Our approach includes two parts: At the model training stage, a multi-classification model is constructed to divide the whole population into several classes for ensuring diversity, a distance regression model and an angle regression model are used to select solutions with better convergence and diversity in each class; At the evolution stage, a coevolutionary framework is used to guide the evolution according to a new selection criterion. Experimental results verify the effectiveness of the proposed algorithm on a set of expensive test problems with up to 10 objectives.

INDEX TERMS Coevolution, expensive optimization, multi-classification, surrogate-assisted evolutionary algorithm

I. INTRODUCTION
Many real-world optimization problems involve more than two conflicting objectives to be optimized simultaneously [1-4]. Formally, these problems can be formulated as

\[
\begin{align*}
\text{minimize} & \quad \{ f_1(x), \ldots, f_M(x) \} \\
\text{subject to} & \quad x \in S
\end{align*}
\]

where \( x \) is the decision variable and \( M \) is the number of objectives [5]. Such a problem of (1) is known as a multi-objective optimization problem (MOP) (if \( M=2 \) and 3), or many-objective optimization problem (MaOP) (if \( M > 3 \)). Different from single-objective algorithms [6], [7], multi-objective evolutionary algorithms (MOEAs) aim to achieve desirable Pareto fronts (PFs) of various MOPs, and have shown an excellent performance for its population-based optimization characteristics. Many MOEAs have been proposed in the past decades for solving MOPs [8-12] and MaOPs [13-17].
However, most existing MOEAs are not efficient to deal with the expensive optimization problems, since they generally require thousands of fitness evaluations (FEs) and one single FE can be very time-consuming or financially very expensive. For example, a single function evaluation based on computational fluid dynamic (CFD) simulations could take from minutes to hours [18], which is computationally prohibitive to MOEAs.

One approach to address above problem is to use efficient surrogates for approximating, which can effectively reduce the computational cost. Such algorithms are called surrogate-assisted evolutionary algorithms (SAEAs). In the past decades, a number of SAEAs have been developed for expensive MOPs [19-26], which are called multiobjective SAEAs (MOSAEAs). However, there are still several challenging issues: First, various types of surrogate models can be used in SAEAs, such as polynomial response surface methodology [27], Gaussian process model (also known as Kriging model) [28], artificial neural networks [29], and support vector machines [30]. However, there are no clear rules for selecting appropriate type of surrogates for the algorithm [31]. The second issue is what should be predicted by the surrogate [32]. At present, the most common SAEAs use surrogate to approximate the objective function or fitness function. Moreover, there are some SAEAs use surrogate to predict the relationship between the solutions. Third, how to select MOEA as the evolution engine according to the property of the problem is another key issue [33]. In addition, how to select effective solutions from the current population to update the surrogate is another important issue.

At present, the most popular method is to use the surrogate to approximate the objective function. However, when MOSAEAs use surrogates to approximate the objective functions, the computational cost of constructing surrogates will grow accordingly as the number of objectives increases. For this problem, the binary classification surrogates are introduced to predict the relationship between the solutions, e.g., the dominance relationship of the solutions [24]. In this way, the surrogate is able to divide the newly generated population into positive and negative groups. However, since it is difficult to compare solutions which are both predicted as positive group, the diversity of solutions cannot be maintained well. In this paper, we propose a new surrogate-assisted many-objective evolutionary algorithm based on multi-classification and coevolution mechanism, called ACDEA, the algorithm integrates three effective surrogates at each generation which could can ensure diversity while controlling the number of surrogates.

Our contributions mainly include:

- A multi-surrogate cooperation mechanism is constructed. In this mechanism, a multi-classification model is used to divide the whole population into a set of classes, which is conductive to the diversity; a distance regression model is used to predict the convergence of candidate solutions and an angle regression model is used to explore uncertain region of each class and improve the diversity.

- A coevolutionary framework is incorporated to guide the evolution to search promising solutions according to a new selection criterion.

In the rest of this paper, the related work and our motivation are described in Section II. In Section III, the proposed many-objective SAEAs ACDEA is introduced. Numerical experiments are detailed in Section IV. Finally, the conclusions are outlined in Section V.

II. RELATED WORK AND MOTIVATION

In this section, we first present a brief description of some existing MOSAEAs, and then introduce our motivation.

A. SURROGATE-ASSISTED MOEAs

In principle, SAEAs use surrogates to approximate the objective functions or related functions [34], which can be formulated as

\[
\hat{f}(x) = f^* (x) + \xi(x)
\]  

(2)

where \( f^* \) is the true function, \( \hat{f} \) is the approximate function gotten by the surrogate and the \( \xi(x) \) is the error function. A certain number of SAEAs were proposed in the past decades, which can be roughly divided into two categories according to the intention of using surrogate.

In the first category, the surrogate is used to approximate the objective function or fitness function. For example, in the MOEA using weight aggregation and efficient global optimization (ParEGO) [19], an aggregation function is set up by randomly selected vectors at each generation, and a Kriging surrogate is created to approximate the aggregation function. In the MOEA/D assisted by efficient global optimization (MOEA/D-EGO) [20], the used surrogate is to approximate the objective function of each subproblem. And in the Kriging assisted RVEA (K-RVEA) [21], the surrogates based on Kriging are created to approximate each objective function at each generation. This sort of SAEAs employ traditional MOEAs after creating the surrogates, and they can lead to solutions with both excellent convergence and diversity.

In the second category, the surrogate is used as a classifier to divide the newly generated population into positive and negative groups. At present, there are few such work. For example, in domination-based MOEA (CPS-MOEA) [22], the training set is divided into positive and negative groups according to the nondominated sorting, then the surrogate is created to select ‘good’ solutions from newly generated population at each generation. In classification and regression-assisted differential evolution algorithm (CRADE) [23], the classification surrogate is used to discard offspring solutions worse than their parents. In classification-based surrogate-assisted evolutionary algorithm (CSEA) [24], the solutions in training sets are divided into good group and bad
group according to a classification boundary consists of several reference solutions, and then an artificial neural network is applied to predict the dominance relationship between newly generated solutions. This category of SAEAs usually needs only one surrogate at each generation.

The above two categories of SAEAs have their own advantages, but they also have their own disadvantages. We put forward our motivation according to their shortcomings.

Figure 1. An example of the surrogate divides the population into positive and negative groups.

B. MOTIVATION

The number of surrogates is an important factor related to the efficiency of MOSAEAs since the growing number of surrogates will increase the difficulty of managing these surrogates, and also the computation cost of constructing surrogates. Controlling the number of surrogates can also provide an opportunity to use more complex surrogates.

Although some MOSAEAs use uni-surrogate at each generation, e.g., CSEA[24]. Those algorithms will instead post a challenging diversity issue, which is caused by the use of the classifier as the surrogate. In such MOSAEAs, the surrogate divides the newly generated population into positive and negative groups. However, it is difficult to further compare the solutions which are both predicted as positive group, which always leads to the poor diversity. An example is given in Fig. 1, we can see that in the positive group, there are four positive solutions gathering together, and another positive solution far from them. Obviously they have different diversity, however the surrogate predicts that they are equally good.

Many MOSAEAs construct a corresponding surrogate for each objective, with the aim of approximating the exact value of the objective, e.g., KRVEA[21]. These algorithms can guarantee better diversity because of the diversity strategy they use, e.g., reference vector guided method used in KRVEA which divides individuals of different diversity into different categories, however, such strategies always need to know each objective value of the individuals. Accordingly, with the increase of the number of objectives, more surrogates are needed. Therefore, the motivation of our algorithm is to establish a method of constructing the surrogates which can control the number of surrogates and ensure the diversity as well.

This paper suggests a MOSAEA called ACDEA, the core ideas is: using a multi-classification model to divide solutions with different diversity into different classes, which can avoid establishing surrogates for each objective. And using the other two surrogates to find promising solutions in each class. The problems presented in above can be addressed properly.

III. PROPOSED METHOD

A. FRAMEWORK

In this paper, an ACDEA is proposed for expensive many-objective optimization. The core concept of ACDEA is using multi-classification surrogate to divide solutions into several classes and using the other two surrogates to find promising solutions in each class. Fig. 2 describes the framework of the proposed ACDEA.

Figure 2. A simple flow chart of framework of proposed ACDEA.

The pseudo code of ACDEA is presented in Algorithm 1, which can be divided into 5 main steps as follows:

1) Initialization (Lines 1–5): An initial population \( P_0 \) with \( 11d-1 \) solutions is generated using Latin hypercube sampling [35]. \( N \) uniformly distributed reference vectors are generated according to \( N \) uniformly distributed reference points which are generated by the canonical simplex-lattice design method [14], where \( d \) is the dimension of decision variables. Set \( F \) to \( 11d-1 \) which represents the number of solutions have already evaluated by
expensive objective functions. The solutions in \( P_0 \) are copied to archive \( A_1 \) and \( T_1 \), which used to save all solutions that have already evaluated by expensive objective functions and the solutions used as training sets respectively. The initial population \( P_0 \) is used as the training set of the first-generation surrogate.

2) **Creating surrogate (Lines 7-9):** \( K \) reference solutions are selected according to the method described in section III-C from the current training set which are already evaluated by the expensive function evaluations. Then three labels of each solution in training set are calculated according to the reference solutions, which are Angle, Distance and Class (described in section III-B). Then a multi-classification surrogate is created based the label Class, and two regression surrogates are created based the label Angle and the label Distance.

3) **Coevolution using surrogate (Lines 10):** Current population is classified into \( K \) classes by the multi-classification surrogate. In each class, offspring solutions are generated by crossover and mutation [36], and three labels of each offspring are predicted by the surrogates, then the promising solutions are selected from each class based on a criteria called ACD. These operations are performed separately in each class. All promising solutions are then reassigned to each class by their predicted value Class and to be the parents of next generation.

4) **Selection of solutions to be re-evaluated (Lines 11-12):** \( K \) solutions are selected from the last generation population \( Q \) of coevolution based the criteria ACD. Then these \( K \) solutions are re-evaluated by expensive function evaluations. These \( K \) solutions are added to archive \( A_1 \), and the value of FE is updated.

5) **Selection of next generation training set (Lines 13):** \( 11d-1 \) solutions are selected from archive \( A_1 \) as the training set of the next generation surrogates.

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

### Algorithm 1 Framework of ACDEA

**Input:**
- \( N \) (number of reference vectors), \( K \) (number of reference solutions = number of classes that search space divided into), \( FE_{max} \) (maximum number of evaluation by expensive function), \( w_{max} \) (maximum iterations of coevolution before the surrogates update), \( d \) (number of decision variables)

**Output:** nondominated solutions in \( A_1 \)

1: \( P_0 \) ← Initialize the population with 11\( d-1 \) solutions using Latin hypercube sampling method
2: \( V \) ← Initialize \( N \) uniformly distributed reference vectors
3: \( FE \) ← 11\( d-1 \)
4: \( T_1 \) ← \( P_0 \) /* save the solutions used as training sets in archive \( T_1 \) */
5: \( A_1 \) ← \( P_0 \)
6: while \( FE \leq FE_{max} \) do
7: \( P_n \) ← ReferenceSelect(\( T_1, K, V \)) /* select reference solutions \( P_n \) */
8: \( \{0, Dic, C\} \) ← calculate three labels of each solution in \( T_1 \)
9: \( \{model\} \) ← Train three models based the labels of \( T_1 \)
10: \( \{model\} \) ← \( \{0, Dic, C\} \) are three labels of each solution*/
11: \( R \) ← Select solutions to be re-evaluated from \( Q \)
12: \( A_1 \) ← \( \{0, Dic, C\} \)
13: \( T_1 \) ← ReferenceSelect(\( A_1, 11d-1, V \))
14: \( FE \) ← \( FE + |R| \)
15: end

### B. Creation of Three Surrogates

In ACDEA, three surrogates are created at each generation according to three labels of the training data, these three labels are Angle, Distance, and Class. The definitions of three labels are as follows:

1) **Class**

The objective space is divided into several classes, label Class represents the class that the solution assigned to. The specific classification method is as follows.

First, \( K \) reference solutions with both good convergence and diversity are selected from current training sets based on the method described in section III-C. Then the entire objective search space is divided into \( K \) classes based on the angle to the reference solutions. Similarly, the other solutions in training sets are assigned to each class according to the angle to the reference solutions. An example is given in Fig. 3. By this way, each solution in training set has its label Class, which are used to train a multi-classification surrogate. With the multi-classification surrogate, the label Class of newly generated solutions could be predicted without expensive fitness evaluations.

We use the multi-classification surrogate because we want to find promising solutions in each class so that could increase the diversity.

2) **Distance**

Label Distance represents the distance from the solution to the origin. A distance regression surrogate is created based on this label, this surrogate is used to predict the label Distance of the newly generated solution. Label Distance could indicate the convergence of solutions to some extent.

![Figure 3. An example of dividing training set into three classes. The classification boundary is between two reference solutions.](image-url)
with two reference solutions.

Label 3) the reference solutions represented by red dots. (b) is an example of selecting two reference solutions, where the reference solution that it assigned to. An angle surrogate are both established by Kriging models. It should be noted that since the predicted value of the multi-classification model may be decimal, we integer it as use algorithm 3.

![Image](image1.png)

**Figure 4.** An example of three labels of two solutions which are divided into class I and II separately.

![Image](image2.png)

**Figure 5.** (a) is an example of four solutions assign to three vectors, vector $V_1$ and $V_3$ are called the active vector since they have solutions assign to. (b) is an example of selecting two reference solutions, where the reference solutions represented by red dots.

3) Angle

Label $\theta$ represents the angle between the solution and the reference solution that it assigned to. An angle regression surrogate is created based on this label, which is used to predict the label $\theta$ of the newly generated solution. Angle is help for searching uncertain region and further improving the diversity, which will be discussed in section III-D.

For instance, let us consider the situation shown in Fig. 4, with two reference solutions $R_1$ and $R_2$, and the other two solutions $S_1$ and $S_2$. As the angle $\theta_1$ between the solution $S_1$ and the reference solution $R_1$ is less than the angle between the $S_1$ and the other reference solutions, this solution is assigned to the reference solution $R_1$ and the distance between $S_1$ and the origin is denoted by $Dic_1$. Therefore, the label $Class$ of $S_1$ is denoted by $I$, and the label $Distance$ and $Angle$ of $S_1$ are denoted by $Dic$ and $\theta_1$ respectively. Similarly, $II$, $Dic_2$, and $\theta_2$ will be the three labels of solution $S_2$ respectively.

The multi-classification surrogate, distance surrogate and angle surrogate are both established by Kriging models. It should be noted that since the predicted value of the multi-classification model may be decimal, we integer it as use algorithm 3.

**Algorithm 2 ReferenceSelect($P$, $K$, $V$)**

**Input:**

$P$, $K$ (select $K$ solutions from $P$), $V$ (reference vectors)

**Output:**

$P_N$ (selected solutions)

1: $Va \leftarrow$ Assign population $P$ to vector $V$, find the active vector with at least one solution
2: if $|Va| > K$
3: $[kmeans (Va, K)]\*$ $Va$ is clustered into $k$ classes, $*$
4: $P = \{P_1, P_2, ..., P_k\} \*$ Clustering results $*$
5: for $i = 1 : K$
6: $P_{N_k} \leftarrow$ find solutions with minimum distance to origin from $P^k$
7: end
8: else
9: $P_{N_1} \leftarrow$ find solution with minimum distance to origin from the solutions that divided into the same vector
10: $P_{N_2} \leftarrow$ find ($K - |Va|$) solutions with minimum distance from the remaining solutions
11: $P_N \leftarrow P_{N_1} \cup P_{N_2}$
12: end

**Algorithm 3 PredictLabel($model$, $P$, $K$)**

**Input:**

$model$ (three surrogates), $K$ (number of classes), $P$ (population to be predicted)

**Output:**

$[Class, Dic, \theta]$ (three predicted labels of population $P$)

1: $[C, Dic, \theta] \leftarrow$ use three models to predict three labels of $P$
2: if $C < 0.5$
3: $Class = 1$
4: else if $C > K$
5: $Class = K$
6: else
7: $Class = round(C)$
8: end

**C. SELECTION OF REFERENCE SOLUTIONS**

The reference solutions should have both good convergence and diversity. Good diversity helps to make the classification of the search space more uniform, and good convergence makes the label $\theta$ could better explore the uncertain region. This paper selects the reference solutions based on a criterion inspired by KRVEA [21], which is summarized in Algorithm 2. The description of selecting reference solutions is as follow:

First, a set of uniformly distributed reference vectors is generated. The solutions in training sets are assigned to the
reference vectors according to the angle between the solution and the vector. An example is shown in Fig. 5(a), where solutions $S_1$, $S_2$, $S_3$ are assigned to the vector $V_2$ since the angle between these solutions and $V_2$ is less than the angle between them and the other vectors. Similarly, solution $S_4$ is assigned to the vector $V_1$. The reference vector having at least one solution assigned to is called the active reference vector.

Then, all active reference vectors are clustered into $K$ groups, where $K$ is the number of reference solutions we need. By this way, all solutions are divided into $K$ groups according to the assigned active vector. Different from APD criterion used in KRVEA [21], we select the solution with the shortest distance to the origin in each group as the reference solution. This is due to the fact that the convergence of reference solutions is more important for ACDEA, which has been validated by experiments. An example is illustrated in Fig. 5(b), where seven solutions are divided into two groups with the clustering of reference vectors. In each group, the selected solutions represented by red.

D. COEVOLUTION BY USING SURROGATES

The coevolutionary framework is used to search promising solutions before updating the surrogates, which is described as follows:

First, the current population $P$ is divided in to $K$ classes $\{P^1, P^2, ..., P^K\}$ according to the multi-classification surrogate (If the real objective values are known, the prediction is omitted), each subpopulation is regarded as the parent population of the class which it belongs to. The number of individuals in each subpopulation is recorded as $n_{class}$.

In each subpopulation $P^{class}$, offspring population $O$ with $n_{class}$ solutions are generated by the simulated binary crossover [36]. The parent and offspring population are combined and $n_{class}$ promising solutions are selected according to the predicted label and criterion ACD. The criterion ACD is defined as following:

$$ACD = (2 - \alpha \ast \theta) \ast Dic$$

(3)

where $Dic$ and $\theta$ are the predicted values of label $Distance$ and $Angle$, respectively. Obviously, a small $Dic$ could indicate a better convergence of solution. Next, we focus on the introduction about how $\theta$ promotes diversity and the search of uncertain region.

The reference solutions have a relatively better convergence in current generation, which are probably chosen to the next generation training sets and the region far from reference solutions may miss corresponding training data.

Label $\theta$ is helpful to select the solutions far from the reference solutions, where may be located at uncertain region, therefore the solutions with bigger $\theta$ can be helpful to search uncertain region. Moreover, a bigger $\theta$ is helpful for diversity, because in some scenarios, the solutions far from the origin may have better convergence than those near the origin, therefore bigger $\theta$ gives an opportunity to select solutions with good convergence but further distance to the origin. Parameter $\alpha$ represents the importance degree of label $\theta$ on ACD. In this paper, we set it to 0.2 empirically.

After the promising solutions are selected from each class, we need to predict their values of label $Class$. Then the promising solutions are re-assigned to each class and treated as the parents of next generation. The promising solutions are re-assigned because even if two parents are in the same class, their offspring may not belong to this class.

The above process is repeated until reach maximum number of iterations $w_{max}$ as shown in algorithm 4. The procedure of coevolution is shown in Fig. 6. By this way, $K$ solutions will be evaluated by objective functions at each generation.

Algorithm 4 CoEvolution ($P, K, model, w_{max}$)

Input:
$P$(population used to create surrogate in this generation),
$K$(number of classes), $model$(three surrogates), $w_{max}$(maximum number of iterations)

Output:
$Q$(last generation population of coevolution)

1: $w = 1$
2: while $w < w_{max}$ do
3:   $[Class, Dic, \theta] \leftarrow PredictLabel (model, P, K)$
4:   $Q \leftarrow \emptyset$
5:   for $i = 1:K$
6:     $O \leftarrow$ generate offsprings based on the parent population $P$, where $P$ consists of the solutions whose predicted label $Class$ is $i$.
7:     $R \leftarrow$ select $n_{class}$ solutions with minimum ACD value from $P \cup O$.
8:   end
9:   $Q \leftarrow [Q, R]$ /*save the promising solutions from all classes*/
10:  $w = w + 1$
11: end

Figure 6. Procedures of coevolution in ACDEA.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
E. SELECTION OF TRAINING SET

In SAEAs, it is necessary to select some solutions that need to be re-evaluated at each generation. These selected solutions are useful to update surrogates because they can expand the selection range of the training sets. To be specific, the detailed process of such selection is as follows:

When the coevolution reaches its maximum number of iterations, the last generation population $Q$ is divided into $K$ classes according to the multi-classification surrogate. In each class, one solution with minimum ACD surrogate will be selected, these $K$ selected solutions are re-evaluated by real (but expensive) objective functions. Note that, the selection of solutions does not use uncertainty information such as the average of the standard deviations obtained from Kriging models, because we select the solution with large label $\theta$, which is helpful to search the unexplored region.

As indicated in [37], the computational complexity of training Kriging is $O(n^3)$, where $n$ is the number of training data. Therefore, in order to reduce the computation time of training surrogates, the amount of training data need to be limited. In this paper, the maximum number of training data is set to $11d − 1$, where $d$ is the number of decision variables. We use the same method as the selection of reference solutions (Algorithm 2) to select the training sets for next generation surrogates.

IV. EXPERIMENTAL STUDY

In this section, we conduct an experimental study to validate the performance of ACDEA in dealing with multi-objective problems. Three state-of-the-art MOSAEAs, i.e., KRVER [21], CSEA [24], and ParEGO [19], are employed as the compared algorithms over the proposed ACDEA. A set of multi-objective benchmark functions are selected from DTLZ [38], where the number of objectives is set to be 3, 4, 6, 8, and 10, and the number of decision variables is set to be 10.

In the experiments, the statistic results are obtained by each algorithm via 25 independent experiments. The Wilcoxon test is employed to show the significant difference of compared algorithms, where symbols “+” means the compared algorithms have a better performance than ACDEA, symbols “−” means the compared algorithms are worse than ACDEA and symbols “=” means there is no significant difference between the compared algorithms and ACDEA.

The other parameter setting are as follows: the Kriging is used as the surrogate which is implemented in DACE toolbox [39], and all experiments are carried out on PlatEMO [40].

A. PARAMETER SETTINGS

1) Maximum number of function evaluations (FEs), $FE_{\text{max}} = 300$.
2) Number of solutions evaluated by real expensive fitness function at the initial stage = maximum number of training set for one surrogate model, i.e., $N = 11d − 1$.
3) Number of reference solutions = number of classes for objective search space, i.e., $K = 6$.
4) Maximum number of iterations in coevolution before surrogate update: $\max_{\text{iter}} = 20$.
5) Parameter for the criteria ACD, $\alpha = 0.2$.
6) Parameters for the other algorithms:

   For ParEGO, as recommended in [19], the number of weight vectors is set to be 11 when the number of objectives is 2, and it is set to be 15 when the number of objectives is 3. The maximum number of evaluations using surrogates before updating the surrogate is set to be 200 000.

   For K-RVEA, as recommended in [21], the parameter $\delta$ is set to be 0.05$N$, where $N$ is the number of populations. The number of iterations of RVEA using the same surrogates before the surrogate updated is set to be 20. The number of solutions to be re-evaluated before the surrogates are updated is set to be 5.

   For CSEA, as recommended in [24], the maximum number $g_{\text{max}}$ of using surrogate before updating is set to be 3000, the number of reference solutions $k$ is set to be 6, and the number of hidden neurons $H$ is set to be 10.

B. PERFORMANCE INDICATOR

In this paper, we use IGD [41] as the metric to evaluate the performance of the algorithms. In principle, IGD can evaluate both convergence and diversity performance of the obtained solutions. For an algorithm, a smaller IGD value means a better quality of solutions for approximating the PF.

To be specific, IGD can be defined as

$$\text{IGD}(P^*, \Omega) = \frac{\sum_{x \in P^*} \text{dis}(x, \Omega)}{|P^*|}$$  \hspace{1cm} (4)

where $P^*$ is a set of evenly distributed reference points along the true PF, $\Omega$ is the set of achieved non-dominated solutions, and $\text{dis}(x, \Omega)$ is the minimum Euclidean distance between $x$ and the points in $\Omega$. In this paper, the number of reference points is set to be around 10 000.

C. PERFORMANCE ON DTLZ PROBLEMS

The experimental results over 25 independent runs are listed in Table I, where the best results are bold. There are no results for ParEGO when the number of objectives is more than four, because the authors of ParEGO limited it to lower than four objectives. The excess part is denoted by “NA”.

As shown in Table I, we can see that ACDEA achieves the best results on the 16 test problems, nearly half of all test problems. These results show the competitiveness of ACDEA over its compared algorithms KRVEA and
PAREGO. In the following, we will analyze the test results obtained by the algorithms on each test function as follow:

For DTLZ1 and DTLZ3 with multimodal landscapes, the static results obtained by the algorithms can be found in Table I. For sake of clarity, the nondominated solutions obtained by the algorithms of the run producing the median IGD values on three and ten-objective DTLZ1 are illustrated in Fig. 7 and Fig. 8. As shown in Fig. 7, we can clearly observe that the convergence of ACDEA in the circle region is better than in the other regions, and the convergence of the other algorithms is similar in the whole region. Such performance gap between ACDEA and its compared algorithms may be due to the coevolutionary framework used in ACDEA. In ACDEA, the search space is divided into several classes by reference solutions, and the coevolution could reduce the interference between populations from different classes. Therefore, the slow convergence of one population will not affect other populations.

However, we can also observe from Table I that the IGD values obtained by all algorithms are very large, showing poor convergence performance. Furthermore, these algorithms need more function evaluations to reach satisfactory results when solving these test problems.

For DTLZ2 and DTLZ4, whose PFs are similar to each other, the final nondominated solutions obtained by the compared algorithms of the run producing the median IGD values on three-objective DTLZ2 instance are show in Fig. 9. As shown in the figure, ACDEA has a better convergence than the other algorithms. This is mainly because that the criteria of ACD used in ACDEA is able to accelerate the convergence performance of the algorithm.

The final nondominated solutions obtained by the compared algorithms of the run producing the median IGD values on three and ten-objective DTLZ4 instance are show in Fig. 10 and Fig. 11. As shown in the figure, ACDEA achieves better diversity compared with CSEA, but worse diversity compared with KRVEA. This is because KRVEA establishes surrogate for each objective, which making the classification results more accurate. But compared with KRVEA, ACDEA controls the number of surrogates when solving MaOPs.

| Problem | Obj. | ACDEA | KRVEA | CSEA | ParEGO |
|---------|------|-------|-------|------|--------|
| DTLZ1  | 3    | 2.5120e+1(1.24e+1) | 8.0954e+1(1.47e+1) | 5.7691e+1(1.05e+1) | 6.8636e+1(9.89e+0) |
|        | 10   | 1.8266e+1(1.26e+2) | 3.5039e+1(1.04e+1) | 2.8755e+1(1.52e+1) | NA |
| DTLZ2  | 3    | 9.5190e+2(1.49e-2) | 1.1602e+1(1.60e-2) | 2.2119e+1(1.07e-2) | 3.6092e+1(3.42e-2) |
|        | 10   | 5.8486e+1(5.23e-2) | 5.0466e+1(1.70e-2) | 6.7915e+1(1.93e-2) | 4.3845e+1(3.15e-2) |
| DTLZ3  | 3    | 7.7417e+2(1.96e+1) | 2.3326e+2(1.07e+1) | 1.4776e+2(2.98e+1) | 1.7121e+2(1.15e+1) |
|        | 10   | 8.5386e+2(1.39e+1) | 1.6317e+2(5.57e+1) | 1.1248e+2(2.80e+1) | 1.5404e+2(9.05e+0) |
| DTLZ4  | 3    | 1.4808e+2(1.33e-2) | 9.0755e-2(9.22e-2) | 1.1640e-1(3.40e-2) | 2.8589e-2(2.05e-2) |
|        | 10   | 1.8157e+2(1.23e-1) | 1.3060e+0(5.72e+1) | 1.0885e+0(2.85e+1) | 5.6082e+0(1.30e+1) |
| DTLZ5  | 3    | 4.2721e-1(1.02e-1) | 3.2312e-1(1.57e-2) | 5.0078e-1(1.45e-1) | 6.441e-1(1.70e-2) |
|        | 10   | 6.5355e-1(1.28e-1) | 5.2439e-1(5.73e-2) | 4.3206e-1(5.09e-2) | 7.0201e-1(5.21e-2) |
| DTLZ6  | 3    | 3.6902e+2(1.95e-2) | 6.4876e-2(3.53e-2) | 1.2305e-1(2.99e-2) | NA |
|        | 10   | 2.6039e+2(2.22e-1) | 3.6195e-2(2.19e-2) | 7.8641e-2(2.77e-2) | NA |
| DTLZ7  | 3    | 1.4242e+1(1.17e+1) | 2.1978e+1(1.18e+1) | 1.5114e+1(7.59e+0) | 4.9749e+1(1.75e+1) |
|        | 10   | 8.5150e+2(1.23e-1) | 1.3060e+0(5.72e+1) | 1.0885e+0(2.85e+1) | 6.441e+0(1.70e+1) |
|        | 3    | 3.8150e+0(1.14e+0) | 3.0619e+0(2.55e-1) | 4.8496e+0(8.05e-1) | 5.9623e+0(3.38e-1) |
|        | 10   | 3.9765e+0(3.38e-1) | 2.7146e+0(2.31e+1) | 5.1279e+0(3.23e+1) | 5.6243e+0(4.30e+1) |
|        | 3    | 1.4688e+0(8.32e-1) | 1.3279e+0(1.43e+1) | 3.2107e+0(4.86e-1) | NA |
|        | 10   | 5.2612e-1(3.53e-1) | 5.0775e-1(2.05e-1) | 1.4548e+0(4.24e-1) | NA |
|        | 3    | 5.9535e-1(6.09e-2) | 1.3486e-1(2.18e-2) | 1.5536e+0(1.04e+0) | 6.4555e-1(1.94e-0) |
|        | 10   | 6.6110e-1(1.22e-1) | 4.8588e-1(9.81e-2) | 2.0414e+0(8.39e-1) | 7.5014e-1(6.53e-1) |
|        | 3    | 1.1144e+0(7.78e-2) | 5.9276e-1(6.44e-2) | 5.1764e-0(1.30e-0) | NA |
|        | 10   | 1.2439e+0(2.52e-2) | 1.0517e+0(2.15e-1) | 6.7424e+0(2.61e+0) | NA |
|        | 3    | 1.4349e+0(7.75e-2) | 1.0975e+0(2.73e-2) | 1.8074e+0(2.33e-1) | NA |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Figure 7. Nondominated solutions obtained with ACDEA, K-RVEA, CSEA, and ParEGO of the run with the median IGD value for three-objective DTLZ1.

Figure 8. Final nondominated front obtained by each algorithm of the run with the median IGD value for the 10-objective DTLZ1, shown by parallel coordinates.

Figure 9. Nondominated solutions obtained with ACDEA, K-RVEA, and CSEA of the run with the median IGD value for three-objective DTLZ2.
Figure 10. Nondominated solutions obtained with ACDEA, K-RVEA, and CSEA of the run with the median IGD value for three-objective DTLZ4.

Figure 11. Final nondominated front obtained by each algorithm of the run with the median IGD value for the 10-objective DTLZ4, shown by parallel coordinates.

Figure 12. Nondominated solutions obtained with ACDEA, K-RVEA, and CSEA of the run with the median IGD value for three-objective DTLZ5.

Figure 13. Final nondominated front obtained by each algorithm of the run with the median IGD value for the 6-objective DTLZ5, shown by parallel coordinates.
For DTLZ5 and DTLZ6, their PF shapes are degenerated curves, which indicates that their PFs only cover a small subspace of the objective space. Fig. 12 and Fig. 13 show the nondominated solutions obtained by the SAEAs in one single run on three- and six-objective DTLZ5 instances. When KRVEA is used to solve this problem, most of its reference vectors have no solutions to be assigned with them, too many inactive vectors are generated, which leads to the slow convergence. However, in ACDEA, the population is classified into sub-populations according to the guidance of the reference points, which is more consistent with the characteristics of the problem’s PF, and each class can easily locate at least one solution assigned to it. In this way, the convergence will not be disturbed during the search. The nondominated solutions obtained by the algorithms of the run producing the median IGD values on three-objective DTLZ6 are shown in Fig. 14. As can be seen, none of the involved algorithms can accurately approximate the PF, due to the poor convergence performance. The above experimental results also show that it is needed to increase the number of function evaluations for these algorithms to solve such problem.

For DTLZ7, the PF is discontinuous, the final nondominated solutions obtained by the compared algorithms of the run producing the median IGD values on three-objective DTLZ7 instance are show in Fig. 15. As shown in the figure, KRVEA achieves the best performance, ACDEA achieves better diversity than CSEA because of
the use of the multi-classification model which assigns individuals of different diversity into different classes.

In order to further prove the performance of our algorithm, ACDEA is tested on WFG [42] and compared with KRVEA, CSEA and ParEGO. The number of objectives is set to 3, the position parameter in WFG is set to 8, and the maximum number of function evaluations set to 250. The other parameters of the algorithms are set as the above.

The IGD values obtained by four algorithms on WFG1-9 are given in Table II. As shown, we can find that ACDEA performs much better in solving WFG1 and WFG2, but performs worse on WFG3 compared with other algorithms. Note that WFG3 has a degenerated front. Although the classification method is used in ACDEA is helpful to solve such problem, there are still many solutions in Pareto set of WFG3, which are far from the origin in the objective space. For ACDEA, label Angle is helpful to find the solutions far from the origin, but it can’t find all of them, which may lead to poor diversity and lower IGD. WFG4-WFG9 have several challenges in the decision space. For these problems, three algorithms have shown their own advantages, as shown in Table II.

| Table 2. Statistic results of four algorithms on WFG. |
|----------------|----------------|----------------|----------------|
|                | ACDEA          | KRVEA          | CSEA           | ParEGO         |
| WFG1           | 1.60e+0        | 1.83e+0        | 1.76e+0        | 2.29e+0        |
| WFG2           | 8.17e-1        | 9.60e-1        | 9.11e-1        | 1.21e+0        |
| WFG3           | 3.65e-1        | 1.21e-1 +      | 3.08e-1        | 5.98e-1        |
| WFG4           | 7.99e-1        | 7.47e-1        | 1.06e+0        | 9.90e-1        |
| WFG5           | 6.00e-1        | 5.43e-1        | 6.45e-1        | 8.15e-1        |
| WFG6           | 8.69-1         | 4.31e-1 +      | 5.71e-1 +      | 8.22e-1 =      |
| WFG7           | 6.56e-1        | 8.09e-1 -      | 8.98e-1 -      | 1.15e+0 -      |
| WFG8           | 9.83e-1        | 9.85e-1 -      | 1.14e+0 -      | 1.29e+0 -      |
| WFG9           | 5.93e-1        | 5.08e-1 -      | 4.98e-1 -      | 7.57e-1 -      |

D. INFLUENCE OF THE NUMBER OF REFERENCE SOLUTIONS

In this section, we analyze the influence of the parameter $K$ over the performance of the proposed ACDEA in dealing with multi-objective problems. For ACDEA, the parameter $K$ not only represents the number of classes of the population, but also denotes the number of new training data (i.e., the solutions that are re-evaluated by real objective functions), which is obtained before the surrogates are updated. This also affects the total number of iterations of the algorithm. As a result, the IGD values obtained by ACDEA with different $K$ on DTLZ6 are shown in Table III.

| Table 3. Statistic results of different $K$ on 3 test instances. |
|----------------|----------------|----------------|----------------|
|                | M=3            | M=6            | M=10           |
| $K$            | 3              | 6              | 8              | 10             | 20             |
| M=3            | 3.95           | 3.82           | 3.41           | 3.88           | 4.06           |
| M=6            | 2.06           | 1.47           | 1.97           | 2.18           | 2.43           |
| M=10           | 0.14           | 0.13           | 0.12           | 0.14           | 0.24           |

From the experimental results, we can observe that when $K$ is 6 or 8, the obtained result is best. When $K$ is too small, the number of classes that the objective space has been divided into is very small, which makes each sub-space that is allocated by each class very large. ACDEA only selects one solution to be re-evaluated from each class at each generation, the selected solution can not well represent this class, which makes the diversity very poor. Therefore, too small $K$ is not practical.

When $K$ is too large, multiple individuals are re-evaluated at each generation, which makes the total number of iterations smaller. The smaller number of surrogate-updating iterations indicates that the surrogates cannot reach the ideal level. Therefore, $K$ cannot be too large. In this paper, we set $K$ to be 6.

E. EFFECTIVENESS OF LABEL CLASS

As described in III-C, in the process of selecting the reference solutions, all individuals are clustered into $K$ classes through the reference vectors before the selection of reference solutions. However, we do not directly use this clustering result as the result of the label Class and use the reference solutions instead. First, the reference solution is a necessary condition for calculating the label $\theta$, which is helpful to select the promising individuals from each class. Secondly, the clustering by reference vectors does not take into account the convergence of individuals, which makes some individuals far away from the pareto front interfere with the classification results. The reference solutions are the individuals with good convergence in each class. Classifying according to these individuals can avoid the interference of individuals with poor convergence.

In order to prove our point, we compare the classification methods using reference vectors and reference solutions. As a result, the IGD values obtained by ACDEA with classification methods on DTLZ6 are shown in Table IV. All other experimental parameters are set to be the same as those in Table I.

| Table 4. Statistic results of different classification methods on 3 test instances. |
|----------------|----------------|----------------|
| classification methods | reference solutions | reference vectors |
| M=3            | 3.82           | 3.90           |
| M=6            | 1.47           | 1.66           |
| M=10           | 0.13           | 0.19           |

From the experimental results, we can observe that the classification method using reference solutions achieves better results.

F. RUNTIME COMPARISON

As discussed in Section II-B, constructing surrogates is an important computation-consumed process for the implementation of MOSAEs. In this section, we will investigate the computation cost of constructing surrogate in the involved algorithms. Especially, we record and compare the runtime of the algorithms under the same computation environment with the same number of FEs. This is feasible because the evaluation time of the objective function used in the experiment is very small, and
accordingly the algorithms’ runtime is approximately equal to the computation cost of constructing surrogates.

In most existing MOSAEAs, the surrogates are used to approximate the exact value of the expensive objective function or the fitness function, which always have an ascendant training time as the number of objectives increases. For the control of training time, ACDEA constructs three surrogates at one generation (before surrogates are updated) no matter how many objectives the problem involves.

In order to study the computational efficiency of ACDEA, we compare ACDEA with the other SAEAs on DTLZ3 and DTLZ7 with different number of objectives. The detailed results are shown in Fig. 16 and 17.

It can be observed that the runtime of KRVEA increases as the number of objectives increases, because KRVEA creates surrogate for each objective, resulting in a longer time for the construction of surrogates. The runtime of CSEA does not increase with the number of objective, because CSEA uses the feed forward neural networks as the surrogate and only use one surrogate at each generation. Encouragingly, the runtime of ACDEA is the lowest in most test problems among the compared algorithms, due to the fact that it only needs three surrogates to be constructed and maintained at each generation, even in the case of multi-objective problems with many objective functions (e.g., more than ten objectives).

It can be shown from Fig. 16 and 17 that the runtime of ACDEA will be decreased as the number of objectives increases. This may be due to that as the number of objectives increases, the region of the objective search space assigned to each class is becoming larger, which means there are more opportunities to get a promising solution to be re-evaluated in each class at one iteration. Therefore, at each iteration, it is more easily to get $K$ solutions to be re-evaluated by real expensive function since each class probably provide one solution, which makes the total number of iterations decreases.

![Figure 16. Runtime comparison of ACDEA, CSEA and KRVEA on DTLZ3 for 3, 4, 6, 8 and 10 objectives](image)

![Figure 17. Runtime comparison of ACDEA, CSEA and KRVEA on DTLZ7 for 3, 4, 6, 8 and 10 objectives](image)

V. CONCLUSION

In this paper, a new surrogate-assisted evolutionary algorithm called ACDEA is proposed to deal with the expensive many-objective optimization problems. ACDEA uses the guidance of the reference solutions to divide the objective search space into several classes, and then three surrogates are constructed to predict the class labels of these solutions in the population, where the multi-classification surrogate is employed to classify the population, and the other two surrogates are used to find promising solutions in each class.

In addition, a coevolutionary framework is adopted to search for the promising solutions with the surrogates, which can reduce the bad influence of poor convergence class on the other classes. The ACD criteria is used to evaluate and compare the solutions, which can not only balance convergence and diversity but also search the uncertain region.

The effectiveness of ACDEA is validated by the comparison experiments with KRVEA, CSEA and PAREGO on a set of benchmark problems from DTLZ and WFG. The computation cost of constructing surrogates is lower compared with the other algorithms. This is mainly because that ACDEA controls the number of surrogates at each generation. Experimental results have shown the efficiency of the proposed ACDEA.

REFERENCES

[1] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” IEEE Trans. Evol. Comput., vol. 3, no. 4, pp. 257–271, 1999.
[2] H. Lu, Y. Liu, S. Cheng and Y. Shi, “Adaptive online data-driven closed-loop parameter control strategy for swarm intelligence algorithm”, Inf. Sci., vol.536, pp.25-52, 2020.
[3] H. Ishibuchi and T. Murata, “A multi-objective genetic local search algorithm and its application to flowshop scheduling,” IEEE Trans. Syst., Man, Cybern. C, vol. 28, no. 3, pp. 392–403, 1998.
[4] L. Ma, X. Wang, M. Huang, Z. Lin, L. Tian and H. Chen, “Two-Level Master-Slave RFID Networks Planning via Hybrid Multiobjective Artificial Bee Colony Optimizer,” IEEE Trans. Syst.,
Man, Cybern. Syst., vol. 49, no. 5, pp. 861–880, 2019.

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

[6] G. Wang and Y. Tan, “Improving metaheuristic algorithms with information feedback models,” IEEE Trans. Cybern., vol. 49, no. 2, pp. 542–555, 2019.

[7] L. Ma, S. Cheng and Y. Shi, “Enhancing Learning Efficiency of Brain Storm Optimization via Orthogonal Learning Design,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 51, no. 11, pp. 6723-6742, Nov. 2021.

[8] K. Yu, J. Liang, B. Qu, Y. Luo and C. Yue, “Dynamic Selection Preference-Assisted Constrained Multiobjective Differential Evolution,” IEEE Trans. Syst., Man, Cybern. Syst., 2021, doi: 10.1109/TSMC.2021.3061698.

[9] J. Sun, S. Gao, H. Dai, J. Cheng, M. Zhou, and J. Wang, “Bi-objective Elite Differential Evolution Algorithm for Multivalued Logic Networks,” IEEE Trans. Cybern., vol.50, no.1, pp. 233-246, 2020.

[10] L. Ma, M. Huang, S. Yang, R. Wang and X. Wang, “An Adaptive Localized Decision Variable Analysis Approach to Large-Scale Multi-objective and Many-objective Optimization,” IEEE Trans. Cybern., 2021, http://dx.doi.org/10.1109/TCYB.2020.3041212.

[11] L. Ma, K. Hu, Y. Zhu and H. Chen, “Cooperative Artificial Bee Colony Algorithm for Multi-objective RFID Network Planning,” J. Netw. Comput. vol.42, pp. 143-162, 2014.

[12] K. Yu, J. Liang*, B. Qu and C. Yue, “Purpose-directed two-phase multiobjective differential evolution for constrained multiobjective optimization,” Swarm Evol. Comput., vol.60, 2021.

[13] J. Zou, W. Liu, J. Zheng, S. Yang and Y. Guo, “A many-objective evolutionary algorithm based on rotated grid,” Appl. Soft Comput., vol.67, pp. 596-609, 2018.

[14] 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.

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

[16] 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.

[17] L. Ma, N. Li, Y. Guo, M. Huang, S. Yang, X. Wang and H. Zhang, “Learning to Optimize: Reference Vector Reinforcement Learning Adaptation to Constrained Many-objective Optimization of Industrial Copper Burdening System,” IEEE Trans. Cybern., http://dx.doi.org/2021, 10.1109/TCYB.2021.3086501.

[18] 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, 2009.

[19] 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.

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

[21] T. Chugh, Y. Jin, K. Miettinnen, 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.

[22] 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, pp. 2883–2890, 2015.

[23] 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.

[24] L. Pan et al., “A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization,” IEEE Trans. Evol. Comput., vol. 23, no. 1, pp. 74–88, Feb. 2019.

[25] H. Wang and Y. Jin, “A Random Forest-Assisted Evolutionary Algorithm for Data-Driven Constrained Multiobjective Combinatorial Optimization of Trauma Systems,” IEEE Trans. Cybern., vol. 50, no. 2, pp. 536–549, 2020.

[26] H. Wang, Y. Jin, C. Sun and J. Doherty, “Offline Data-Driven Evolutionary Optimization Using Selective Surrogate Ensembles,” IEEE Trans. Evol. Comput. vol.23, no.2, pp.203-216, 2019.

[27] G. Box and N. Draper, “Empirical Model-Building and Response Surfaces,” SIAM Rev., vol. 31, no. 1, pp. 137–139, 1989.

[28] D. 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.

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

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

[31] R. Allmendinger, E. Emmerich, J. Hakanen, Y. 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.

[32] M. Pilat and R. Neraud, “An evolutionary strategy for surrogate-based multiobjective optimization,” in Proc. IEEE Congr. Evol. Comput. (CEC), Brisbane, QLD, Australia, pp. 1–7, 2012.

[33] Y. Jin, H. Wang, T. Chugh, D. Guo and K. Miettinnen, “Data-Driven Evolutionary Optimization: An Overview and Case Studies,” in IEEE Transactions on Evolutionary Computation, vol. 23, no. 3, pp. 442-458, June 2019, DOI 10.1109/TEVC.2018.2869001.

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

[35] M. McKay, R. Beckman, and W. 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.

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

[37] J. Hensman, N. Fusi, and N. Lawrence, “Gaussian processes for big data,” in Proc. Conf. Uncertainty Artif. Intell., Bellevue, WA, USA, pp. 282–290, 2013.

[38] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in Proc. IEEE Congr. Evol. Comput., Honolulu, HI, USA, pp. 825–830, 2002.

[39] S. Løhøven, H. Nielsen, and J. Sondergaard, “DACE: A MATLAB Kriging toolbox,” Dept. Informat. Math. Model., Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep. IMM-TR-2002-12, 2002.

[40] 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.

[41] P. Bosman and D. Thierens, “The balance between proximity and diversity in multiobjective evolutionary algorithms,” IEEE Trans. Evol. Comput., vol. 7, no. 2, pp. 174–188, Apr. 2003.

[42] S. Huband, L. Barone, L. While, and P. Hingston, “A scalable multi-objective test problem toolkit,” in Evolutionary Multi-Criterion Optimization, C. A. C. Coello, A. H. Aguirre, and E. Zitzler, Eds. Heidelberg, Germany: Springer, pp. 280–295, 2005.