A Kriging-Assisted Two-Archive Evolutionary Algorithm for Expensive Many-Objective Optimization

Zhenshou Song, Handing Wang, *Member, IEEE,* Cheng He, *Member, IEEE* and Yaochu Jin, *Fellow, IEEE*

***Abstract*—Only a small number of function evaluations can be afforded in many real-world multi-objective optimization problems where the function evaluations are economically/computationally expensive. Such problems pose great challenges to most existing multi-objective evolutionary algorithms which require a large number of function evaluations for optimization. Surrogate-assisted evolutionary algorithms have been employed to solve expensive multi-objective optimization problems. Specifically, a certain number of expensive function evaluations are used to build computationally cheap surrogate models for assisting the optimization process without conducting expensive function evaluations. Infill sampling criteria in most existing surrogate-assisted evolutionary algorithms take all requirements on convergence, diversity, and model uncertainty into account, which is, however, not the most efficient in exploiting the limited computational budget. Thus, this paper proposes a Kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization. The proposed algorithm uses one influential point-insensitive model to approximate each objective function. Moreover, an adaptive infill criterion that identifies the most important requirement on convergence, diversity, or uncertainty is proposed to determine an appropriate sampling strategy for re-evaluations using the expensive objective functions. Experimental results on a set of expensive multi/many-objective test problems have demonstrated its superiority over five state-of-the-art surrogate-assisted evolutionary algorithms.**

***Index Terms*—Kriging, expensive multi-objective optimization, surrogate-assisted, evolutionary algorithms, adaptive sampling strategy.**

I. Introduction

Many-objective optimization problems (MaOPs) encountered in many real world engineering areas involve more than three, often conflicting, objectives [1], whose optima are a set of Pareto optimal solutions (PS). For instance, in hybrid electric vehicle control optimization [2], it contains seven objectives which cannot be reduced. Evolutionary algorithms (EAs) have been widely used in multi-objective engineering design optimization in the past decades due to several advantages they have [3], [4]. For instance, EAs do not need analytical objective functions or constraints, and they are capable of obtaining a set of non-dominated solutions in a single run [5]. Most existing research on multi-objective evolutionary algorithms (MOEAs) [6] require a large number, typically more than 10,000 [7], of function evaluations. Conventional MOEAs often fail to achieve satisfactory results if only a small number of candidate solutions can be evaluated in the optimization process [8], [9]. Taking the air intake ventilation system design problem [10] as an example, it has three optimization objectives where some of them are calculated by computationally expensive computational fluid dynamics (CFD) simulations. Since a single function evaluation using CFD simulations could take hours, only hundreds of expensive function evaluations are available for MOEAs [11]. Such optimization problems can be formulated as data-driven multi-objective optimization problems (MOPs) [12], [13] or expensive MOPs.

The use of surrogate models (also known as meta-models [14]) as the replacement of real evaluations in EAs has been widely investigated for solving expensive optimization problems [15]. EAs adopting surrogate models are called surrogate-assisted EAs (SAEAs) [16]. A variety of machine learning models, including Kriging model (also called Gaussian process regression [17]), response surface method (RSM) [18], artificial neural networks (ANNs) [19], support vector machines (SVMs) [20], and radial basis function networks (RBFNs) [21], are used in SAEAs. Multiple objectives in MOPs pose challenges to both MOEAs and surrogate-based fitness approximation. As the number of objectives increases, the number of non-dominated solutions increases exponentially, the Pareto-based MOEAs cannot provide enough selection pressure to converge to the true Pareto front (PF) [22]. Moreover, the hardness significantly increases for expensive MaOPs due to the conflicts of a large number of objectives and accumulative approximation errors of multiple surrogate models [8].

Many efforts have been made to embed the surrogate model into MOEA frameworks for expensive MOPs. MOEAs work as the optimizers in SAEAs to provide the selection pressure to push the population toward the PF. These optimizers can be roughly classified into three different categories: Pareto-, indicator-, and aggregation-based MOEAs [23]. In the first category of optimizers, candidate solutions are selected according to the Pareto dominance. For example, the classification based pre-selection multi-objective evolutionary algorithm (CPS-

This work was supported in part by the National Natural Science Foundation of China (No. 61976165, 61903178, 61590922, and U20A20306). (*Corresponding author: Handing Wang*)

Z. Song and H. Wang are with School of Artificial Intelligence, Xidian University, Xi'an 710071, China. (e-mail: songzhenshou@gmail.com, hdwang@xidian.edu.cn).

C. He is with Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China. (e-mail: chenghehust@gmail.com).

Y. Jin is with the Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: yaochu.jin@surrey.ac.uk).

MOEA) [24] applies a non-dominated sorting scheme to select solutions into the next generation. The classification based surrogate-assisted evolutionary algorithm (CSEA) [9] employs a Pareto-based MOEA, i.e the radial space division based many-objective evolutionary algorithm [25], as its optimizer. The second category optimizers employ an indicator as their fitness function to evaluate candidate solutions [26], [27], e.g., the S-metric selection-based efficient global optimization (SMS-EGO) [28] uses the S-metric, a hypervolume (HV)-based metric, to select solutions for expensive evaluation. The third category optimizers are the most popularly adopted in SAEAs which decompose the MOP into a number of single-objective problems by reference vectors, such as the Pareto based efficient global optimization (ParEGO) [29], the efficient global optimization-assisted MOEA/D (MOEA/D-EGO) [30], and the Kriging assisted reference vector guided evolutionary algorithm (KRVEA) [8]. Note that the diversity of all aggregation-based optimizers relies on the pre-defined reference vectors. In addition, some SAEAs hybridized the above optimizers, e.g. the hybrid surrogate-assisted many-objective evolutionary algorithm (HSMEA) [31] uses both aggregation- and indicator-based optimizers for selection.

Surrogate models are used for approximating various fitness evaluations in MOEAs. Many modeling methods are proposed to enhance the quality of surrogate models, which can be roughly divided into two categories as follows:

- **Regression model based SAEAs**: Objective functions, the Pareto rank, and scalarized functions can be approximated by surrogate models. For example, ParEGO applies one Kriging model to approximate a randomly selected aggregation function. In the ordinal regression surrogate-assisted evolutionary algorithm (OREA) [32], a Kriging model is employed for approximating the ranks after non-dominated sort. In SMS-EGO, each objective function is approximated by an independent surrogate model. The neural networks assisted steady state indicator-based evolutionary algorithm (NN-SS-IBEA) [33] uses neural networks to estimate the hypervolume contribution. In addition, some approaches are developed to improve model quality. For example, MOEA/D-EGO applies a fuzzy clustering-based modeling method to the decision space for building two or more local surrogate models for each objective. These local surrogate models can be more accurate than the global surrogate model in local areas. KRVEA groups the training data into several clusters according to their relationship to the reference vectors, then it randomly selects one data point from each cluster for building surrogate models. In HSMEA, each objective function is approximated using four different surrogate models (Kriging, one order RSM, two order RSM, and RBFN models) and the model with a minimum root mean squared error (RMSE) is selected to approximate the current objective function.
- **Classification model based SAEAs**: Several algorithms use classifiers as surrogate models. CPS-MOEA applies a classification tree to filter out possible 'negative' candidate offspring solutions for saving function evaluations. Furthermore, the classification-based pre-selection is also applied to MOEA/D to solve expensive MOPs [34]. In order to enhance the performance of the classifier, CSEA employs one single feed-forward neural network (FNN) model to predict the dominance relation between the candidate solution and the reference solutions, then it uses a validation set to estimate the prediction reliability.

Apart from the above efforts on enhancing the quality of surrogate models, designing a proper sampling criterion is another way of improving the performance of SAEAs. Infill sampling criteria are used to select the offspring solution with improvements in convergence, diversity, and model accuracy. Most existing infill sampling criteria in SAEAs take one or two aspects of improvement into account. For example, ParEGO and MOEA/D-EGO use the expected improvement (EI) [35] as the infill sampling criterion. In this criterion, the solution with the best EI will be selected for re-evaluation, which considers both convergence and model uncertainty. As considered in [12], the sample that has large uncertainty of its predicted fitness is located in a space where the surrogate model has low confidence in the prediction and has not been fully explored by the EA. Thus, sampling these solutions can improve the approximation accuracy of the surrogate model and promote exploration of potentially promising regions in the search space. SMS-EGO, an improved ParEGO algorithm, uses an HV based metric as the infill sampling criterion which considers both convergence and diversity. In CSEA, offspring solutions with better prediction will be selected for re-evaluation, which considers convergence. HSMEA uses the Euclidean distance between the candidate solution and the ideal point for selection, which considers convergence. Moreover, KRVEA considers both convergence and uncertainty in its infill sampling strategy. It uses the number of inactive reference vectors, i.e. the reference vectors without any associated solution, to measure the diversity of surrogate models. When the diversity degenerates, the solution with the maximum amount of uncertainty, usually measured by the standard deviation (STD) in the Kriging model, is chosen as the infilling sample. Otherwise, offspring solutions with best angle penalized distance (APD) will be selected for re-evaluation.

Although the above SAEAs have achieved satisfactory performance on expensive MOPs, many challenges remain. One is the model sensitivity to training data, especially influential points. A training data point is influential if it unduly influences any part of a regression model [36], [37], e.g. the predicted responses. Influential points in the training data seriously lower the global accuracy of the surrogate model. Although some Kriging models remove the influential points from the training data [8], [30], the model may lose the useful information provided by influential points. The second challenge is the choice of the optimizer. As the number of objectives increases, Pareto-based MOEAs fail to distinguish solutions while the HV based MOEAs, a popular class of indicator-based MOEAs, are hampered by an exponentially increasing computational cost [38]. Furthermore, the aggregation-based MOEAs are sensitive to reference vectors

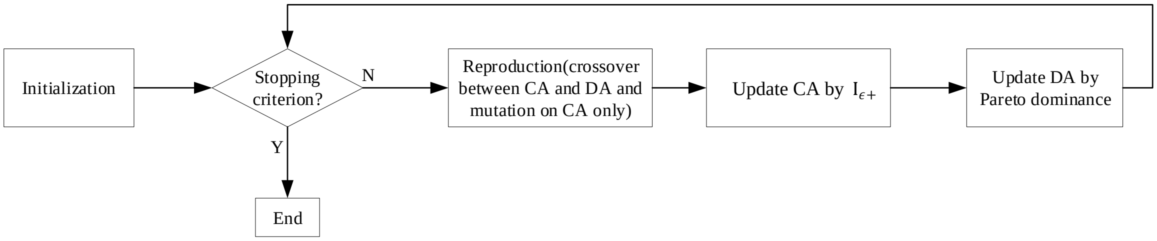

Initialization
Stopping criterion?
N
Y
Reproduction(crossover between CA and DA and mutation on CA only)
Update CA by $I_{\epsilon+}$
Update DA by Pareto dominance
End

Fig. 1: Flowchart of Two_Arch2.

which are hard to pre-define. The third challenge is how to design an effective infill sampling criterion to select candidate solutions for re-evaluation. The infill sampling criteria in most existing SAEAs for MOPs use mixed indicators as the measurement of potential benefits to their performance on convergence, diversity and model accuracy. They do not explicitly consider the changing requirements in different search stages. Infill sampling which taking into account of the subtle changes in priority is able to achieve a better balance between exploration and exploitation. Moreover, how to properly use the uncertainty information is still an open question for SAEAs. Therefore, we propose a Kriging-assisted Two_Arch2 (KTA2) algorithm for solving expensive MOPs/MaOPs. We adopt the non-RV-based Two_Arch2 [39] as the optimizer. Two_Arch2 maintains two archives (termed convergence archive (CA) and diversity archive (DA)) to separately promote convergence and diversity, which are updated by an $I_{\epsilon+}$ based indicator and an L_p-norm distance, respectively. To reduce the negative effect of influential points, we propose an influential point-insensitive Kriging model to replace most expensive function evaluations during the optimization process. Also, we design an adaptive sampling strategy to guide the infill sampling to make a full use of the limited number of evaluations. The main contributions of this paper can be summarized as follows:

1) Two_Arch2 is introduced into the proposed algorithm as an optimizer which generates two sets of candidate solutions. It assures that KTA2 can obtain good performance on convergence and diversity without any manual setting in advance. Moreover, two sets of candidate solutions contain the potential information of the optimization state, which can be exploited in the adaptive sampling strategy.
2) An influential point-insensitive model is proposed to improve the prediction accuracy. The influential point-insensitive model can reduce the negative impact of influential points and without completely losing the information about influential points at the same time.
3) An adaptive sampling strategy is designed for guiding the solution selection by taking account of the requirement on convergence, diversity, and model uncertainty separately. It consists of an optimization state assessment and three different sampling strategies. A tailored sampling strategy is selected for each optimization state according to the detected optimization state.

The remainder of this paper is organized as follows. In Section II, an introduction to MOPs, Two_Arch2, Kriging model, and influential points is given. Section III gives a detailed description of the proposed algorithm. The experimental results and analysis are given in Section IV. The conclusion and future work are provided in Section V.

II. Preliminaries

A. Multi/Many-Objective Optimization Problems

A minimization MOP can be formulated as:

$$\begin{aligned} \text{minimize} \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_m(\mathbf{x}))^T, \\ \text{s.t.} \quad & \mathbf{x} = (x_1, ..., x_d)^T, \end{aligned} \tag{1}$$

where $\mathbf{x} \in R^d$, d is the dimension of decision variables, $f_1(\mathbf{x}), ..., f_m(\mathbf{x})$ are $m (\geq 2)$ objective functions to be optimized. In MaOPs, m is usually larger than 3. Since different solutions may have conflicting performance over each other on different objectives, the concept of the Pareto dominance [40] is critical for comparing solutions of an MOP. To define the Pareto dominance of two solutions $\mathbf{x}$ with their corresponding objective $\mathbf{F}(\mathbf{x})$ and $\mathbf{F}(\mathbf{y})$, $\mathbf{x}$ dominates $\mathbf{y}$ if and only if $\forall i \in \{1, 2, ..., m\}, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\exists j \in \{1, 2, ..., m\}, f_i(\mathbf{x}) < f_j(\mathbf{y})$.

B. Improved Two-Archive Algorithm

Two_Arch2 [39] is an improved Two_Arch [41] which was proposed to solve MaOPs. The main idea of Two_Arch2 is to maintain two different archives (termed CA and DA) to independently promote convergence and diversity during the evolutionary search. An flowchart of Two_Arch2 is shown in Fig. 1, where CA is used to provide the selection pressure for pushing the population towards the true PF. DA is used to ensure the population diversity in the high-dimensional objective space. Thus, Two_Arch2 only applies the mutation operations on CA and performs the crossover operation between CA and DA for generating candidate solutions.

In order to enhance convergence for solving MaOPs, Two_Arch2 adopts the $I_{\epsilon+}$ based indicator as the selection criterion for CA, which has an acceptable complexity and good performance on convergence in IBEA [27]. $I_{\epsilon+}$ is a quality indicator which is defined as follows:

$$I_{\epsilon+}(\mathbf{x}_1, \mathbf{x}_2) = \min_{\epsilon}\{f_i(\mathbf{x}_1) - \epsilon \leq f_i(\mathbf{x}_2), 1 \leq i \leq m\}, \tag{2}$$

where m is the number of objectives. $I_{\epsilon+}(\mathbf{x}_1, \mathbf{x}_2)$ gives the minimal distance ϵ that $\mathbf{x}_1$ needs in order to dominate $\mathbf{x}_2$. The Equation (2) can also be written as:

$$I_{\epsilon+}(\mathbf{x}_1, \mathbf{x}_2) = \max(f_i(\mathbf{x}_1) - f_i(\mathbf{x}_2)), 1 \leq i \leq m, \tag{3}$$

Accordingly, the fitness assignment ranks the population members based on Equation (4), where P is the population. $Q(\mathbf{x}_1)$ is a measure of $I_{\epsilon+}$ if $\mathbf{x}_1$ is dominated by other solutions from the population.

$$Q(\mathbf{x}_1) = \sum_{\mathbf{x}_2 \in P \backslash \{\mathbf{x}_1\}} -e^{-I_{\epsilon+}(\mathbf{x}_1, \mathbf{x}_2)/0.05}, \tag{4}$$

The updating steps of CA in Two_Arch2 as below.

Step 1: Add offspring solutions to CA.

Step 2: Delete the extra solutions with the smallest $I_{\epsilon+}$ loss according to Equation (4).

Step 3: Update the $I_{\epsilon+}$ values of the remaining members in CA.

Step 4: Repeat **Step 2** and **Step 3** until CA obtains a fixed number of solutions.

Two_Arch2 selects non-dominated solutions according to similarity and removes extra solutions if DA overflows, and only keeps the non-dominated solutions in DA during the evolutionary search. The non-dominated solutions with small similarity (Lp-norm distance) to the rest of population would be selected for the next generation [42]. The updating steps of DA when the non-dominated solutions overflow are shown as follows.

Step 1: Move boundary solutions which have the maximal or minimal objective values to DA

Step 2: Add one solution with a small similarity to the selected solutions to DA.

Step 3: Repeat **Step 2** until DA obtains enough solutions.

The updating processes of CA and DA are independent with each other in every generation. Compared to Two_Arch, Two_Arch2 uses DA as the final output because the diversity of CA is poor.

C. Kriging Model

For data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, its corresponding output can be represented as $\mathcal{Y} = \{y(\mathbf{x}_1), y(\mathbf{x}_2), \ldots, y(\mathbf{x}_N)\}$. A Kriging model can learn a latent function $f(\mathbf{x})$ by assuming $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \delta_n^2)$ is an independently and uniformly distributed Gaussian noise. For a new input data $\mathbf{z}^* \in \mathbf{R}$, the mean and variance of the approximated $y(\mathbf{z}^*)$ are estimated as [43]:

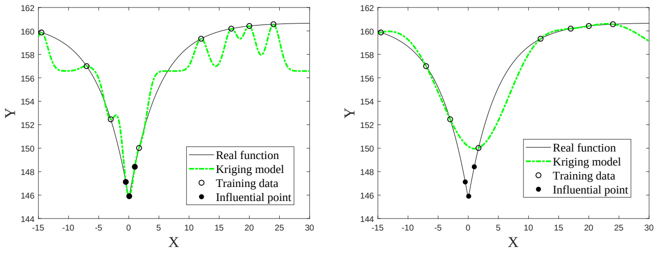
$$\begin{gathered} \bar{f}(\mathbf{z}^*) = \psi(\mathbf{z}^*) + \mathbf{k}^{*\mathbf{T}}(K + \delta_n^2 I)^{-1}(\mathcal{Y} - \psi(\mathcal{X})), \\ \sigma^2[f(\mathbf{z}^*)] = k(\mathbf{z}^*, \mathbf{z}^*) - \mathbf{k}^{*\mathbf{T}}(K + \delta_n^2 I)^{-1}k^*, \end{gathered} \tag{5}$$

where $\psi(\mathcal{X})$ is the mean vector of $\mathcal{X}$, $\mathbf{k}^*$ is the covariance vector between $\mathcal{X}$ and $\mathbf{z}^*$, and K is the covariance matrix of $\mathcal{X}$. In particular, a covariance function is used to measure the similarity between two data points $\mathbf{x}$ and $\mathbf{x}^{'}$. In this work, we use the following squared exponential function as the covariance function:

$$k(\mathbf{x}, \mathbf{x}^{'}) = \theta \exp(-\frac{1}{2l^2}(\mathbf{x} - \mathbf{x}^{'})^T(\mathbf{x} - \mathbf{x}^{'})), \tag{6}$$

where θ is the scale parameter and l is the length-scale parameter for determining the number of modal of the approximated landscape. Note that this covariance function is negatively related to the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}^{'}$. The predicted mean $\bar{f}(\mathbf{z}^*)$ is directly used as the prediction of $y(\mathbf{z}^*)$, and the prediction variance $\sigma^2[f(\mathbf{z}^*)]$ shows the uncertainty. As recommended in [43], the hyperparameters in Equation (5) are learned by maximizing the log marginal likelihood function:

$$\begin{aligned} \log\, p(\mathcal{Y}|\mathcal{X}) = -\frac{1}{2}(\mathcal{Y} - m(\mathcal{X}))^T(K + \delta_n^2 I)^{-1}(\mathcal{Y} - m(\mathcal{X})) \\ -\frac{1}{2}\log|K + \delta_n^2 I| - \frac{N}{2}\log 2\pi. \end{aligned} \tag{7}$$



(a)

(b)

Fig. 2: Illustration of the influence of influential points to original Kriging model. Sub-figure (a) is the illustration of using influential points to build Kriging model. Sub-figure (b) is the illustration of without using influential points to build Kriging model.

D. Effects of Influential Points

Influential point is a general term for the training data that has a greater influence on regression analysis [36], [37], [44]. Deleting or adding influence points will swing the prediction in the regression model. Kriging models also suffer from the influential point. Fig. 2(a) gives an illustration in building the original Kriging model with influential points. We can observe that the approximated curve has multiple local optima in some areas. These local optimum may mislead the population. Mathematically, the reason for these fluctuations is that the optimal parameters obtained by maximizing the log marginal likelihood function are not the global optimal solution. Specifically, the solver obtains a small l in Equation (6) while approximating this area needs a large l. Kriging models always approximate descriptions of more complicated processes, which enhances the effect of influential points but results in the degeneration of the global accuracy [45]. Therefore, the modest modification of the training data becomes important to improve the global accuracy of the Kriging model. Although there are some methods that claim to be able to detect these influential points, the acceptable degree of influence is problem-dependent and must be specified in advance [36], [37]. Moreover, some modified Kriging-modeling methods delete these influential points from the training data [8], [30]. However, directly removing these influential points may cause the model to lose some useful information. As shown in Fig.

2(b), the minimum value of the curve approximated by the original Kriging model without the influence point has been shifted, which will mislead optimization algorithms. Thus, how to properly use influential points is still an open question.

III. Proposed algorithm

In this work, KTA2 is proposed for expensive multi/many-objective optimization. Similar to existing SAEAs, KTA2 consists of three parts: an optimizer, surrogate models, and infill sampling criteria for updating training data. In KTA2, we employ Two_Arch2 as the optimizer to separately consider the performance on the convergence and diversity. The schematic diagram of our algorithm is shown in Fig. 3, which is based on the main framework of Two_Arch2, except for offspring pre-selection and updating surrogate models. KTA2 updates CA and DA in every ten generations (as the setting in [8]) of the right part, where the offspring has been generated using crossover and mutation operations and pre-selected according to the prediction of the surrogate models. η solutions from the pre-selected offspring are chosen for the real function evaluations. Those re-evaluated solutions are used to update CA, DA, and surrogate models. KTA2 can be further divided into six main steps.

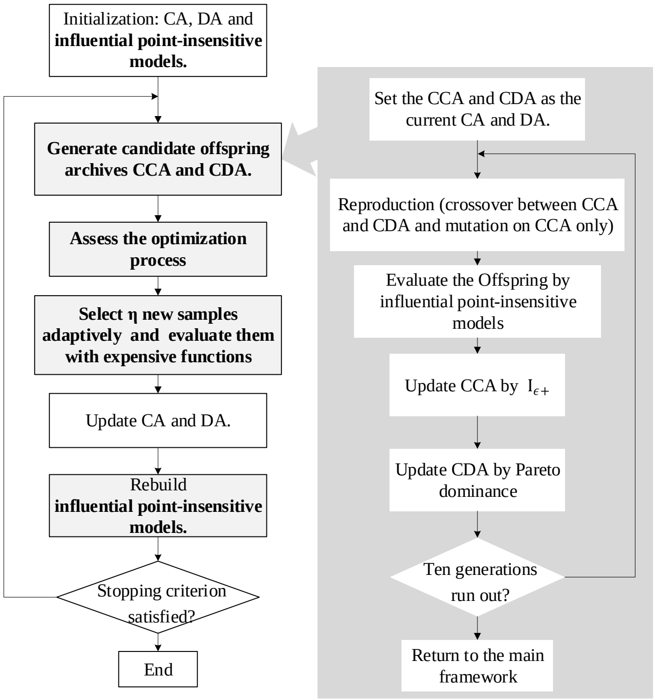

Initialization: CA, DA and influential point-insensitive models.
Generate candidate offspring archives CCA and CDA.
Assess the optimization process
Select η new samples adaptively and evaluate them with expensive functions
Update CA and DA.
Rebuild influential point-insensitive models.
Stopping criterion satisfied?
End
Set the CCA and CDA as the current CA and DA.
Reproduction (crossover between CCA and CDA and mutation on CCA only)
Evaluate the Offspring by influential point-insensitive models
Update CCA by $I_{\epsilon+}$
Update CDA by Pareto dominance
Ten generations run out?
Return to the main framework

Fig. 3: A framework of the proposed KTA2. The left part is the main framework of KTA2. The right part is the evolutionary search for generating candidate archives CCA and CDA.

1) **Initialization**: An initial population is generated using Latin hypercube sampling [46]. These solutions are evaluated by the expensive objective functions and added to CA and DA as Two_Arch2. Moreover, they are used as the training data to build an influential point-insensitive model for each objective function.
2) **Candidate Offspring Generation**: Two candidate archives (candidate CA (CCA) and candidate DA (CDA)) are employed to conduct the optimization process of Two_Arch2 on the approximated MOPs/MaOPs, where the function evaluations are replaced by the influential point-insensitive models. Note that, CCA and CDA are the current CA and DA, respectively, and they will be evolved for a number of generations before they are used to update CA and DA. Same with KRVEA [8], we set 10 generations for this step to ensure the convergence.
3) **Optimization Process Assessment and Adaptive Sampling**: We divide the optimization states into three cases: convergence-, diversity-, and uncertainty-demand states. An assessment strategy is used to distinguish those three states above. In different cases, we adopt different sampling strategies (including STD-based, $I_{\epsilon+}$-based, and L_p-norm distance-based sampling strategies) to select η new samples for real function evaluations
4) **CA and DA Update**: With the new samples, CA and DA are updated by the $I_{\epsilon+}$ based indicator and the L_p-norm distance, respectively.
5) **Influential Point-Insensitive Models Rebuilding** : An influential point-insensitive model for each objective function is rebuilt using all the training data.
6) Repeat Steps 2) to 5) until the computation budget runs out.

In the following sections, we will present the details of the influential point-insensitive models and adaptive sampling strategy.

A. *Influential Point-Insensitive Model*

As mentioned in Section II-D, the training data with a small objective value may cause the Kriging model to generate several local optima. Likewise, a large objective value will produce a similar effect. In this paper, we consider that the training data with overlarge and oversmall objective values are two types of potential influential points, which might affect the Kriging models. It should be noted that 'overlarge' and 'oversmall' are just vague definitions here in order to better understand our method. In this paper, we set a parameter τ as a threshold to define these overlarge and oversmall individuals. Specifically, after sorting in ascending order, the first $(1-\tau)N$ and last $(1-\tau)N$ data points are identified as influential points with oversmall and overlarge objective values after an ascending sorting. A pilot study is conducted in Section IV-B to determine the threshold. In order to reduce the negative effect of influential points on the Kriging models, we propose an influential point-insensitive model. The influential point-insensitive model consists of a model called sensitive model using all the training data, and two models termed insensitive models using the training data without influential points. On one hand, the insensitive model can reduce the influence from one type of influential points. On the other hand, the model can also maintain the information about the other type of influential points. In the influential point-insensitive model, a sensitive model is used to predict the objective values of candidate solutions to determine their relation to those two insensitive models. Then, an insensitive model is selected as the final prediction model.

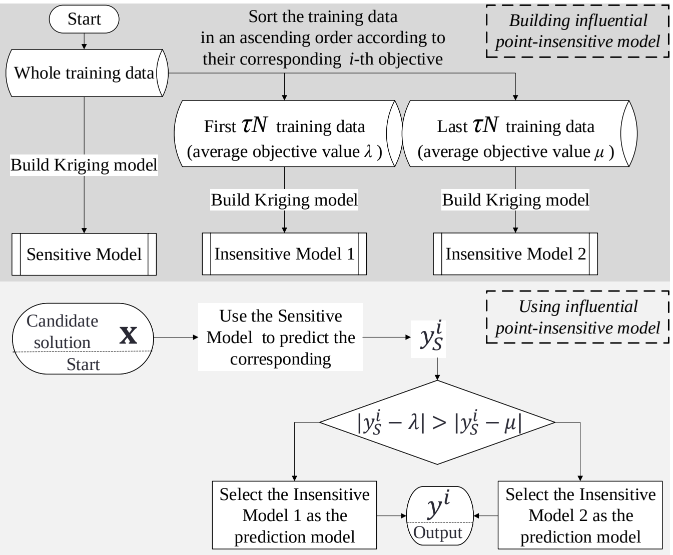

Start
Sort the training data in an ascending order according to their corresponding i-th objective
Building influential point-insensitive model
Whole training data
First τN training data (average objective value λ)
Last τN training data (average objective value μ)
Build Kriging model
Build Kriging model
Build Kriging model
Sensitive Model
Insensitive Model 1
Insensitive Model 2
Candidate solution
$\mathbf{x}$
Start
Use the Sensitive Model to predict the corresponding
y_S^i
Using influential point-insensitive model
$|y_S^i - \lambda| > |y_S^i - \mu|$
Select the Insensitive Model 1 as the prediction model
y^i
Output
Select the Insensitive Model 2 as the prediction model

Fig. 4: Procedure of building and using the influential point-insensitive model to predict the i-th objective value.

The process of building and using the influential point-insensitive model to predict the i-th objective of a candidate solution $\mathbf{x}$ is shown in Fig. 4, where y^i is the final prediction fitness. To build two insensitive models for each objective, KTA2 first sorts all training data in an ascending order according to their objective values, then it separately selects the top τ training data and the last τ training data to build two insensitive models and calculate their average values μ and λ, respectively. The next step is to use the sensitive model built by the whole training data to predict the objective value of $\mathbf{x}$. With the predicted objective value y_s^i obtained by the sensitive model, the insensitive model whose average objective value of training data is closer to y_s^i is chosen as the final prediction model of the candidate solution. A detailed description can be found in Fig. 4

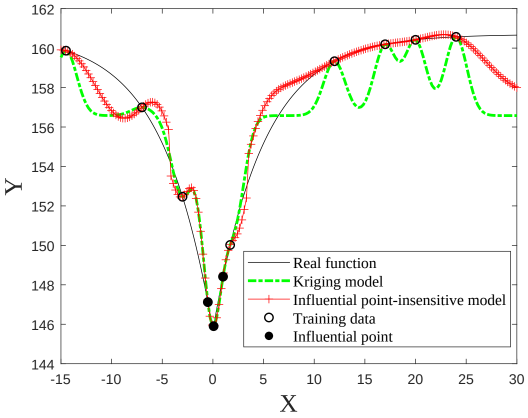


Fig. 5: Illustration of approximating a real problem by the influential point-insensitive model and the original Kriging model, respectively.

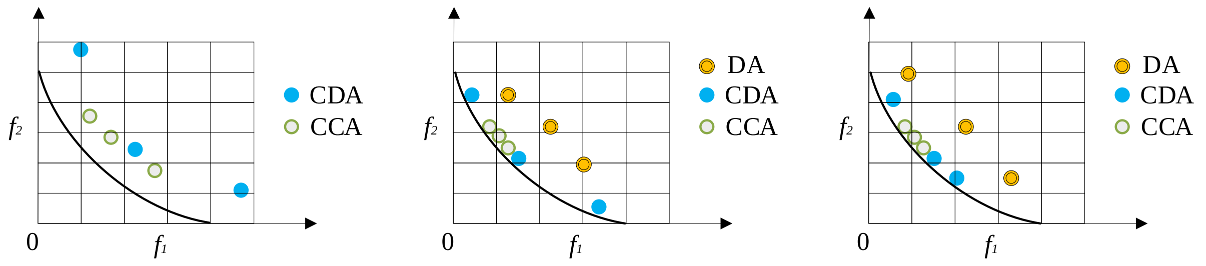
Fig. 5 shows a comparison of the influential point-insensitive model with original Kriging model on an illustrative example. Three influential points of the training data are located in the range [-0.5, 0.5]. The prediction of the original Kriging model in the range [10, 25] produced multiple local optimal regions due to these influential points, while the prediction accuracy of the influential point-insensitive model is obviously better than that of the original Kriging model. In the range [-0.5, 0.5], the information of influential points is also maintained in the influential point-insensitive model. Moreover, without any influential points, the influential point-insensitive model degenerates into an original Kriging model.

Compared with the computational complexity of the original Kriging model, the computational complexity of the influential point-insensitive model has increased. Given N training data, building one Kriging model has a complexity of $O(N^3)$ [47]. Since the influential point-insensitive model contains three Kriging models, the computational complexity is increased from $O(N^3)$ to $O(N^3)+2O((\tau N)^3)$ compared to the original Kriging, where τN is the number of training data of other two insensitive model. The overall computational complexity remains cubic.

B. Adaptive Sampling Strategy

The selection of promising and uncertain solutions for updating the training data brings different improvements to the surrogate model. Sampling promising solutions with good convergence and diversity can enhance the optimization performance in both aspects, while sampling uncertain solutions may improve the global approximation accuracy of surrogate models. However, not all kinds of samples are necessary at different optimization states. In order to fully use the limited number of function evaluations, we propose an adaptive sampling strategy based on the optimization state assessment. As shown in Fig. 6, we divide the optimization state into three cases:

1) **Convergence-demand state**: The improvement of convergence is the priority of KTA2. As shown in Fig. 6(a), both CCA and CDA have not yet converged to the true PF. However, CCA is closer to the true PF than CDA. It is because that CCA provides more selection pressures of convergence than CDA, while CDA pays more attention to the population diversity. In the convergence-demand state, selecting the solution with good convergence can improve the quality of the population.
2) **Diversity-demand state**: The evolutionary process needs to improve the diversity when the solution set is close to the true PF. Fig. 6(b) provides an illustrative example of the diversity-demand state, both CCA and CDA have converged to the true PF. In this state, selecting the solution with good diversity can spread the population over the entire PF evenly.
3) **Uncertainty-demand state**: A global but inaccurate surrogate model may reduce the diversity of candidate archives. As shown in Fig. 6(c), although the solution set is near the true PF, the diversity of CDA is worse



(a) Convergence-demand state (b) Diversity-demand state (c) Uncertainty-demand state

Fig. 6: Three kinds of optimization state. (a) is a convergence-demand state, selecting the solution with good convergence can improve the population quality. (b) is a diversity-demand state, selecting the solution with good diversity can maximize the improvement of the population. (c) is a uncertainty-demand state, selecting uncertain solutions can improve the global accuracy of the surrogate models.

than that of DA. The deterioration of diversity in CDA is most likely caused by the poor global accuracy of the surrogate model. In this state, it is necessary to select uncertain solutions to improve the global accuracy of surrogate models

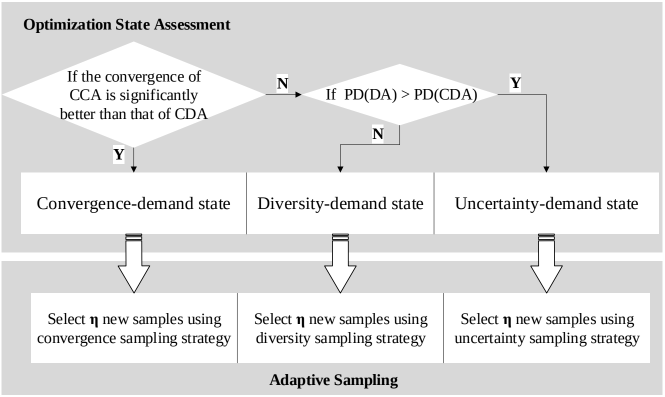

Optimization State Assessment
If the convergence of CCA is significantly better than that of CDA
N
If PD(DA) > PD(CDA)
Y
N
Y
Convergence-demand state
Diversity-demand state
Uncertainty-demand state
Select η new samples using convergence sampling strategy
Select η new samples using diversity sampling strategy
Select η new samples using uncertainty sampling strategy
Adaptive Sampling

Fig. 7: Flowchart of the optimization state assessment and adaptive sampling strategy.

In order to identify the above three optimization states, we use two indicators for convergence and diversity, respectively. To measure the convergence of one archive, we calculate the distances from solutions to the estimated ideal point. In addition, we employ the pure diversity (PD) indicator [48] to assess the diversity of archives. PD is defined as follows:

$$PD(A) = \max_{s_i \in A}(PD(A - s_i) + d(s_i, A - s_i)), \tag{8}$$

where,

$$d(s, A) = \min_{s_i \in A}(\text{dissimilarity}(s, s_i)), \tag{9}$$

As suggested in [48], the dissimilarity is measured by L_p-norm distance, where p is set to 0.1. As shown in Fig. 7, KTA2 first calculates the mentioned distance sets of CCA and CDA and uses the rank-sum test to compare both sets. If the distance set of CCA is significantly shorter than that of CDA, KTA2 identifies the state as the convergence-demand state, otherwise, KTA2 compares the PD values of DA and CDA. If the PD value of CDA is larger than that of DA, KTA2 identifies the state as the diversity-demand state, otherwise as the uncertainty-demand state.

For each optimization state, we employ a tailored sampling strategy to select η new samples for expensive function evaluations:

- **Convergence sampling strategy** deletes the extra solutions with the smallest $I_{\epsilon+}$ loss from CCA until obtaining η convergence improvement solutions. The details can be found in Algorithm 1.
- **Diversity sampling strategy** firstly selects one candidate solution from CDA that is most different from DA based on the L_p-norm distance. Then, it selects η-1 most different solutions one by one as shown in Algorithm 2.
- **Uncertainty sampling strategy** first randomly selects ϕ solutions from CDA. And then the solution that has the largest uncertainty among ϕ solutions will be selected for re-evaluation. Note that the uncertainty is calculated using the average of the STDs obtained from the insensitive models for m objectives. The above selection procedure is repeated until η samples are obtained as shown in Algorithm 3.

Algorithm 1 Convergence sampling strategy

```
Input: CCA, $\eta$.
Output: $\mathbb{A}$.
Set $\mathbb{A}$ equals CCA.
while $|\mathbb{A}| > \eta$ do
    Find the individual $\mathbf{x}^*$ with the minimal $Q(\mathbf{x}^*)$.
    Delete $x^*$ from $\mathbb{A}$.
    Update the remaining individual by $Q(\mathbf{x}) = Q(\mathbf{x}) + e^{-I_{\epsilon+}(\mathbf{x}^*,\mathbf{x})/0.05}$.
end while
```

IV. Empirical Studies

In this section, in order to thoroughly validate the effectiveness of the KTA2, we empirically compare it with some state-of-art SAEAs, including ParEGO [29], MOEA/D-EGO [30], KRVEA [8], CSEA [9], and HSMEA [31]. All the compared

Algorithm 2 Diversity sampling strategy

```
Input: DA, CDA, $\eta$.
Output: $\mathbb{A}$.
Set $\mathbb{A}$ empty
while $|\mathbb{A}| \leq \eta$ do
    Find the individual $\mathbf{x}^*$ from CDA that has the maximal differently to DA.
    Add $\mathbf{x}^*$ to $\mathbb{A}$ and DA, respectively.
end while
```

Algorithm 3 Uncertainty sampling strategy

```
Input: CDA, $\eta$, $\phi$.
Output: $\mathbb{A}$.
Set $\mathbb{A}$ empty.
while $|\mathbb{A}| \leq \eta$ do
    Randomly select $\phi$ individuals from CDA.
    Calculate the uncertainty using the average of the STDs obtained from $m$ influential point-insensitive models.
    Find the individual $\mathbf{x}^*$ with maximal uncertainty from $\phi$ selected individuals.
    Add $\mathbf{x}^*$ to $\mathbb{A}$.
end while
```

algorithms are implemented in PlatEMO [49], except for HSMEA. These compared algorithms are all representatives of SAEAs which can be summarized as follows:

- **ParEGO** is an aggregation-based MOEA. It uses the Chebyshev method to scalarize an MOP into a single-objective problem based on a randomly selected vector in each generation. A Kriging model is employed to approximate the scalarization function. The solution with the best prediction will be selected for re-evaluation.
- **MOEA/D-EGO** is an aggregation-based MOEA, where MOEA/D [50] is used as the optimizer. An MOP is decomposed into several single-objective problems by a set of weight vectors. A fuzzy clustering-based Kriging model is employed to approximate each objective function. For each single-objective problem, the solution with the best EI will be selected for re-evaluation.
- **KRVEA** uses the reference vector guided evolutionary algorithm (RVEA) [51] as the optimizer. It employs a fixed number of training data to build a Kriging model for each objective function. When the diversity degenerates, the solution with the maximum average STD is chosen as the infilling sample. Otherwise, the solution with the best predicted APD will be selected for re-evaluation.
- **CSEA** is a classification based SAEA. It employs a single FNN model to pre-select potentially good solutions to be re-evaluated.
- **HSMEA** is an aggregation-based based MOEA. It uses multiple surrogate models to approximate each objective function. Two sets of reference vectors and a local improvement mechanism for promoting diversity and accelerating convergence, respectively.

The experiments are conducted on 80 test instances taken from test suite DTLZ and WFG [52] with 3, 4, 6, 8, and 10 objectives, respectively. For each test instance, the maximum number of expensive function evaluations is set to 300. As the experimental setting in [31], the numbers of decision variables are set to 10 for all the DTLZ problems, while the numbers of decision variables are set to 10, 10, 9, 9, and 11 for WFG problems with 3, 4, 6, 8, and 10 objectives, respectively.

We use the inverted generational distance (IGD) [53] and the modified IGD (IGD$^+$ [54]) metrics as performance indicators for assessing the performance of the compared SAEAs. Performance comparison results by the IGD$^+$ are always consistent with the Pareto dominance relation [55]. In addition, we also employ PD to measure the diversity of the obtained population of the above-mentioned algorithms. The Wilcoxon rank-sum (WRS) test is also executed to compare the results achieved by KTA2 and other five compared SAEAs under comparison at a significance level of 0.05. Symbols ”$+$” and ”$-$” indicate that the KTA2 is significantly superior to and inferior to the compared SAEAs, respectively. Symbol ”$\approx$” denotes that there is no statistically significant difference between KTA2 and the compared algorithms.

A. Parameter Settings

The common parameter settings of all the compared algorithms are listed below.

- The initial data size and population size are set to 100.
- The maximum number of expensive function evaluations is set to 300.

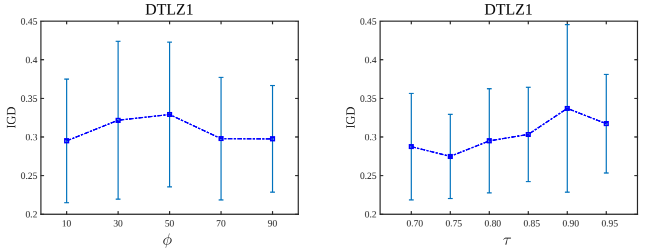
Furthermore, for a fair comparison, we adopt the recommended settings in the original literature for specific parameters of the compared algorithms. The specific parameter settings in KTA2 are shown as below.

- The sizes of each archive (i.e. CA, DA, CCA, and CDA) are set to $N = 100$.
- The number of infilling samples for re-evaluation is set to $\eta = 5$.
- The range of the hyperparameter of Matlab Toolbox DACE [56] for building Kriging models is set to $[10^{-5}, 10^5]$
- The parameters for reproduction (crossover and mutation) are set to $p_c = 1.0$, $p_m = 1/d$, $\eta_c = 20$, $\eta_m = 20$ as same as its set in Two_Arch2.
- The distance calculation in the adaptive sampling strategy is measured by the Manhattan distance.

B. Sensitivity Analysis of Parameters

KTA2 contains two newly introduced parameters, namely ϕ and τ. Note that ϕ is used to define number of randomly selected individuals in Algorithm 3. τ is used to define the proportion of one type non-influential points in the training data. Different τ and ϕ parameter values may affect the performance of KTA2. Here, we through experiments on DTLZ1 with ten objectives to analyze the sensitivity of the newly introduced parameters in KTA2. DTLZ1 is hard to convergence due to its multimodal landscape. That is the reason why we select it as the test instance in this sub-section.

To evaluate the effects of different values of ϕ, we first test a series of values $\phi \in \{0.1N, 0.3N, 0.5N, 0.7N, 0.9N\}$



(a) (b)

Fig. 8: Average IGD profile plots over the different parameter values in KTA2 on the ten objectives DTLZ1 based on 30 independent runs. Subfigure (a) is the performance profile plots of parameter ϕ. Subfigure (b) is the performance profile plots of parameter τ.

when $\tau = 0.80$, where N is the population size. Fig. 8(a) provides the comparison result of KTA2 with different ϕ on 10 objectives DTLZ1 problems based on 30 independent runs. It can be observed that KTA2 with $\phi = 0.1N$ has achieved the best result on DTLZ1, which might be due to the fact that the randomly selecting fewer individuals from CDA has the capability to maintain the diversity. On the other hand, KTA2 with $\phi = 0.9N$ has achieved the second best on DTLZ1, as a larger ϕ has a greater probability of select solutions with greater uncertainty to enhance the global accuracy of the model. Considering that individuals with large uncertainty may may cause the algorithm converged, ϕ is set to $0.1N$ as a general setting for all test instances.

Next, we vary parameter τ to examine the effects of different τ values. In this set of experiments, $\tau \in \{0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$ are tested on DTLZ1 with 10 objectives when $\phi = 0.1N$. The experimental results are shown in Fig. 8(b). It is clear that KTA2 with $\tau = 0.75$ has achieved the best performance. As expected, increase in the value of τ deteriorates the performance. This is because DTLZ1 is a multimodal test problem that brings a lot of influential points. However, the performance also decreases when $\tau = 0.70$, as it defines too many influential points, the normal data are also removed from the training data. Hence, we select $\tau = 0.75$ as a general setting for the rest test instances.

C. *Effects of Influential Point-Insensitive Models*

We first investigate the effects of influential point-insensitive models. In this part, the original Kriging model replaces the surrogate model in KTA2 as a compared variant, termed KTA2(O). The average IGD results of KTA2(O) and KTA2 based on 30 independent runs on DTLZ1 problems with different numbers of objectives are presented in Table I, where the WRS test is also listed and the best results are highlighted. As shown in Table I, KTA2 obtains the best result on DTLZ1 problems with 3, 4, 6, 8, and 10 objectives. Due to the increase of influential points brought by the multimodal landscape of DTLZ1 during the evolutionary process, the negative effects of the influential points on the original Kriging model become significant. The result demonstrates the superiority of the influential point-insensitive model that can effectively reduce the negative effects of the influential points in MaOPs with the multimodal landscape.

TABLE I: Statistical results for IGD values obtained by KTA2(O) and KTA2 for DTLZ1 after 300 function evaluations. The best results are highlighted. The symbols '$+$', '$\approx$', '$-$' indicate that KTA2 is statistically significantly superior to, inferior to, and almost equivalent to KTA(O), respectively (the significance level is 0.05).

M	KTA2(O)	KTA2
3	5.48E+01(2.17E+01)$\approx$	**4.64E+01(1.84E+01)**
4	4.31E+01(1.55E+01)$+$	**3.51E+01(1.41E+01)**
6	1.79E+01(6.79E+01)$\approx$	**1.69E+01(6.75E+00)**
8	5.39E+00(3.41E+00)$\approx$	**4.26E+00(2.13E+00)**
10	3.92E-01(1.47E-01)$+$	**2.75E-01(5.45E-02)**

D. *Effects of Adaptive Sampling strategy*

To further discuss the role of the adaptive sampling strategy in KTA2, we compare KTA2 with the other six variants which are described as follows.

- **KTA2(C)** only uses the convergence sampling strategy.
- **KTA2(D)** only uses the diversity sampling strategy.
- **KTA2(U)** only uses the uncertainty sampling strategy.
- **KTA2(D+U)** removes the convergence sampling strategy from the adaptive sampling strategy. It employs PD value to identify the diversity-demand state and uncertainty-demand state.
- **KTA2(C+U)** removes the diversity sampling strategy from the adaptive sampling strategy. It employs the convergence indicator to identify the convergence-demand and uncertainty-demand state.
- **KTA2(C+D)** removes the uncertainty sampling strategy from the adaptive sampling strategy. It employs the convergence indicator to identify the convergence-demand and diversity-demand state.

The experiments are conducted on DTLZ1, DTLZ2, DTLZ4, and WFG5 with 3 objectives. The true PF of DTLZ2 is nonconvex and easy to converge, but needs more attention in the diversity maintenance. DTLZ4 is very sensitive to the initial training data because it is hard to initialize a well-distributed solution set. WFG5 is a deceptive problem. The statistical results are summarized in Table II. Seven algorithms have been ranked through the mean IGD values. The average rank on four problems for each algorithm is also listed in Table II.

As we expected, KTA2(D), KTA2(U), and KTA2(D+U) without the convergence sampling strategy rank the last three ranks on DTLZ1. KTA2(C), KTA2(U), and KTA2(C+U) without the diversity sampling strategy rank the last three ranks on DTLZ2. Interestingly, KTA2(C), KTA2(D) without the uncertainty sampling strategy have worse performance on WFG5 while KTA2(U) obtains the third rank. It can be concluded that the three sampling strategies are separately effective on three different types of problems, where the uncertainty sampling strategy might have advantages in dealing with the deceptive

TABLE II: Statistical results for IGD values obtained by KTA2 and other six variants for three objectives test problems after 300 function evaluations. The best results are highlighted.

Problems	DTLZ1	DTLZ2	DTLZ4	WFG5	Average Rank
KTA2(C)	5.02E+01(2.54E+01)	1.93E-01(1.62E-02)	3.87E-01(1.47E-01)	6.40E-01(5.50E-02)	6.00
KTA2(D)	6.38E+01(1.69E+01)	6.22E-02(2.69E-03)	3.30E-01(1.35E-01)	3.21E-01(1.28E-01)	3.75
KTA2(U)	7.83E+01(1.86E+01)	9.34E-02(8.28E-03)	4.35E-01(1.34E-01)	3.02E-01(6.13E-02)	5.75
KTA2(D+U)	7.19E+01(1.61E+01)	6.27E-02(2.64E-03)	3.39E-01(1.28E-01)	2.96E-01(7.78E-02)	3.75
KTA2(C+U)	4.78E+01(1.60E+01)	9.09E-02(8.68E-03)	3.67E-01(1.48E-01)	3.25E-01(9.40E-02)	4.75
KTA2(C+D)	4.71E+01(1.36E+01)	6.11E-02(2.84E-03)	3.41E-01(1.03E-01)	3.11E-01(9.61E-02)	3.00
KTA2	**4.64E+01(1.84E+01)**	**5.99E-02(3.01E-03)**	**3.24E-01(1.17E-01)**	**2.74E-01(7.86E-02)**	**1.00**

problems. However, unlike our expectations, the results on DTLZ4 seem to be similar to those on DTLZ2. KTA2(C), KTA2(U), and KTA2(C+U) rank the last three ranks. The characteristics of DTLZ4 are different from those of DTLZ2, the role of diversity sampling strategy should be different.

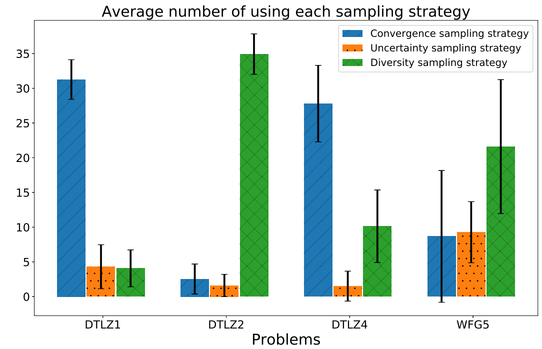


Fig. 9: The average number of using each sampling strategy on DTLZ1, DTLZ2, DTLZ4, and WFG5 problems in 30 independent runs.

In order to further explore the demand characteristics of different problems, we collect the number of using each sampling strategy on different problems in 30 independent runs. The average outcome is summarized in Fig. 9. We can conclude that the convergence sampling strategy is the most frequently used in the DTLZ1. In the DTLZ2, the diversity sampling strategy is the most used. Although the most frequently used strategy in WFG5 is the diversity sampling strategy, the uncertainty sampling strategy is used significantly more times than that on other problems. These results verify the above conclusions that three sampling strategies are separately effective on three different types of problems. In the DTLZ4, the convergence sampling strategy is the most frequently used which is different from the Table II. However, we can observe in Fig. 9 that the usage number of the diversity sampling strategy on DTLZ4 is also relatively large. We speculate that it is because of the characteristics of DTLZ4, the performance will be improved as long as the solution for diversity improvement is sampled. Thus, the cooperation of both convergence and diversity sampling strategies is beneficial for optimizing DTLZ4.

Clearly, some variants perform well in one case but perform worse in other cases while KTA2 achieves the best performance over four different types of problems, which verify the effectiveness of the adaptive sampling strategy. The adaptive sampling strategy can automatically select the required sampling strategy during the optimization process, which makes KTA2 obtain satisfactory performance on all test problems.

E. Comparison over Other Algorithms

In order to discuss the behavior of the proposed algorithm, we compare KTA2 with ParEGO, MOEA/D-EGO, KRVEA, CSEA, and HSMEA on 80 MOPs (DTLZ and WFG) with different numbers of objectives (3, 4, 6, 8, and 10). Note that for presenting the dominance and quality of the obtained solutions of all compared SAEAs, we use IGD$^+$ as the performance indicator in following experiments. Furthermore, the final non-dominated solutions obtained by six algorithms on 3 objectives DTLZ1-7 and WFG1-9 in the run associated with the median IGD$^+$ values are plotted in Figs. S1-16 in the supplementary file. The results under PD metric are presented in Table S-I of the supplementary file.

1) Results on DTLZ Problems: The results of IGD$^+$ values achieved by the six algorithms under comparison over 30 independent runs on DTLZ1 to DTLZ7 are summarized in Table III, where the best results are highlighted. Among them, we perform the WRS test for the comparison of KTA2 with the other five compared algorithms. It can be observed that KTA2 and HSMEA obtain the best performance with 10 best results on the 35 test instances, followed by KRVEA with 9 best results, CSEA with 6 best results. MOEA/D-EGO and ParEGO have not achieved the best results among the DTLZ test problems.

Both DTLZ1 and DTLZ3 have multimodal landscapes that are hard to converge. This might be the reason that KRVEA has a poor performance on DTLZ1 and DTLZ3. KTA2 is the best-performing algorithm on DTLZ1 and DTLZ3, followed by CSEA and ParEGO. The convergence-demand state frequently occurs during the optimization of multimodal MOPs, which causes that the convergence sampling strategy is frequently used. Furthermore, both CSEA and ParEGO take convergence into account, where CSEA selects potentially better converged solution and ParEGO selects the best prediction solution for the scalarization function. For DTLZ2, both KTA2 and KRVEA achieve a satisfactory performance, followed by HSMEA. The diversity sampling strategy is frequently used during the optimization of DTLZ2. Moreover, KRVEA and HSMEA adopt two sets of reference vectors which maintain a well-distribution of solutions set. MOEA/D-EGO has the worst performance on DTLZ1-3. This might be attributed to the sampling strategy in MOEA/D-EGO which tries to improve both convergence and diversity.

TABLE III: Statistic IGD^{+} results of the five compared algorithms and KTA2 on 35 DTLZ test instances by Wilcoxon signed rank test (the significance level is 0.05). The best results for each instance are highlighted. The symbols ’$+$’, ’$\approx$’, ’$-$’ indicate that KTA2 is statistically significantly superior to, inferior to, and almost equivalent to the particular compared algorithm, respectively.

Problems	M	HSMEA	CSEA	KRVEA	MOEA/D-EGO	ParEGO	KTA2
DTLZ1	3	6.12E+01(6.35E+00)+	5.82E+01(1.59E+01)+	8.86E+01(1.97E+01)+	8.80E+01(1.94E+01)+	6.50E+01(5.28E+00)+	**4.64E+01(1.84E+01)**
	4	4.58E+01(7.15E+00)+	3.95E+01(1.26E+01)≈	6.17E+01(1.25E+01)+	6.57E+01(1.56E+01)+	5.60E+01(5.70E+00)+	**3.51E+01(1.41E+01)**
	6	3.02E+01(5.30E+00)+	**1.49E+01(6.22E+00)**≈	3.23E+01(7.06E+00)+	3.49E+01(8.29E+00)+	3.31E+01(1.30E+01)+	1.69E+01(6.75E+00)
	8	1.08E+01(4.25E+00)+	4.50E+00(1.98E+00)≈	9.04E+00(3.84E+00)+	1.12E+01(4.83E+00)+	8.71E+00(4.34E+00)+	**4.26E+00(2.13E+00)**
	10	3.99E-01(1.74E-01)+	2.13E-01(9.53E-02)≈	3.11E-01(1.14E-01)+	3.51E-01(1.38E-01)+	2.85E-01(1.11E-01)+	**2.03E-01(6.33E-02)**
DTLZ2	3	4.50E-02(4.79E-03)+	1.66E-01(4.15E-02)+	9.58E-02(2.76E-02)+	2.25E-01(2.46E-02)+	3.19E-01(6.33E-02)+	**3.58E-02(2.91E-03)**
	4	1.24E-01(1.69E-02)+	2.48E-01(3.90E-02)+	1.31E-01(1.70E-02)+	2.78E-01(2.79E-02)+	3.86E-01(3.75E-02)+	**8.46E-02(6.35E-03)**
	6	2.12E-01(1.31E-02)−	3.36E-01(2.59E-02)+	**1.99E-01(1.63E-02)**−	3.29E-01(2.20E-02)+	4.40E-01(3.67E-02)+	2.25E-01(3.05E-02)
	8	2.31E-01(1.11E-02)−	3.88E-01(1.88E-02)+	**2.23E-01(1.21E-02)**−	3.58E-01(2.02E-02)+	3.96E-01(2.58E-02)+	3.02E-01(1.99E-02)
	10	**2.30E-01(7.54E-03)**−	4.10E-01(2.09E-02)+	2.36E-01(1.43E-02)−	2.88E-01(1.06E-02)−	4.03E-01(2.78E-02)+	3.08E-01(1.75E-02)
DTLZ3	3	1.77E+02(9.33E+00)+	1.49E+02(3.37E+01)≈	2.34E+02(4.92E+01)+	1.96E+02(2.67E+01)+	1.76E+02(8.26E+00)+	**1.43E+02(4.96E+01)**
	4	1.43E+02(1.29E+01)+	1.19E+02(3.40E+01)≈	1.83E+02(4.81E+01)+	1.65E+02(2.98E+01)+	1.50E+02(1.01E+01)+	**1.17E+02(4.24E+01)**
	6	9.17E+01(1.82E+01)+	5.69E+01(1.70E+01)≈	9.65E+01(2.53E+01)+	1.03E+02(1.92E+01)+	8.96E+01(2.88E+01)+	**5.63E+01(2.42E+01)**
	8	3.22E+01(1.28E+01)+	**1.40E+01(8.38E+00)**≈	3.07E+01(1.18E+01)+	3.60E+01(1.19E+01)+	2.83E+01(1.01E+01)+	1.59E+01(8.22E+00)
	10	1.02E+00(4.16E-01)+	7.51E-01(3.70E-01)≈	9.49E-01(5.05E-01)+	9.51E-01(3.47E-01)+	7.95E-01(2.38E-01)≈	**7.41E-01(2.16E-01)**
DTLZ4	3	2.32E-01(8.94E-02)≈	**1.83E-01(5.51E-02)**−	2.70E-01(8.66E-02)≈	4.50E-01(6.69E-02)+	2.13E-01(6.41E-02)≈	2.22E-01(7.88E-02)
	4	3.31E-01(8.44E-02)≈	**2.03E-01(3.39E-02)**−	3.03E-01(6.29E-02)≈	4.92E-01(3.94E-02)+	4.27E-01(9.50E-02)+	2.98E-01(8.20E-02)
	6	3.44E-01(5.21E-02)≈	**2.32E-01(2.71E-02)**−	2.81E-01(5.73E-02)−	4.33E-01(4.82E-02)+	2.84E-01(5.60E-02)−	3.48E-01(6.21E-02)
	8	3.01E-01(2.42E-02)−	**2.77E-01(3.40E-02)**−	2.86E-01(3.66E-02)−	3.48E-01(2.80E-02)≈	3.62E-01(3.86E-02)≈	3.45E-01(5.08E-02)
	10	**2.55E-01(9.72E-03)**−	2.95E-01(2.62E-02)≈	2.65E-01(3.08E-02)−	2.70E-01(1.16E-02)−	3.43E-01(3.52E-02)+	2.96E-01(3.30E-02)
DTLZ5	3	**8.68E-03(1.41E-03)**≈	9.68E-02(2.71E-02)+	8.73E-02(2.68E-02)+	1.76E-01(2.76E-02)+	2.26E-01(5.75E-02)+	8.69E-03(2.12E-03)
	4	**1.39E-02(2.00E-03)**−	1.03E-01(2.58E-02)+	4.81E-02(1.51E-02)+	1.51E-01(2.27E-02)+	2.49E-01(3.59E-02)+	3.18E-02(7.44E-03)
	6	2.25E-02(3.32E-03)−	6.75E-02(1.64E-02)−	**2.21E-02(7.47E-03)**−	1.03E-01(1.35E-02)+	1.30E-01(2.94E-02)+	8.54E-02(1.90E-02)
	8	1.93E-02(2.40E-03)−	3.97E-02(8.97E-03)−	**1.39E-02(3.82E-03)**−	4.86E-02(6.24E-03)−	5.49E-02(8.27E-03)≈	5.41E-02(9.33E-03)
	10	7.76E-03(1.19E-03)−	9.38E-03(1.37E-03)−	**6.32E-03(1.00E-03)**−	1.11E-02(9.58E-04)−	1.21E-02(1.10E-03)≈	1.26E-02(1.83E-03)
DTLZ6	3	**3.48E-01(4.10E-01)**−	5.67E+00(5.35E-01)+	3.13E+00(3.57E-01)+	1.75E+00(8.91E-01)≈	9.57E-01(2.96E-01)−	1.68E+00(4.74E-01)
	4	**2.49E-01(2.59E-01)**−	5.01E+00(4.42E-01)+	2.34E+00(3.40E-01)+	1.46E+00(7.55E-01)−	6.65E-01(3.09E-01)−	1.88E+00(5.03E-01)
	6	**1.73E-01(1.19E-01)**−	3.12E+00(4.97E-01)+	1.30E+00(2.56E-01)−	9.65E-01(6.04E-01)−	5.34E-01(1.65E-01)−	1.47E+00(4.95E-01)
	8	**6.15E-02(3.03E-02)**−	1.43E+00(5.61E-01)+	5.47E-01(2.36E-01)−	3.00E-01(3.23E-01)−	3.23E-01(8.25E-02)−	7.53E-01(4.25E-01)
	10	4.34E-02(1.61E-02)−	4.82E-02(5.78E-02)−	**2.64E-02(8.31E-03)**−	6.41E-02(1.83E-02)≈	7.73E-02(1.59E-02)≈	8.80E-02(5.18E-02)
DTLZ7	3	**5.72E-02(2.01E-02)**−	1.67E+00(7.82E-01)+	9.32E-02(4.10E-02)≈	1.41E-01(5.49E-02)≈	8.90E-02(2.25E-02)−	1.47E-01(1.42E-01)
	4	**2.23E-01(5.57E-02)**≈	3.75E+00(1.65E+00)+	2.53E-01(1.14E-01)≈	3.97E-01(4.43E-02)+	2.49E-01(4.06E-02)≈	2.59E-01(1.92E-01)
	6	5.06E-01(6.09E-02)+	8.72E+00(2.15E+00)+	**4.18E-01(7.93E-02)**≈	6.95E-01(6.06E-02)+	6.53E-01(4.40E-02)+	4.64E-01(1.82E-01)
	8	7.26E-01(5.71E-02)≈	6.94E+00(2.22E+00)+	**5.94E-01(8.36E-02)**−	8.33E-01(5.24E-02)+	9.40E-01(5.91E-02)+	6.75E-01(1.53E-01)
	10	9.51E-01(3.84E-02)≈	1.74E+00(4.35E-01)+	**8.43E-01(4.34E-02)**−	1.01E+00(4.28E-02)+	1.15E+00(5.89E-02)+	9.39E-01(1.28E-01)
+/≈/−		13/7/15	17/10/8	16/5/14	24/4/7	22/7/6	

As shown in Table III, KRVEA and CSEA are the top two algorithms for all DTLZ4 problems, followed by HSMEA and KTA2. As mentioned in Section IV-D, an initial training dataset with poor diversity will cause the surrogate model inaccurate on some objectives. Since most training data are converged on certain objectives, some points with good diversity have little effect on surrogate models. The surrogate models employed in the above SAEAs have good diversity. KRVEA builds Kriging models by means of selecting training data according to the reference vectors and HSMEA uses multiple surrogate models, both of which enhance the diversity of surrogate models. CSEA distinguishes good solutions from poor solutions instead of approximating objective value, which reduces the negative effect of poorly distributed training data. KTA2 predicts the objective value by the influential point-insensitive models. As mention in Section III-A, the influential point-insensitive models can degenerate the influence of the convergence training data and maintain the solutions with good diversity.

Since DTLZ5-7 have irregular PFs, maintaining diversity in solving these problems is challenging. Among them, the true PFs of DTLZ5 and DTLZ6 are degenerated curves and the PF of DTLZ7 is discontinuous. It can be seen from Table III, for DTLZ5 and DTLZ7, HSMEA and KRVEA have achieved the best results, followed by KTA2. Both HSMEA and KRVEA employ reference vectors that can achieve a set of diverse and well-converged solutions on instances with irregular PFs. Different from DTLZ5, DTLZ6 is a nonseparable-reduced problem. HSMEA is the best-performing algorithm on DTLZ6. This might because that HSMEA uses the nadir point-based reference vectors, which provide the flexibility of dealing with different PF shapes. Although KTA2 is not the best algorithm on DTLZ5-7, it still shows a competitive performance compared with the other five algorithms.

2) Results on WFG Problems: To further analyze the behavior of our algorithm, we choose the WFG problems as test problems. Among them [39], WFG1 has the most transformation functions t among the WFG problems, which makes it hard for MOEAs to achieve a good diversity. WFG2 is a disconnected problem which has an irregular PF. WFG3

TABLE IV: Statistic IGD$^+$ results of the five compared algorithms and KTA2 on 45 WFG test instances. The best results for each instance are highlighted. The symbols ’+’, ’≈’, ’−’ indicate that KTA2 is statistically significantly superior to, inferior to, and almost equivalent to the particular compared algorithm, respectively (the significance level is 0.05)

Problems	M	HSMEA	CSEA	KRVEA	MOEA/D-EGO	ParEGO	KTA2
WFG1	3	**1.66E+00(5.02E-02)**−	1.72E+00(9.42E-02)≈	1.71E+00(7.19E-02)≈	2.07E+00(8.20E-02)+	1.66E+00(3.73E-02)−	1.78E+00(1.57E-01)
	4	**1.84E+00(6.16E-02)**−	1.97E+00(1.00E-01)−	1.99E+00(1.08E-01)−	2.26E+00(7.69E-02)+	1.88E+00(5.17E-02)−	2.06E+00(1.48E-01)
	6	**2.27E+00(5.48E-02)**≈	2.39E+00(5.50E-02)+	2.38E+00(9.64E-02)+	2.61E+00(6.74E-02)+	2.27E+00(3.49E-02)≈	2.28E+00(1.56E-01)
	8	2.53E+00(4.80E-02)≈	2.63E+00(1.14E-01)+	2.60E+00(1.01E-01)≈	2.85E+00(6.05E-02)+	2.59E+00(5.08E-02)≈	**2.48E+00(3.04E-01)**
	10	2.81E+00(4.72E-02)+	2.83E+00(2.33E-01)+	2.86E+00(7.78E-02)+	3.08E+00(3.58E-02)+	2.83E+00(7.29E-02)+	**2.67E+00(2.06E-01)**
WFG2	3	4.06E-01(1.29E-01)+	4.91E-01(7.53E-02)+	3.23E-01(4.28E-02)+	5.75E-01(4.74E-02)+	8.40E-01(6.40E-02)+	**2.14E-01(4.27E-02)**
	4	5.49E-01(2.65E-01)+	6.83E-01(9.59E-02)+	3.64E-01(5.25E-02)≈	7.44E-01(7.94E-02)+	1.14E+00(9.85E-02)+	**3.51E-01(5.05E-02)**
	6	7.19E-01(3.95E-01)+	9.43E-01(2.79E-01)+	**2.93E-01(5.24E-02)**−	9.86E-01(1.43E-01)+	1.45E+00(4.51E-01)+	4.43E-01(8.39E-02)
	8	9.52E-01(5.17E-01)+	1.53E+00(8.35E-01)+	**2.76E-01(5.08E-02)**−	1.34E+00(2.61E-01)+	1.82E+00(6.89E-01)+	5.82E-01(1.16E-01)
	10	1.69E+00(5.43E-01)+	2.89E+00(1.05E+00)+	**3.91E-01(1.74E-01)**−	1.64E+00(4.71E-01)+	3.00E+00(7.06E-01)+	7.61E-01(1.54E-01)
WFG3	3	**1.93E-01(3.22E-02)**−	5.53E-01(3.82E-02)+	5.12E-01(7.95E-02)+	6.11E-01(4.33E-02)+	6.63E-01(3.63E-02)+	3.06E-01(8.09E-02)
	4	**1.67E-01(3.13E-02)**−	6.07E-01(5.61E-02)+	6.19E-01(7.25E-02)+	7.57E-01(3.75E-02)+	7.79E-01(4.75E-02)+	4.94E-01(6.74E-02)
	6	**3.09E-01(5.74E-02)**−	5.66E-01(9.04E-02)−	4.39E-01(8.54E-02)−	8.42E-01(6.43E-02)+	5.22E-01(1.21E-01)−	6.38E-01(7.14E-02)
	8	**2.27E-01(3.80E-02)**−	4.04E-01(1.04E-01)−	4.05E-01(7.18E-02)−	7.14E-01(7.10E-02)≈	4.18E-01(5.55E-02)−	6.80E-01(6.01E-02)
	10	**3.07E-01(5.93E-02)**−	5.08E-01(8.48E-02)−	5.80E-01(1.30E-01)−	8.14E-01(7.89E-02)+	4.98E-01(1.13E-01)−	7.61E-01(9.69E-02)
WFG4	3	3.93E-01(3.09E-02)+	3.79E-01(3.60E-02)+	3.88E-01(2.65E-02)+	4.10E-01(2.02E-02)+	5.60E-01(2.60E-02)+	**3.47E-01(2.48E-02)**
	4	5.75E-01(7.96E-02)≈	7.75E-01(1.69E-01)+	5.75E-01(4.49E-02)≈	6.26E-01(3.71E-02)+	8.52E-01(5.30E-02)+	**5.64E-01(5.29E-02)**
	6	1.05E+00(1.43E-01)+	2.53E+00(3.42E-01)+	1.11E+00(2.27E-01)≈	1.09E+00(8.90E-02)+	1.40E+00(1.34E-01)+	**1.02E+00(2.28E-01)**
	8	1.58E+00(4.79E-01)−	4.92E+00(6.93E-01)+	2.04E+00(4.09E-01)≈	1.48E+00(2.04E-01)−	**1.45E+00(2.05E-01)**−	1.96E+00(3.48E-01)
	10	2.97E+00(9.01E-01)−	7.94E+00(6.18E-01)+	3.89E+00(8.04E-01)≈	**2.10E+00(4.93E-01)**−	2.38E+00(3.89E-01)−	3.60E+00(4.65E-01)
WFG5	3	2.12E-01(2.03E-02)+	3.48E-01(3.84E-02)+	3.34E-01(4.37E-02)+	3.86E-01(3.19E-02)+	2.13E-01(1.45E-02)+	**2.00E-01(6.05E-02)**
	4	5.01E-01(1.26E-01)≈	5.85E-01(5.57E-02)+	5.26E-01(4.71E-02)+	6.64E-01(4.21E-02)+	4.67E-01(6.66E-02)≈	**4.59E-01(4.93E-02)**
	6	1.02E+00(1.18E-01)−	1.71E+00(2.50E-01)+	**8.96E-01(1.12E-01)**−	1.58E+00(1.55E-01)+	1.95E+00(2.23E-01)+	1.20E+00(1.48E-01)
	8	2.23E+00(4.61E-01)≈	3.66E+00(3.49E-01)+	**1.57E+00(3.12E-01)**−	3.34E+00(3.48E-01)+	3.94E+00(3.62E-01)+	2.19E+00(4.46E-01)
	10	3.89E+00(6.84E-01)+	6.06E+00(5.29E-01)+	**2.71E+00(9.98E-01)**−	5.41E+00(6.01E-01)+	6.40E+00(3.43E-01)+	3.50E+00(5.27E-01)
WFG6	3	4.61E-01(3.87E-02)−	5.92E-01(4.87E-02)+	6.63E-01(5.79E-02)+	5.81E-01(3.52E-02)+	**4.38E-01(1.59E-02)**−	5.42E-01(7.68E-02)
	4	7.23E-01(7.25E-02)−	8.31E-01(6.36E-02)+	9.06E-01(6.09E-02)+	7.85E-01(3.40E-02)≈	**7.12E-01(3.46E-02)**−	7.95E-01(6.33E-02)
	6	1.19E+00(5.60E-02)−	1.89E+00(3.85E-01)+	**1.17E+00(1.30E-01)**−	1.31E+00(3.26E-02)≈	1.45E+00(1.58E-01)+	1.29E+00(1.14E-01)
	8	1.37E+00(1.42E-01)−	3.76E+00(6.18E-01)+	**1.20E+00(5.37E-02)**−	2.23E+00(4.19E-01)≈	3.10E+00(6.30E-01)+	2.04E+00(1.71E-01)
	10	1.88E+00(4.16E-01)−	6.22E+00(7.16E-01)+	**1.34E+00(6.44E-02)**−	2.65E+00(3.79E-01)−	5.79E+00(7.80E-01)+	3.88E+00(3.68E-01)
WFG7	3	4.98E-01(3.25E-02)≈	**4.88E-01(3.89E-02)**≈	5.48E-01(2.51E-02)+	5.27E-01(1.94E-02)+	6.33E-01(4.16E-02)+	5.02E-01(2.72E-02)
	4	7.35E-01(1.70E-01)≈	7.72E-01(1.35E-01)≈	**7.19E-01(4.06E-02)**≈	8.02E-01(5.73E-02)+	9.32E-01(6.07E-02)+	7.35E-01(2.86E-02)
	6	1.34E+00(3.27E-01)+	2.14E+00(4.10E-01)+	1.24E+00(1.15E-01)+	1.85E+00(3.46E-01)+	1.99E+00(2.85E-01)+	**1.17E+00(1.18E-01)**
	8	2.53E+00(7.02E-01)+	4.63E+00(5.16E-01)+	2.35E+00(2.05E-01)+	3.70E+00(5.54E-01)+	3.20E+00(7.48E-01)+	**2.18E+00(1.99E-01)**
	10	4.57E+00(8.22E-01)+	7.07E+00(5.66E-01)+	**3.48E+00(4.23E-01)**−	5.77E+00(9.15E-01)+	5.84E+00(1.49E+00)+	3.78E+00(3.93E-01)
WFG8	3	5.36E-01(4.77E-02)+	7.27E-01(4.43E-02)+	6.78E-01(4.62E-02)+	7.36E-01(2.65E-02)+	8.46E-01(3.33E-02)+	**4.85E-01(4.07E-02)**
	4	1.08E+00(1.00E-01)+	1.12E+00(7.68E-02)+	1.01E+00(6.11E-02)+	1.07E+00(4.06E-02)+	1.16E+00(3.06E-02)+	**9.03E-01(5.44E-02)**
	6	1.63E+00(4.05E-02)−	2.61E+00(2.19E-01)+	**1.57E+00(9.72E-02)**−	1.78E+00(7.41E-02)≈	2.26E+00(2.49E-01)+	1.74E+00(9.24E-02)
	8	2.53E+00(5.01E-01)−	4.74E+00(5.52E-01)+	**1.80E+00(2.02E-01)**−	2.95E+00(5.07E-01)≈	3.49E+00(9.86E-01)+	2.92E+00(2.86E-01)
	10	4.04E+00(1.20E+00)−	7.24E+00(5.03E-01)+	**1.84E+00(2.57E-01)**−	3.51E+00(4.96E-01)−	6.71E+00(1.02E+00)+	4.73E+00(4.66E-01)
WFG9	3	5.42E-01(8.05E-02)≈	5.78E-01(1.03E-01)+	5.58E-01(8.23E-02)+	6.55E-01(6.12E-02)+	7.19E-01(6.43E-02)+	**5.02E-01(7.49E-02)**
	4	9.09E-01(1.71E-01)+	1.07E+00(1.09E-01)+	8.08E-01(8.43E-02)≈	9.86E-01(8.61E-02)+	1.18E+00(1.30E-01)+	**7.92E-01(1.25E-01)**
	6	1.99E+00(3.32E-01)+	2.41E+00(3.06E-01)+	1.39E+00(2.33E-01)≈	2.15E+00(3.87E-01)+	2.71E+00(2.00E-01)+	**1.39E+00(2.81E-01)**
	8	3.72E+00(5.87E-01)+	4.23E+00(6.45E-01)+	2.74E+00(6.29E-01)+	3.84E+00(6.10E-01)+	4.86E+00(4.39E-01)+	**2.28E+00(4.41E-01)**
	10	5.65E+00(7.46E-01)+	6.80E+00(6.51E-01)+	4.47E+00(9.67E-01)≈	6.11E+00(6.92E-01)+	7.24E+00(5.44E-01)+	**4.14E+00(6.20E-01)**
+/≈/−		19/8/18	38/3/4	17/11/17	35/6/4	33/3/9	

replaces the disconnected PF of WFG2 with a continuous one. WFG4 has a multi-modal landscape, which is hard for MOEAs to converge. WFG5 is a deceptive problem. Both WFG6 and WFG9 are nonseparable-reduced problems. WFG7 is a separable and uni-modal problem. WFG8 is a nonseparable problem that is hard for MOEAs to maintain diversity. Thus, comparing with the DTLZ problems, it is more difficult to maintain diverse solutions on the WFG problems.

The statistical results of compared algorithms on WFG problems are summarized in Table IV, where the best results are highlighted. We also perform the WRS test for the comparison of KTA2 with the other five compared algorithms. It can be observed that KTA2 is the overall best-performing algorithm with 18 best results among the compared SAEAs, followed by KRVEA (14) and HSMEA (8), ParEGO (2), CSEA(1), and MOEA/D-EGO (1).

In the comparison between KTA2 and KRVEA, the WRS test results show that KTA2 achieves 17 ”better”, 17 ”worse”, and 11 ”similar” results on WFG problems. It can be observed that KTA2 outperforms KRVEA on WFG4 and WFG9 problems but performs worse on WFG2. KTA2 shows similar performance to KRVEA on other six WFG problems. Interestingly, the results on WFG2 and WFG3 show that only KRVEA and KTA2 degrade their performance when the disconnected problem connected. Since the PF of WFG3 is a degenerated curve, KRVEA and KTA2 frequently sample too many uncertain solutions which result in poor performance. Comparing KTA2 with HSMEA, we can see that KTA2 achieves 19 ”better”, 18 ”worse”, and 8 ”similar” results on WFG problems. HSMEA is the best-performing algorithm on WFG1 and WFG3. We speculate that it is because of the employing of multiple surrogate models and a set of nadir point-based reference vectors. The surrogate models provide more approximation methods to approximates WFG1. Nadir point-based reference vectors are more effective than that based on ideal point in maintaining the diversity on WFG3. For the WRS test between MOEA/D-EGO and KTA2, KTA2 achieves 35 ”better”, 4 ”worse”, and 6 ”similar” results while results on ParEGO are 33 ”better”, 9 ”worse”, and 3 ”similar” results. In the comparison between KTA2 and CSEA, hardly any result of CSEA is better than KTA2, where KTA2 achieves 28 ”better”, 4 ”worse”, and 3 ”similar” results.

3) Discussions: From the experimental results on 80 test instances, it can be observed that none of the SAEAs is able to successfully handle all types of problems. Generally speaking, ParEGO, MOEA/D-EGO, KRVEA, and HSMEA are all aggregation-based MOEAs, which are sensitive to reference vectors. In ParEGO, limited evaluations are concentrated on optimizing a single aggregation problem, which brings a strong convergence ability to the algorithm such that ParEGO achieves the best performance on DTLZ1 and DTLZ3 among the above aggregation-based MOEAs. However, randomly weighted aggregations may result in poor diversity. As the steady-state nature performs a poor diversity in ParEGO, MOEA/D-EGO selects several solutions in one iteration for re-evaluation which enhances the diversity. Also, MOEA/D-EGO employs the EI as the selection metric which considers the uncertainty information of surrogate models, it makes MOEA/D-EGO has a better performance than ParEGO. In HSMEA, it benefits from the nadir and the ideal point-based reference vectors, making it highly competitive in dealing with irregular PF problems and has good convergence and diversity during the optimization process. As we can see in Table III and IV, HSMEA achieves good performance on DTLZ5-7 and WFG1 and WFG3. HSMEA also employs multiple models for a robust approximation. Therefore, it has advantages to handle problems with many transform functions (e.g. WFG1 which has the most transform functions among the WFG problems). KRVEA uses RVEA as the optimizer which provides both good convergence and diversity. However, the key point of KRVEA is that it uses the numbers of active and inactive reference vectors to measure the diversity and then determines which solutions should be selected. Thus, it achieves the second best performance with 23 best results among 80 test instances. CSEA achieves the satisfactory performance on DTLZ1, DTLZ3, DTLZ4, WFG3. The results show that CSEA has advantages to handle problems that are hard for MOEAs to converge. However, more attention should be paid to maintaining diversity in CSEA. KTA2 is the best-performing algorithm with 28 best results compared to the other five SAEAs. The optimization state changes differently on different problems. For example, the convergence demand state frequently occurs during the optimization of DTLZ1, while the diversity demand state often appears for DTLZ2. The adaptive sampling strategy can effectively adjust the sampling strategy according to the optimization state. Therefore, KTA2 is more robust than other compared SAEAs. Moreover, the results show that diversity and convergence could also be enhanced without using reference vectors. However, it also can be observed that KTA2 performs unsatisfactorily on DTLZ4-7 and WFG1-3 compared to other test problems. Different from HSMEA and KRVEA, the improvement of convergence is the priority of KTA2. Once the optimization state is assessed as the convergence-demand state, the diversity sampling strategy should not work. As mentioned in Section III-B, the convergence of one archive is measured by calculating the distances from solutions to the estimated point. The convergence sampling strategy is frequently used, even if there is only a slight improvement of convergence. When the optimization state assessment works, the irregularity of the current PF will lead to the inability to distinguish the diversity-demand state. That is the reason why the performance of KTA2 is not good on DTLZ4-7 and WFG1-3.

In general, the main benefits of KTA2 can be summarized as follows:

1) KTA2 employs an influential point-insensitive model as the surrogate model. The influential point-insensitive model is designed for reducing the negative effect of influential points. Experimental result on an multimodal problem (DTLZ1) has demonstrated its superiority over the original Kriging model.
2) Different from the existing SAEAs for MOPs, which employ one population, KTA2 uses Two_Arch2 with two populations targeted on convergence and diversity as the optimizer. The dual population structure provides the demand information for the current optimization state. Based on the information provided by Two_Arch2, we have designed an adaptive sampling strategy that can respond to different demands of optimization state.
3) Compared with the existing SAEAs that use mixed indicator sampling strategies, KTA2, which uses tailored pure indicator-based sampling strategies, is more competitive than existing SAEAs.

V. Conclusions

In this work, we propose a Kriging-assisted Two_Arch2 algorithm for expensive multi/many-objective optimization, where each objective function is approximated by a new influential point-insensitive model and an adaptive sampling strategy is designed. The influential point-insensitive model can effectively reduce the negative effects of the influential points, also the proposed model is proved to have better performance than the original Kriging model on multimodal problems. We independently consider convergence, diversity, and model accuracy in selecting new solutions for re-evaluation. The adaptive sampling strategy reasonably divides the optimization states into three states according to their demands. Furthermore, we employ a tailored sampling strategy for each demand state to select candidate solutions to be re-evaluated. In addition, we adopt Two_Arch2 as the optimizer to maintain two archives to separately promote convergence and diversity without any manual settings in advance.

The overall performance on the widely used test problems DTLZ and WFG show that KTA2 achieves much better performance than the compared SAEAs given the same number of expensive function evaluations. This work proves that the adaptive sampling strategy is promising for solving expensive MOPs. However, KTA2 lacks strategies to handle irregular PFs. Therefore, it deserves further efforts to develop the optimization process assessment and sampling strategies by taking account of irregular PFs. Further, designing more suitable indicators to replace the convergence and PD metrics is a potential way to improve the performance of KTA2. In addition, as the dimension limitation of Kriging model, our method can be only applied to low-dimensional problems. A promising way to address this issue is to use other regression models instead of Kriging, such as RBFN or dropout neural networks [57], where the uncertainty information of the predicted solution can be replaced by calculating the distance between the solution and the training data or the disagreement in the base learners of an ensemble [58].

VI. Acknowledgments

We would like to thank Mr. Hongbin Xu for discussions and sharing insights regarding the influential point-insensitive model and Dr. Ahsanul Habib for sharing HSMEA codes.

References

[1] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *International Conference on Evolutionary Multi-criterion Optimization*. Springer, 2005, pp. 14–32.

[2] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 2, pp. 97–111, 2017.

[3] D. Guo, T. Chai, J. Ding, and Y. Jin, "Small data driven evolutionary multi-objective optimization of fused magnesium furnaces," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016.

[4] S. K. Mishra, G. Panda, and S. Meher, "Multi-objective particle swarm optimization approach to portfolio optimization," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. IEEE, 2009, pp. 1612–1615.

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

[6] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, pp. 1–35, 2015.

[7] H. Wang, S. He, and X. Yao, "Nadir point estimation for many-objective optimization problems based on emphasized critical regions," *Soft Computing*, vol. 21, no. 9, pp. 2283–2295, 2017.

[8] 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.

[9] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 74–88, 2018.

[10] T. Chugh, K. Sindhya, K. Miettinen, Y. Jin, and P. Makkonen, "Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system," in *IEEE Congress on Evolutionary Computation 2017*, 2017.

[11] 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.

[12] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: an overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2018.

[13] C. He, Y. Tian, H. Wang, and Y. Jin, "A repository of real-world datasets for data-driven evolutionary multiobjective optimization," *Complex & Intelligent Systems*, pp. 1–9, 2019.

[14] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou, "Metamodel-Assisted evolution strategies," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2002, pp. 361–370.

[15] Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang, "Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 350–364, 2020.

[16] Y. Jin, M. Olhofer, and B. Sendhoff, "On evolutionary optimization with approximate fitness functions," in *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, 2000, pp. 786–793.

[17] D. G. Krige, "A statistical approach to some mine valuation and allied problems on the Witwatersrand: By DG krige," Ph.D. dissertation, University of the Witwatersrand, 1951.

[18] G. E. Box and N. R. Draper, *Empirical model-building and response surfaces*. Wiley New York, 1987, vol. 424.

[19] J. M. Zurada, *Introduction to artificial neural systems*. West St. Paul, 1992, vol. 8.

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

[21] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Royal Signals and Radar Establishment Malvern (United Kingdom), Tech. Rep., 1988.

[22] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770–784, 2007.

[23] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *International conference on evolutionary multi-criterion optimization*. Springer, 2007, pp. 742–756.

[24] 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.

[25] 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.

[26] Y. Sun, G. G. Yen, and Z. Yi, "Igd indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2019.

[27] 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.

[28] 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.

[29] 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.

[30] 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, 2009.

[31] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 1000–1014, 2019.

[32] X. Yu, X. Yao, Y. Wang, L. Zhu, and D. Filev, "Domination-based ordinal regression for expensive multi-objective optimization," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 2058–2065.

[33] N. Azzouz, S. Bechikh, and L. Ben Said, "Steady state IBEA assisted by MLP neural networks for expensive multi-objective optimization problems," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 581–588.

[34] 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.

[35] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, no. 117-129, p. 2, 1978.

[36] J. P. Stevens, "Outliers and influential data points in regression analysis." *Psychological Bulletin*, vol. 95, no. 2, p. 334, 1984.

[37] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John Wiley & Sons, 2005, vol. 589.

[38] K. Bringmann, "Bringing order to special cases of klee¡¯s measure problem," in *International Symposium on Mathematical Foundations of Computer Science*. Springer, 2013, pp. 207–218.

[39] 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, 2014.

[40] P. L. Yu, "Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives," *Journal of Optimization Theory and Applications*, vol. 14, no. 3, pp. 319–377, 1974.

[41] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimisation algorithm: The two-archive algorithm," in *2006 International Conference on Computational Intelligence and Security*, vol. 1. IEEE, 2006, pp. 286–291.

[42] R. Morgan and M. Gallagher, "Sampling techniques and distance metrics in high dimensional continuous landscape analysis: Limitations and improvements," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 456–461, 2013.

[43] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[44] N. R. Draper and H. Smith, *Applied regression analysis*. John Wiley & Sons, 1998, vol. 326.

[45] M. Meloun, M. Hill, J. Militkỳ, J. Vrbíková, S. Stanicka, and J. Škrha, "New methodology of influential point detection in regression model building for the prediction of metabolic clearance rate of glucose," *Clinical Chemistry and Laboratory Medicine (CCLM)*, vol. 42, no. 3, pp. 311–322, 2004.

[46] 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.

[47] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," in *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2013, pp. 282–290.

[48] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1510–1522, 2016.

[49] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [Educational Forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[50] 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.

[51] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, 2016.

[52] S. Huband, P. Hingston, L. Barone, 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.

[53] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[54] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 110–125.

[55] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, "Comparison of hypervolume, igd and igd+ from the viewpoint of optimal distributions of solutions," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2019, pp. 332–345.

[56] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, *DACE: a Matlab kriging toolbox*. Citeseer, 2002, vol. 2.

[57] D. Guo, X. Wang, K. Gao, Y. Jin, J. Ding, and T. Chai, "Evolutionary optimization of high-dimensional multi- and many-objective expensive problems assisted by a dropout neural network," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020 (Accepted).

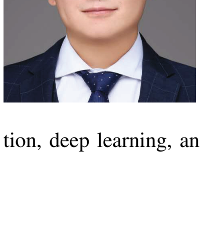
[58] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1012–1025, 2018.

Zhenshou Song received the B.Eng. from Nanchang University, Nanchang, China, in 2019.

Currently, he is a Ph.D. degree candidate student in the School of Artificial Intelligence, Xidian University, Xi'an, China. His current research interests include swarm intelligence, multiobjective optimization and surrogate-assisted evolutionary optimization.

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 professor with School of Artificial Intelligence, Xidian University, Xi'an, China. Dr. Wang is an Associate Editor of IEEE Computational Intelligence Magazine, Memetic Computing, and Complex & Intelligent Systems. Her research interests include nature-inspired computation, multiobjective optimization, multiple criteria decision making, surrogate-assisted evolutionary optimization, and real-world problems.



Cheng He (M¡¯2016) received the B.Eng. degree from the Wuhan University of Science and Technology, Wuhan, China, in 2012, and the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2018.

He is currently a Research Assistant Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His current research interests include model-based evolutionary algorithms, multiobjective optimization, large-scale optimization, deep learning, and their applications.

Yaochu Jin (M’98-SM’02-F’16) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996 respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Germany, in 2001.

He is currently a Distinguished Chair Professor in Computational Intelligence, Department of Computer Science, University of Surrey, Guildford, UK. He is also a Finland Distinguished Professor funded by the Finnish Agency for Innovation (Tekes) and a Changjiang Distinguished Visiting Professor appointed by the Ministry of Education, China. His research interests include computational intelligence, computational neuroscience, and computational systems biology. He is particularly interested in nature-inspired, real-world driven problem-solving. He has (co)authored over 400 peer-reviewed journal and conference papers and been granted eight patents on evolutionary optimization.

Dr Jin is the Editor-in-Chief of the IEEE Transactions on Cognitive and Developmental Systems and Complex & Intelligent Systems. He is an IEEE Distinguished Lecturer (2017-2019) and was the Vice President for Technical Activities of the IEEE Computational Intelligence Society (2014-2015). He is a Fellow of IEEE. He was named a ”Highly Cited Researcher” by the Web of Science Group in 2019 and 2020.