Adaptive Switching between Search and Model Refinement via Explainable Machine Learning in Surrogate-assisted Evolutionary Algorithms

1st Kei Nishihara
Organization for the Promotion of Education
Yokohama National University
Yokohama, Japan
nishihara-kei-pv@ynu.ac.jp

2nd Takahiro Sato, 4th Shinya Watanabe
Graduate School of Engineering
Muroran Institute of Technology
Muroran, Japan
{t-sato, sin}@muroran-it.ac.jp

3rd Kazuhiro Izui
Graduate School of Engineering
Kyoto University
Kyoto, Japan
izui.kazuhiro.8c@kyoto-u.ac.jp

***Abstract*—Surrogate-assisted evolutionary algorithms (SAEAs) alternate expensive evaluations using surrogate models of objective functions, and thus have become a key approach for solving expensive optimization problems. Several efforts have been made to enhance the prediction accuracy of surrogate models. However, this does not necessarily lead to performance improvement of SAEAs, as solutions contributing to the update of the best objective value and model refinement are technically different from each other in nature. Thus, balancing the two aspects is crucial for accelerating the performance of SAEAs. Nevertheless, the resource allocation method to sample solutions for optimization or for model refinement remains unclear. Accordingly, this work provides a novel structure-aware adaptive switching methodology between evolutionary search and model refinement modes, grounded in explainable machine learning. Specifically, SHapley Additive exPlanations (SHAP) are employed to quantify the structural instability of the surrogate model. When the surrogate model is structurally unstable or inaccurate, inspired by active learning, the proposed SAEA framework generates solutions by anisotropically perturbing decision variables according to dimension-wise SHAP instability to refine the model. Otherwise, it switches to the evolutionary search mode to discover better solutions. The effectiveness of the proposed framework over state-of-the-art SAEAs is demonstrated through comparison on benchmark and engineering design problems.**

***Index Terms*—Adaptive mode switching, explainable machine learning, surrogate-assisted evolutionary algorithm, radial basis function network, differential evolution.**

I. Introduction

Real-world optimization problems involve computationally/financially expensive objective function evaluations (FEs) due to the use of time-consuming numerical simulations or costly physical experiments, e.g., motor design [1], portfolio optimization [2], and pharmaceutical product development [3]. For such expensive optimization problems (EOPs), surrogate-assisted evolutionary algorithms (SAEAs) have been widely studied as an effective approach [4]. SAEAs construct inexpensive surrogate models of expensive objective functions using machine learning models to predict the quality of candidate solutions, thereby reducing the number of expensive FEs required to find high-quality solutions.

This work was supported by JSPS KAKENHI Grant No. 26K21330.

Many efforts have been made to improve the prediction accuracy of surrogate models, as it has a significant impact on the quality of solution selection in SAEAs. For example, GPEME [5] and SAEO [6] introduce dimensionality reduction techniques while many SAEAs focus on local regions [7]–[9], to reduce the difficulty of approximation. While most SAEAs adopt approximation models for surrogate models, classification or relationship models have also been studied in IDRCEA [10] and DESSA [11], respectively, to mitigate serious prediction errors. Some literature considers adaptation mechanisms to adjust surrogate models and/or its settings during the optimization process automatically [4], because appropriate model types/settings vary depending on problems and search situations. For instance, machine learning model types [12]–[14], model settings such as kernel functions or hyperparameters [15], and sampling criterion of training data [16] are selected as the adaptation targets. An ensemble of surrogate models is another powerful method to improve the prediction accuracy and robustness [17]. GS-SOMA [17] and DSP-SAEA [12] prepare ensemble members with different model types, while HESNFO [15] and PS-SAEA [18] employ various kernel functions of radial basis function network (RBFN) [19] for the ensemble. The simultaneous use of different degrees of smoothness of the approximated fitness landscape contributes to improving the stability of the prediction.

The other direction to enhance the optimization performance is the development of search strategies. SAPO [8], EBADE [20], and HSADE-CS [21] combine different mutation strategies of differential evolution (DE) [22] to generate a variety of offspring solutions. A quasi-Newton method was employed in [23] instead of evolutionary algorithms (EAs) to accelerate convergence speed in local regions. SAHO [24] and SAF-TD [25] switch between two EAs when the best solution is not updated for a certain period. The idea of tabu search was included in SAEAs in [26]. AutoSAEA [13] and AutoSAPSO [14] adaptively select infill criterion and EA during search, respectively, in addition to the model type.

Both the model refinement to improve the prediction accuracy and the acceleration of evolutionary search are important and should be balanced for excellent performance of SAEAs. However, not only in global regions but also in local ones, solutions contributing to the model refinement and update of the best tentative objective value are technically different from each other in nature. To be specific, solutions having the best predicted objective value do not always improve the prediction accuracy of surrogate models because it ignores the approximation error, i.e., uncertainty [27]. This sampling is effective for the optimization performance only when the surrogate model is already accurate, but not conducive to improving the prediction accuracy of the model, as the evaluated solutions are concentrated near the current best one [28]. Conversely, of course, simply sampling solutions in regions with high uncertainty does not directly help SAEAs find the superior solutions. Traditional attempts to find solutions profitable both for the search and model refinement include infill criteria using uncertainty and the combination use of exploitation and exploration. Nevertheless, finding such solutions at once is still highly challenging because the effects dissipate for each one due to the conflicting characteristics of the two targets [29].

Therefore, the adaptive switching between the two modes of pursuance of the global optimum and model refinement can be a natural approach. However, the resource allocation method, i.e., when and how many solutions SAEAs should sample for each mode, is not discussed enough in the literature and thus remains unclear. Accordingly, this work establishes a novel adaptive switching methodology between evolutionary search and model refinement modes as well as an effective switching criterion. Specifically, we utilize SHapley Additive exPlanations (SHAP) [30] for adaptive switching and model refinement. SHAP is a game theory-based method to explain the prediction results. It fairly distributes the prediction difference among features and assigns each feature a value representing its contribution to the predictions. The proposed algorithm, named explainable adaptive switching-based SAEA (XAS-SAEA), quantifies the structural instability of the constructed surrogate models. It considers the surrogate model as unstable when the Shapley values of data points near a certain point in the training dataset fluctuate significantly. This is because the metric reflects whether the model consistently understands which decision variables are important. Then, when the surrogate model is structurally unstable or inaccurate, XAS-SAEA proceeds to the model refinement mode. Inspired by active learning, XAS-SAEA generates solutions to correct this instability by utilizing the previously obtained dimension-wise Shapley values. Otherwise, it switches to the evolutionary search mode to discover better solutions.

The main contributions of this work are as follows;

- To deal with the two contradictory targets of obtaining excellent solutions and refining the surrogate models, this work establishes a structure-aware adaptive switching criterion between the optimization and model refinement modes for the first time. The proposed criterion is grounded in SHAP, an explainable machine learning, and we quantify the model instability with it. This provides how reliable the surrogate model can be and whether it should be modified.
- The utilization of the Shapley values above also furnishes an active learning-like guideline in the model refinement. Specifically, sampling solutions to modify the dimension-wise instability leads to efficient model improvement.
- Based on them, this work proposes XAS-SAEA, which adaptively switches between the optimization and model refinement modes depending on the surrogate model instability and the prediction accuracy. Our framework supplies a useful direction to handle the conflict of optimization and model refinement for future research.

The remainder of this work is organized as follows. Section II introduces the fundamental techniques, and Section III describes the details of XAS-SAEA. In Section IV, we demonstrate the experimental results of performance comparisons between XAS-SAEA and state-of-the-art SAEAs on the CEC 2020 bound-constrained single-objective optimization benchmark suite [31]. Section V is devoted to discussions on the effectiveness of XAS-SAEA. Lastly, we conclude our work and demonstrate future directions in Section VI.

II. Preliminary

This section explains DE, RBFN, and SHAP as the components of XAS-SAEA. This work focuses on single-objective continuous minimization problems with D-dimensional solutions $\boldsymbol{x} = [x_1, \dots, x_D]^\mathsf{T} \in \mathbb{R}^D$ and an objective function $f : \mathbb{R}^D \to \mathbb{R}$. The lower and upper bounds for the d-th decision variable are denoted as x_d^l and x_d^u, respectively.

A. DE: Differential Evolution

DE is an evolutionary algorithm, and its procedure consists of initialization, mutation, crossover, and solution selection. The N solutions of the initial population $\mathcal{P} = \{\boldsymbol{x}_i\}_{i=1}^N$ are randomly sampled from a uniform distribution with a range $x_{i,d} \in [x_d^l, x_d^u]$. The mutation procedure produces a mutant solution $\boldsymbol{v}_i = [v_{i,1}, \cdots, v_{i,D}]^\mathsf{T}$ for each $\boldsymbol{x}_i$ by applying a mutation strategy. This work employs the *best/1* mutation strategy $\boldsymbol{v}_i = \boldsymbol{x}_{best} + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$, where $\boldsymbol{x}_{best} \in \mathcal{P}$ is the best solution, $\boldsymbol{x}_{r_1}$ and $\boldsymbol{x}_{r_2} \in \mathcal{P}$ ($r_1 \neq r_2 \neq i$) are randomly selected mutually exclusive solutions, and $F \in [0, 1]$ is the scaling factor. Then, a trial vector $\boldsymbol{u}_i = [u_{i,1}, \cdots, u_{i,D}]^\mathsf{T}$ is generated for each $\boldsymbol{x}_i$ via a crossover strategy. For $u_{i,d}$, the *binomial* crossover strategy selects $v_{i,d}$ if $rand(0, 1) \leq CR$ or $j = j_{rand}$, where $rand(0, 1)$ is a real uniformly distributed random value in the range of $(0, 1)$, $CR \in [0, 1]$ is the crossover rate, and j_{rand} is an integer randomly chosen from 1 to D. Otherwise, $x_{i,d}$ is used for $u_{i,d}$. After the minimum and maximum values are bounded within $[x_d^l, x_d^u]$, the trial vectors are evaluated. As the solution selection, $\boldsymbol{x}_i$ is replaced with $\boldsymbol{u}_i$ if $f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i)$ for each i. DE goes back to the mutation and repeats the procedure until the termination criteria are met.

B. RBFN: Radial Basis Function Network

An RBFN is a three-layer feed-forward neural network and provides an approximation of $f(\boldsymbol{x})$ for $\boldsymbol{x}$ as $\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\boldsymbol{x} - \boldsymbol{x}_i\|) + p(\boldsymbol{x})$. Here, a training dataset (size n), a vector of function values, and the kernel function are $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^{n}$, $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)]^{\mathsf{T}}$, and $\phi(r)$, respectively. We use the *cubic* kernel function $\phi(r) = r^3$. Besides, $p(\boldsymbol{x}) = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + c_0$ ($\boldsymbol{c} \in \mathbb{R}^D, c_0 \in \mathbb{R}$) is a regularization term with a linear polynomial function, and $\lambda_i \in \mathbb{R}$ is the i-th element of the weight vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^{\mathsf{T}}$. The parameters $\boldsymbol{\lambda}$, $\boldsymbol{c}$, and c_0 are calculated by solving the following equation:

$$\begin{bmatrix} \Phi & P \\ P^{\mathsf{T}} & 0_{\mathrm{m}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{c}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix}, \tag{1}$$

where Φ is the $n \times n$ matrix consisting of $\phi_{ij} = \phi(|\boldsymbol{x}_i - \boldsymbol{x}_j|)$, $P \in \mathbb{R}^{n \times (D+1)}$ is a matrix whose i-th row is $\left[1,\ \boldsymbol{x}_i^{\mathsf{T}}\right]$, 0_{m} is a $(D+1) \times (D+1)$ matrix of zeros, $\boldsymbol{c}' = \left[\boldsymbol{c}^{\mathsf{T}},\ c_0\right]^{\mathsf{T}}$, and $\boldsymbol{0}$ is a $(D+1)$-sized column vector of zeros.

C. FastSHAP - a computationally efficient variant of SHAP

Given a predictive model $\hat{f}$, SHAP quantifies the contribution of each feature, i.e., decision variable, by fairly distributing the prediction difference between the model output and a baseline value among all features. This attribution satisfies desirable properties such as efficiency, symmetry, and consistency, making SHAP a theoretically grounded and widely adopted explanation framework. The SHAP explanation for $\boldsymbol{x}$ is defined as a Shapley vector $\boldsymbol{\Phi}(\boldsymbol{x}) = [\phi_1(x_1), \ldots, \phi_D(x_D)]^{\mathsf{T}}$, where $\phi_d(x_d)$ represents the Shapley value as the contribution of the d-th decision variable x_d, defined as:

$$\phi_d(x_d) = \sum_{S \subseteq \mathcal{I} \setminus \{d\}} \frac{|S|!(D - |S| - 1)!}{D!} (\hat{f}(S \cup \{d\}) - \hat{f}(S)), \tag{2}$$

where $\mathcal{I} = \{1, \ldots, D\}$ denotes the index set of dimesions and $\hat{f}(S)$ is the value function associated with subset S.

However, SHAP is time-consuming as it requires evaluating all possible feature subsets [32]. XAS-SAEA repeatedly uses SHAP to assess the current instability of the surrogate models. Thus, this work uses a variant named FastSHAP [32] instead of the original one. FastSHAP is a learning-based Shapley value approximation method that trains a standalone explainable model to directly predict Shapley values. The explainer is optimized by minimizing a loss function derived from the weighted least-squares formulation of Shapley values, without relying on precomputed ground-truth explanations. After training, FastSHAP produces Shapley value estimates in a single forward propagation, achieving substantial computational savings compared to sampling-based methods.

A little work incorporated SHAP or FastSHAP into EAs or SAEAs. SHAP is used in mutation and crossover procedures of genetic programming to quantify the superiority of function nodes in [33], while EXO-SAEA [34] employs FastSHAP for crossover to enhance search efficiency by guiding solutions towards directions in which performance improvements are expected in expensive multiobjective optimization problems. Unlike them, we utilize FastSHAP to evaluate whether the model is suitable for optimization, and if not, dimension-aware model refinement is performed using the FastSHAP results.

III. XAS-SAEA

A. Concept

To conduct both optimization and model refinement effectively, XAS-SAEA is designed to switch between the optimization mode ($Mode = \mathrm{O}$) and model refinement mode ($Mode = \mathrm{M}$) adaptively depending on the prediction accuracy and reliability of the current surrogate model. It considers the reliability to exclude cases where the model coincidentally achieves high accuracy. Figure 1 illustrates the proposed adaptive switching and the details of model refinement. As a metric of the reliability of surrogate models, we define the structural model instability using SHAP as follows. We examine whether surrogate models consistently understand which decision variables are important in a local region consisting of the best tentative solution $\boldsymbol{x}_c$ and the surrounding total s points. SHAP provides dimension-wise contributions for each point as the Shapley values $\phi_d(x_d)$. Since peaks and valleys of approximated fitness landscapes often flip within the same interval, and the magnitude of contributions is first important, we obtain absolute normalized values as Shapley vectors $\boldsymbol{\Phi}(\boldsymbol{x}) = [\phi_1(x_1), \ldots, \phi_D(x_D)]^{\mathsf{T}}$. Then, if $\phi_d(x_d)$ for each d and $\boldsymbol{x}$ widely varies against $\phi_d(x_{c,d})$, we consider that the model failed in obtaining the structure of important decision variables. Also, the model should be improved if the prediction accuracy for certain generations is low. In such cases, XAS-SAEA proceeds to $Mode$ M, wherein the average differences of the Shapley values between $\boldsymbol{x}_c$ and the other $\boldsymbol{x}$ are used as weights w_d for each d in the following model refinement sampling. This enables XAS-SAEA to correct the model instability around $\boldsymbol{x}_c$ because the broad sampling in unstable directions of decision variables contributes to obtaining their structures. On the other hand, if the model is accurate and stable, XAS-SAEA conducts the DE-based $Mode$ O.

B. Algorithm

Algorithm 1 describes the detailed procedures of XAS-SAEA. It prepares the initial population $\mathcal{P} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ consisting of N solutions produced by Latin Hypercube Sampling (LHS) and evaluates them using the expensive objective function. Then, an archive $\mathcal{A} = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^{N}$ to store all evaluated solutions is prepared, and the FE counter FE is set to N. At the beginning of the execution, we set $Mode$ to M because the surrogate model is not verified.

In the main loop, XAS-SAEA extracts the best and surrounding total N_{data} points in $\mathcal{A}$ to obtain the training dataset $\mathcal{D}$ and constructs an RBFN using it. Next, the prediction error err is calculated using points in $\mathcal{D}$ and their past predictions $\hat{f}_{\mathrm{past}}(\boldsymbol{x})$. Specifically, a standardized mean absolute error is employed, which can be expressed as:

$$err = \frac{\sum_{\boldsymbol{x} \in \mathcal{D}} |\hat{f}_{\mathrm{past}}(\boldsymbol{x}) - f(\boldsymbol{x})| / |\mathcal{D}|}{\mathrm{std}(f(\boldsymbol{x}_{1:|\mathcal{D}|})) + \varepsilon}, \tag{3}$$

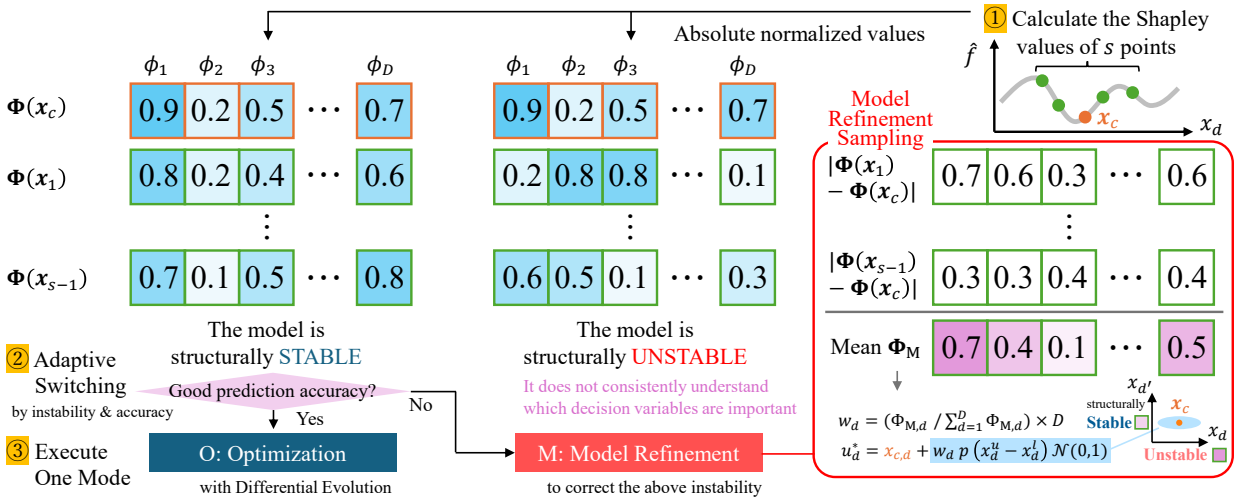

Fig. 1. A diagram of adaptive switching between two modes in XAS-SAEA. SHAP assesses the structural instability of surrogate models.

where $\mathrm{std}(f(\boldsymbol{x}_{1:|\mathcal{D}|}))$ calculates the standard deviation of objective function values for all points in $\mathcal{D}$, and $\varepsilon = 1\text{E-}10$ is a small value to avoid zero divisions. As explained in subsection III-A, the model instability ist is computed using absolute normalized Shapley values $\phi_d(x_d)$ as follows:

$$ist = \frac{\sum_{i=1}^{s-1}\sum_{d=1}^{D}|\phi_d(x_{i,d}) - \phi_d(x_{c,d})|}{s-1}, \tag{4}$$

where $\sum_{d=1}^{D}|\phi_d(x_{i,d}) - \phi_d(x_{c,d})|$ provides how much the Shapley values of $\boldsymbol{x}_i$ are different from those of $\boldsymbol{x}_c$ for all dimensions, and the mean for all $\{i\}_1^{s-1}$ is used as ist.

After ws generations have passed, XAS-SAEA conducts the adaptive switching of modes. Given boundaries τ_l, τ_u ($\tau \in [0, 100]$), τ_l, τ_u percentiles of err or ist in the past ws generations, i.e., $\theta_l^{err}, \theta_u^{err}$ or $\theta_l^{ist}, \theta_u^{ist}$, respectively, are prepared. Then, $Mode$ is set to M if $err > \theta_u^{err}$ or $ist > \theta_u^{ist}$, and O if $err < \theta_l^{err}$ and $ist < \theta_l^{ist}$. Here, $[\theta_l^{err}, \theta_u^{err}]$ and $[\theta_l^{ist}, \theta_u^{ist}]$ are margins also known as hystereses to avoid too frequent switching. Besides, the "and/or" conditions are carefully designed to choose $Mode$ O only when surrogate models are reliable enough to use.

In $Mode$ M, XAS-SAEA uses the mean of the differences of ϕ_d between $\boldsymbol{x}_c$ and the other points, i.e., $\boldsymbol{\Phi}_\mathrm{M} = \sum_{i=1}^{s-1}|\phi_d(x_{i,d}) - \phi_d(x_{c,d})|/(s-1)$, for the weight $\boldsymbol{w}$ in the model refinement sampling. Each dimension w_d of $\boldsymbol{w}$ is calculated as $w_d = (\Phi_{\mathrm{M},d}/\sum_{d=1}^{D}\Phi_{\mathrm{M},d}) \times D$, where D is multiplied for the scalability against D. Subsequently, XAS-SAEA generates a sample whose d-th dimension is $u_d^* = x_{c,d} + w_d\, p\, (x_d^u - x_d^l)\, \mathcal{N}(0,1)$ using the standard normal distribution $\mathcal{N}(0,1)$. Conversely, in $Mode$ O, the DE offspring is generated from the top N solutions from $\mathcal{A}$, and a solution $\boldsymbol{u}^*$ having the best predicted objective value $\hat{f}(\boldsymbol{u})$ is selected. For both modes, XAS-SAEA evaluates the selected $\boldsymbol{u}^*$ with the expensive objective function, then $\mathcal{A}$ and FE are updated.

IV. Experimental Results

We evaluate the effectiveness of XAS-SAEA through a performance comparison with state-of-the-art SAEAs.

Algorithm 1 XAS-SAEA

Require: the maximum number of function evaluations $FE_{\max}$, population size N, scaling factor for DE F, crossover rate for DE CR, data size for surrogate modeling N_{data}, data size for SHAP s, window size ws, boundaries for model assessment τ_l, τ_u ($\tau \in [0, 100]$), coefficient for model refinement sampling p

Initialize $\mathcal{P} = \{\boldsymbol{x}_i\}_{i=1}^N$ by LHS and Evaluate $\forall \boldsymbol{x} \in \mathcal{P}$
$\mathcal{A} = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^N$, $FE = N$
$Mode = \mathrm{M}$ // M: Model Refinement, O: Optimization
while $FE < FE_{\max}$ **do**
  /* Model Construction and Assessment */
  $\mathcal{D} \leftarrow$ Get the best and surrounding total N_{data} points in $\mathcal{A}$
  $\hat{f} \leftarrow$ Construct an RBFN with $\mathcal{D}$
  $err \leftarrow$ Calculate the prediction error of past predictions
  $ist \leftarrow$ Calculate the model instability using the Shapley values $\boldsymbol{\Phi}(x)$ for s center points in $\mathcal{D}$ for $\hat{f}$
  /* Adaptive Switching */
  if $FE > N + ws$ **then**
    $\theta_l^{err}, \theta_u^{err} \leftarrow \tau_l, \tau_u$ percentiles of err in the past ws gens.
    $\theta_l^{ist}, \theta_u^{ist} \leftarrow \tau_l, \tau_u$ percentiles of ist in the past ws gens.
    $Mode \leftarrow \begin{cases} \mathrm{M}, & \text{if } err > \theta_u^{err} \text{ or } ist > \theta_u^{ist}, \\ \mathrm{O}, & \text{if } err < \theta_l^{err} \text{ and } ist < \theta_l^{ist}, \end{cases}$
  if $Mode = \mathrm{M}$ **then**
    /* Model Refinement Mode */
    $\boldsymbol{u}^* \leftarrow$ model refinement sampling using p and dimension-wise variance of $\boldsymbol{\Phi}(x)$ of s points
  else
    /* Optimization Mode */
    $\mathcal{P} \leftarrow$ Get top N solutions from $\mathcal{A}$
    for $i = 1$ **to** N **do**
      $\boldsymbol{v}_i \leftarrow$ mutation with $\mathcal{P}$, F
      $\boldsymbol{u}_i \leftarrow$ crossover with $\boldsymbol{x}_i$, $\boldsymbol{v}_i$, CR
    $\boldsymbol{u}^* \leftarrow$ select one $\boldsymbol{u}$ having the best $\hat{f}(\boldsymbol{u})$
  $f(\boldsymbol{u}^*) \leftarrow$ Evaluate $\boldsymbol{u}^*$ with true f
  $\mathcal{A} = \mathcal{A} \cup \{(\boldsymbol{u}^*, f(\boldsymbol{u}^*))\}$, $FE = FE + 1$
Ensure: The best solution in $\mathcal{A}$

A. Experimental Design

This work employed the CEC 2020 bound-constrained single-objective optimization benchmark suite [31]. The prob-

lem dimension was set to $D \in \{50, 100\}$ to evaluate the scalability of the performance of XAS-SAEA to the increase of D. Thus, we can verify in 20 cases in total. The bound constraint was $[x_d^l, x_d^u] = [-100, 100]$ for every dimension. We conducted the following experiments with the Intel Ultra 7 265 (2.4 GHz) CPU, 32GB RAM, and the PlatEMO software [35].

We chose GL-SADE [7], IDRCEA [10], DSP-SAEA [12], AutoSAEA [13], and MHS-QPSO [9] for the compared SAEAs due to their use of RBFN and their novelty. Their hyperparameter settings followed the papers while those of XAS-SAEA were set to $N = 100$, $F = 0.5$, $CR = 0.9$, $N_{data} = 5D$, $s = 30$, $ws = 20$, $\tau_l = 50\%$, $\tau_u = 80\%$, $p = 0.01$.

Considering EOPs, we set the maximum number of FEs to $1,000$. The performance was reported with the mean objective function values of 25 independent trials. Additionally, we applied the Wilcoxon signed-rank test with a significance level of 0.05. The results are reported with "+," "−," or "∼," indicating the statistical superiority, inferiority, or competitiveness of the compared algorithm to XAS-SAEA, respectively.

B. Results

TABLE I summarizes the mean objective function values discovered at $1,000$ FEs, as well as the statistical test results and the average rank. The best and worst values are highlighted in bold green and italic pink, respectively. XAS-SAEA obtained the best results in four and eight problem instances for $D = 50$ and 100, respectively. This demonstrates the highly scalable and better performance of XAS-SAEA with the increase of D. The statistical test results also indicate the superiority of XAS-SAEA. Specifically, it statistically outperformed the compared algorithm on 11 cases at minimum and 20 cases at maximum while significantly underperformed on six cases at maximum out of 20. AutoSAEA was the relatively most competitive in terms of the test results. However, XAS-SAEA derived a stabler performance as it obtained no worst results while AutoSAEA became the worst in three problem instances. It is worth noting that the compared algorithms include various approaches to improve the prediction accuracy and search mechanisms as introduced in Section I, such as local surrogate models, classification models, and adaptation and/or ensemble of models and/or infill criteria. Even considering it, the effectiveness of our SHAP-based adaptive switching and model refinement sampling was observed, which can be confirmed by the best average rank of XAS-SAEA.

TABLE I
MEAN OBJECTIVE FUNCTION VALUES AT $1,000$ FITNESS EVALUATIONS FOR $D \in \{50, 100\}$ ON THE CEC2020 BENCHMARK SUITE.

ID	D	GL-SADE	IDRCEA	DSP-SAEA	AutoSAEA	MHS-QPSO	XAS-SAEA
F1	50	6.244e+09 −	*1.669e+11* −	**1.835e+07** +	1.983e+08 −	5.874e+07 +	6.657e+07
	100	1.078e+10 −	*4.799e+11* −	7.325e+09 −	8.447e+10 −	2.064e+10 ∼	**2.899e+09**
F2	50	1.502e+04 −	1.541e+04 −	1.475e+04 −	*1.544e+04* −	1.375e+04 −	**8.607e+03**
	100	3.272e+04 −	3.322e+04 −	3.273e+04 −	*3.348e+04* −	3.276e+04 −	**1.823e+04**
F3	50	5.010e+02 +	*3.875e+03* −	4.622e+02 +	**4.338e+02** +	6.950e+02 +	1.117e+03
	100	1.694e+03 +	*1.106e+04* −	**1.126e+03** +	2.455e+03 +	2.139e+03 +	3.486e+03
F4	50	*4.043e+07* −	2.409e+07 −	2.273e+04 −	2.141e+02 −	3.526e+03 −	**1.254e+02**
	100	1.700e+08 −	*2.420e+08* −	7.510e+06 −	7.456e+05 −	2.340e+06 ∼	**9.802e+04**
F5	50	8.236e+07 −	*2.197e+08* −	5.665e+07 −	**6.011e+06** +	1.418e+08 −	8.618e+06
	100	1.442e+08 −	*1.252e+09* −	2.789e+08 −	8.838e+07 ∼	3.587e+08 −	**7.462e+07**
F6	50	6.356e+03 −	*6.744e+03* −	3.282e+03 ∼	**2.072e+03** +	4.686e+03 −	3.225e+03
	100	1.213e+04 −	*2.772e+04* −	1.015e+04 −	7.727e+03 ∼	1.187e+04 −	**7.177e+03**
F7	50	*6.266e+07* −	4.277e+07 −	1.671e+07 −	**4.383e+06** +	3.382e+07 −	7.838e+06
	100	1.480e+08 −	*5.407e+08* −	1.582e+08 −	3.286e+07 ∼	2.166e+08 −	**3.119e+07**
F8	50	1.509e+04 −	*1.581e+04* −	1.103e+04 ∼	1.529e+04 −	1.351e+04 −	**8.867e+03**
	100	3.414e+04 −	3.436e+04 −	3.337e+04 −	*3.463e+04* −	3.371e+04 −	**2.025e+04**
F9	50	8.132e+02 +	*1.527e+03* −	9.118e+02 ∼	**6.588e+02** +	9.667e+02 ∼	9.060e+02
	100	**1.949e+03** ∼	*4.773e+03* −	2.070e+03 ∼	3.418e+03 −	2.781e+03 −	2.065e+03
F10	50	1.157e+03 −	*2.696e+04* −	7.402e+02 −	7.372e+02 −	1.059e+03 −	**6.897e+02**
	100	2.804e+03 −	*8.043e+04* −	5.168e+03 −	7.061e+03 −	5.176e+03 −	**1.590e+03**
+/−/∼		3/16/1	0/20/0	3/13/4	6/11/3	3/14/3	-
Ave. Rank		3.700	*5.750*	2.850	3.150	3.750	**1.800**

TABLE II
ABLATION STUDY ON ADAPTATION OF OPTIMIZATION AND MODEL REFINEMENT MODES ($Mode$ O AND M, RESPECTIVELY).

ID	D	*Mode* O	*Mode* M	XAS-SAEA	ID	D	*Mode* O	*Mode* M	XAS-SAEA
F1	50	*1.855e+09* −	1.771e+09 −	**6.657e+07**	F6	50	**2.406e+03** +	*5.027e+03* −	3.225e+03
	100	5.504e+09 −	*3.784e+10* −	**2.899e+09**		100	**6.088e+03** +	*1.345e+04* −	7.177e+03
F2	50	9.073e+03 ∼	*1.227e+04* −	**8.607e+03**	F7	50	*1.462e+07* −	1.071e+07 −	**7.838e+06**
	100	2.007e+04 −	*2.781e+04* −	**1.823e+04**		100	4.066e+07 −	*4.223e+07* −	**3.119e+07**
F3	50	**5.599e+02** +	*2.417e+03* −	1.117e+03	F8	50	**8.799e+03** ∼	*1.223e+04* −	8.867e+03
	100	**1.987e+03** +	*6.637e+03* −	3.486e+03		100	2.175e+04 −	*2.893e+04* −	**2.025e+04**
F4	50	*3.516e+03* −	1.018e+03 −	**1.254e+02**	F9	50	**7.541e+02** +	*1.805e+03* −	9.060e+02
	100	*6.715e+05* −	2.624e+05 −	**9.802e+04**		100	**1.739e+03** +	*4.633e+03* −	2.065e+03
F5	50	*2.275e+07* −	1.133e+07 −	**8.618e+06**	F10	50	*9.609e+02* −	9.370e+02 −	**6.897e+02**
	100	7.534e+07 ∼	*9.886e+07* −	**7.462e+07**		100	2.146e+03 −	*4.269e+03* −	**1.590e+03**
+/−/∼							6/11/3	0/20/0	-
Ave. Rank							1.950	*2.700*	**1.350**

V. DISCUSSION

A. Impact of Adaptive Switching of Execution Mode

We first conduct an ablation study on the adaptation of execution modes. We prepared two variants of XAS-SAEA; one conducts $Mode$ O only while the other conducts $Mode$ M alone. Note that the latter calculates the Shapley values for the model refinement sampling. The experimental design was identical to Section IV-A except for the compared algorithms.

TABLE II lists the mean performance, statistical test results, and the average rank at $1,000$ FEs. The table should be read in the same way as before. First, $Mode$ M alone unsurprisingly led to the inferior performance as its aim was not optimization. It statistically underperformed XAS-SAEA in all the problem instances. On the other hand, $Mode$ O significantly outperformed XAS-SAEA on six problem instances but underperformed on 11. This demonstrates the need of combinational use of $Mode$ O and M. Moreover, without $Mode$ M, $Mode$ O lacked the stability of performance; it sometimes derived worse performance than $Mode$ M. Thus, the adequacy of our adaptive switching of two execution modes was validated.

B. Relationship between Optimization Performance, Prediction Accuracy, Model Instability, and Execution Modes

Next, we observe the relationship between the optimization performance, prediction accuracy and instability of surrogate models, and execution modes. Figure 2 illustrates an example obtained in the seventh trial on F4 ($D = 50$), where green, orange, and blue lines indicate the performance, prediction error, and instability, respectively, and white/pink patches show the $Mode$ O/M, respectively. XAS-SAEA statistically outperformed the compared algorithms in TABLEs I and II with the difference of several orders of magnitude on F4.

During the beginning $ws = 20$ generations, the prediction accuracy and model instability were improved by $Mode$ M. Then, the model instability took various values over approxi-

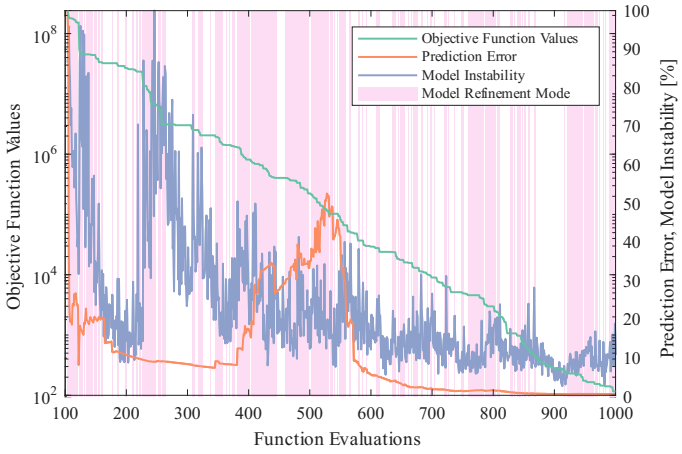

Fig. 2. An example of the relationship between performance (green), prediction error (orange), model instability (blue), and *Modes* O/M (white/pink, respectively) obtained in the seventh trial on F4 ($D = 50$).

mately 250 generations, as the current best solution changed frequently in a wide region due to *Mode* O. Subsequently, XAS-SAEA mainly conducted *Mode* M from almost 400 to 550 FEs. Although XAS-SAEA suffered from improving the prediction accuracy due to the flat landscape of F4 near the global optimum [31], it reduced the model instability in every *Mode* M period and repeated it whenever the model became unstable. Conversely, some compared algorithms took a longer time to or could not refine surrogate models against this difficulty, resulting in poor performance. Thanks to our model refinement sampling, XAS-SAEA succeeded in rapidly decreasing the prediction error after 550 FEs. This led to sustainable performance improvement. Moreover, XAS-SAEA executed *Mode* M before 800 FEs and after 900 FEs, and each of them contributed to further acceleration of performance improvement after the model refinement.

C. Sensitivity Analyses of Hyperparameters

We conducted three sensitivity analyses of hyperparameters, i.e., the data size for SHAP s, window size ws, and coefficient for model refinement sampling p. We tested $s \in \{10, 30(\text{default}), 50\}$, $ws \in \{10, 20(\text{default}), 30\}$, and $p \in \{0.005, 0.01(\text{default}), 0.05\}$. TABLE III contains the summary of the analyses, that is, the statistical tests and average rank. The signs "+," "−," and "∼," mean that XAS-SAEA with the changed hyperparameters statistically outperformed, underperformed, and was competitive with XAS-SAEA with the default values, respectively.

For s, $s = 10$ lacked the stability of the performance due to the small sample size for SHAP, resulting in the worst average rank. To enhance the reliability of the SHAP assessment, we use the larger values. When s was set to 50, the performance became better than $s = 30$. However, since the results were statistically competitive to each other, we used $s = 30$ to reduce the execution time. Specifically, the average time over 25 trials and 10 problems with $D = 50$ was 347.9, 947.7, and 1729 seconds for $s = 10$, 30, and 50, respectively.

TABLE III
SENSITIVITY ANALYSES OF HYPERPARAMETERS.

	s		(default↓)	ws		(default↓)	p		(default↓)
	10	50	30	10	30	20	0.005	0.05	0.01
+/ − / ∼	2/2/16	0/0/20	-	0/0/20	1/1/18	-	3/4/13	6/11/3	-
Ave. Rank	*2.200*	**1.650**	2.150	**1.900**	*2.100*	2.000	1.900	*2.350*	**1.750**

The window size ws was also insensitive to the performance. The performance of the three variants was statistically competitive, and the average ranks were around two for all.

However, p was sensitive to the performance of XAS-SAEA. To be specific, the statistical results of the default setting were slightly and significantly better than $p = 0.005$ and 0.05, respectively. The default setting also obtained the best average rank. These results demonstrate the importance of carefully selecting the range for model refinement sampling. With too wide a range, sampled solutions fell out of the focused regions. Conversely, the solutions within a too narrow range had less effect on modifying the model instability.

D. Real-world Application: Topology Optimization of Shield

Finally, we applied XAS-SAEA and the previously used competitors to the shield problem as a real-world application. As shown in Fig. 3 (a), this is a problem of designing a shield that minimizes the magnetic flux density B reaching the target area from the coil by placing a limited amount of iron within the design area. We used a topology optimization approach with kernel level set functions ((b), (c)). More information can be found in [1]. The number of kernel functions, i.e., the problem dimension, was set to 96. The objective funciton is $f = B + \max(S - S_T, 0)$, where S and $S_T = 0.0036$ m^2 are the area of the shield and a threshold, respectively.

Figure 4 exhibits the convergence curves of the average performance of 25 trials with standard deviations. XAS-SAEA demonstrated excellent convergence speed throughout. This was achieved by our SHAP-based mode switching and quick adaptation of surrogate models to the shield problem.

VI. CONCLUSION

This work presented an SAEA with adaptive switching between optimization and model refinement modes, namely XAS-SAEA. Specifically, SHAP evaluates the instability of surrogate models as a consistent ability to understand key decision variables in local regions. Utilizing this model instability along with the prediction accuracy, we proposed a switching criterion. Moreover, the Shapley values also contribute to modifying the model instability for successive optimization. Our explainable machine learning-based mechanisms helped XAS-SAEA to outperform the state-of-the-art algorithms. We also conducted an ablation study, behavior observation, sensitivity analyses of hyperparameters, and a real-world application.

Future work includes the utilization of positive/negative signs of the Shapley values to enhance the efficiency and development of more carefully crafted optimization and model refinement modes. We are also motivated to provide a theoretical justification for the contributions of the proposed instability metric against prediction error or improvement probability.

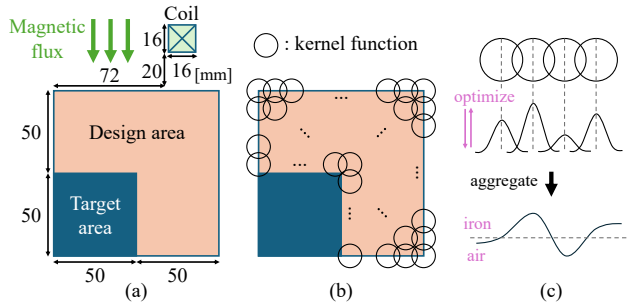

Fig. 3. Magnetic shield design problem. (a) Problem definition. (b) Kernel disposion. (c) Topology optimization using kernel level set functions.

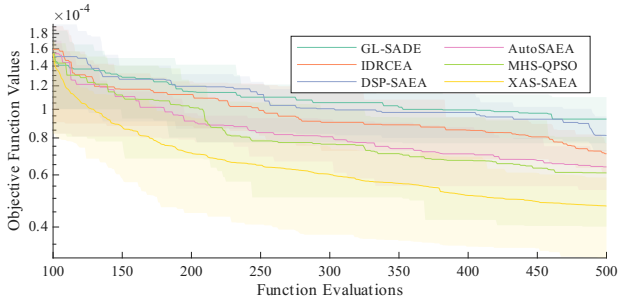

Fig. 4. Convergence curves of the performance in the shield design.

REFERENCES

[1] T. Sato, K. Watanabe, and H. Igarashi, "An evolutional topology optimization method based on kernel level set function," *IEEE Access*, vol. 13, pp. 47 181–47 200, 2025.

[2] I. Suemitsu and K. Izui, "Surrogate-assisted scenario-generation method for simulation-based stochastic programming problems," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 13 161–13 174, 2025.

[3] S. Sano, T. Kadowaki, K. Tsuda, and S. Kimura, "Application of Bayesian optimization for pharmaceutical product development," *J. Pharm. Innov.*, vol. 15, no. 3, pp. 333–343, Sep. 2020.

[4] C. He, Y. Zhang, D. Gong, and X. Ji, "A review of surrogate-assisted evolutionary algorithms for expensive optimization problems," *Expert Syst. Appl.*, vol. 217, p. 119495, May 2023.

[5] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian Process Surrogate Model Assisted Evolutionary Algorithm for Medium Scale Expensive Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[6] M. Cui, L. Li, M. Zhou, and A. Abusorrah, "Surrogate-Assisted Autoencoder-Embedded Evolutionary Optimization Algorithm to Solve High-Dimensional Expensive Problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 676–689, Aug. 2022.

[7] W. Wang, H.-L. Liu, and K. C. Tan, "A Surrogate-Assisted Differential Evolution Algorithm for High-Dimensional Expensive Optimization Problems," *IEEE Trans Cybern*, vol. 53, no. 4, pp. 2685–2697, 2023.

[8] K. Nishihara and M. Nakata, "A Surrogate-Assisted Partial Optimization for Expensive Constrained Optimization Problems," in *Int. Conf. Parallel Probl. Solving Nat. (PPSN)*, 2024, pp. 391–407.

[9] C. Li, Q. Zhang, V. Palade, H. Lu, and J. Sun, "Multi-region hierarchical surrogate-assisted quantum-behaved particle swarm optimization for expensive optimization problems," *Expert Syst. Appl.*, vol. 261, p. 125496, Feb. 2025.

[10] G. Li, L. Xie, and M. Gong, "Evolutionary algorithm with individual-distribution search strategy and regression-classification surrogates for expensive optimization," *Inf. Sci.*, vol. 634, pp. 423–442, Jul. 2023.

[11] X. Lu, K. Tang, B. Sendhoff, and X. Yao, "A new self-adaptation scheme for differential evolution," *Neurocomput.*, vol. 146, pp. 2–16, Dec. 2014.

[12] Y. Liu, J. Liu, and S. Tan, "Decision space partition based surrogate-assisted evolutionary algorithm for expensive optimization," *Expert Syst. Appl.*, vol. 214, p. 119075, Mar. 2023.

[13] L. Xie, G. Li, Z. Wang, L. Cui, and M. Gong, "Surrogate-assisted evolutionary algorithm with model and infill criterion auto-configuration," *IEEE Trans. Evol. Comput.*, vol. 28, no. 4, pp. 1114–1126, Aug. 2024.

[14] R. Dai, J. Jie, Z. Wang, H. Zheng, and W. Wang, "Automated surrogate-assisted particle swarm optimizer with an adaptive parental guidance strategy for expensive engineering optimization problems," *J. Comput. Des. Eng.*, vol. 12, no. 3, pp. 145–183, Feb. 2025.

[15] M. Yu, J. Liang, Z. Wu, and Z. Yang, "A twofold infill criterion-driven heterogeneous ensemble surrogate-assisted evolutionary algorithm for computationally expensive problems," *Knowl. Based. Syst.*, vol. 236, p. 107747, Jan. 2022.

[16] K. Nishihara and M. Nakata, "Surrogate-assisted Differential Evolution with Adaptation of Training Data Selection Criterion," in *IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2022, pp. 1675–1682.

[17] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing Surrogate-Assisted Evolutionary Computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.

[18] S. Wang and J.-Y. Li, "Probability selection-based surrogate-assisted evolutionary algorithm for expensive optimization," *Appl. Sci. (Basel)*, vol. 15, no. 21, p. 11404, Oct. 2025.

[19] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, 1991.

[20] K. Nishihara and M. Nakata, "Emulation-based adaptive differential evolution: fast and auto-tunable approach for moderately expensive optimization problems," *Complex Intell. Syst.*, vol. 10, no. 3, pp. 3633–3656, Jun. 2024.

[21] L. Yu, Z. Meng, and H. Zhu, "A Hierarchical Surrogate-Assisted Differential Evolution With Core Space Localization," *IEEE Trans. Cybern.*, vol. 55, no. 2, pp. 939–952, Feb. 2025.

[22] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *J. Global Optimiz.*, vol. 11, pp. 341–359, 1997.

[23] K. Nishihara and M. Nakata, "Utilizing the Expected Gradient in Surrogate-assisted Evolutionary Algorithms," in *Annu. Conf. Genet. Evol. Comput. (GECCO) Companion*, Jul. 2023, pp. 447–450.

[24] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Inf. Sci.*, vol. 561, pp. 304–325, Jun. 2021.

[25] X. Lin and Z. Meng, "Surrogate-assisted evolutionary framework with an ensemble of teaching-learning and differential evolution for expensive optimization," *Inf. Sci.*, vol. 680, p. 121137, Oct. 2024.

[26] K. Nishihara, M. Nakata, and S. Watanabe, "Sustainable Performance Improvement of Surrogate-assisted Evolutionary Algorithms using Tabu Search," in *Int. Jt. Conf. Comput. Intell. (IJCCI)*, 2026, pp. 386–404.

[27] D. Han, W. Du, X. Wang, and W. Du, "A surrogate-assisted evolutionary algorithm for expensive many-objective optimization in the refining process," *Swarm Evol. Comput.*, vol. 69, p. 100988, Mar. 2022.

[28] C. Chen, J. Liu, and P. Xu, "Comparison of parallel infill sampling criteria based on Kriging surrogate model," *Sci. Rep.*, vol. 12, no. 1, p. 678, Jan. 2022.

[29] W. Ponweiser, T. Wagner, and M. Vincze, "Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models," in *IEEE Congr. Evol. Comput. (CEC)*. IEEE, Jun. 2008, pp. 3515–3522.

[30] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 1–10.

[31] C. T. Yue, K. V. Price, P. N. Suganthan, J. J. Liang, M. Z. Ali, B. Y. Qu, N. H. Awad, and P. P. Biswas, "Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization," Zhengzhou U., China, and Nanyang Tech. U., Singapore, Tech. Rep. 201911, Nov. 2019.

[32] N. Jethani, M. Sudarshan, I. Covert, S.-I. Lee, and R. Ranganath, "FastSHAP: Real-Time Shapley Value Estimation," in *Int. Conf. Learn. Represent. (ICLR)*, Mar. 2022, pp. 1–23.

[33] X. Shi, J. Gao, L. L. Minku, and X. Yao, "Evolving parsimonious circuits through shapley value-based genetic programming," in *Annu. Conf. Genet. Evol. Comput. (GECCO) Companion*, Jul. 2022.

[34] B. Li, Y. Yang, D. Liu, Y. Zhang, A. Zhou, and X. Yao, "Accelerating surrogate assisted evolutionary algorithms for expensive multi-objective optimization via explainable machine learning," *Swarm Evol. Comput.*, vol. 88, p. 101610, Jul. 2024.

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