Surrogate-assisted Differential Evolution with Adaptation of Training Data Selection Criterion

Kei Nishihara
Graduate School of Engineering Science
Yokohama National University
Yokohama, Japan
nishihara-kei-jv@ynu.jp

Masaya Nakata
Graduate School of Engineering Science
Yokohama National University
Yokohama, Japan
nakata-masaya-tb@ynu.ac.jp

***Abstract*—In surrogate-assisted evolutionary algorithms (SAEAs), the selection criterion of training data is a crucial option to improve the prediction accuracy of surrogate models. In this paper, we hypothesize that a proper selection criterion changes dependent on the problems and/or search situations. Accordingly, this paper proposes a surrogate-assisted differential evolution, which adapts the training data selection criterion to enhance the model accuracy and then optimization performance of our SAEA. In detail, the proposed algorithm builds multiple approximation models under different selection criteria. Then, the best-fitting model is utilized to screen candidate solutions. Experimental results show that the proposed algorithm outperforms the state-of-the-art SAEAs on a single-objective optimization benchmark suite up to 100 dimensions.**

***Index Terms*—surrogate-assisted evolutionary algorithm, adaptive data selection, radial basis function, differential evolution**

I. Introduction

Surrogate-assisted evolutionary algorithms (SAEAs) [1] are a representative methodology to solve expensive optimization problems (EOPs), wherein the fitness evaluation is computationally or financially expensive. A general idea of SAEAs is to perform pre-screening of candidate solutions by machine learning models as surrogates. Typically, the models are used to estimate the fitness values of candidate solutions without any real function evaluation, and thus, improving the model accuracy of prediction is crucial to enhancing the performance of SAEAs [2]. Hereinafter, SAEAs for single-objective optimization problems (SOPs) are discussed in this paper if not stated differently.

Over the last two decades, various SAEAs have been introduced in terms e.g., machine learning techniques and infill criteria [3], [4], which have been useful options in designing SAEAs. Those works empirically demonstrated the importance of using proper options for particular problems. Besides, presented observations have also suggested that the performance of SAEAs depends on the setting of surrogates and training data [2], [5], e.g., hyper-parameters and data size, respectively.

Consequently, certain works have proposed adaptive SAEAs, which adjust surrogate configurations, e.g., model type, model settings, and training data during a single run. We briefly summarize related works below.

- ***Model Type***: SUMO [6] and ASMDE/ASMPSO [7] adaptively choose a model with minimum root-mean-square error (RMSE) from heterogeneous machine learning models. ASAGA [8] utilizes the correlation coefficient between true and predicted fitness values to select an approximation model.
- ***Model Settings***: For the radial basis function (RBF) fitting model, aRBF-NFO [9] and SACOBRA MQ-Cubic [10] adaptively select the type of kernel functions in terms of RMSE and mean absolute error (MAE), respectively. SACOBRA MQ-Cubic and RBFBS [11] adapt the epsilon value in kernels of the RBF model utilizing MAE and RMSE, respectively. For the DACE model [12] known as an implementation of the Kriging model, the length scale parameter defined in the Gaussian kernel (or correlation function) is adapted by the Hookes-Jeeves method [13]. As other examples, Sa-DE-DPS [14] and iDEaSm [15] adapt the length scale parameter using the Differential Evolution algorithm (DE) [16] and covariance matrix, respectively.
- ***Training Data***: However, to the best of our knowledge, few works intend to adapt training data. HESNFO [17] and DSS-DE [18] use bootstrap sampling to construct multiple RBF models. Then HESNFO ensembles some highly accurate models while DSS-DE does not select models but searches on every model.

While few works have attempted to adapt the training data as aforementioned, many works have defined various selection criteria of training data and shown its impact on improving performance. For example, RBFBS constructs the RBF model with all data in the archive and this helps to grasp the entire fitness landscape. SHPSO [19] uses superior solutions in terms of fitness values to simply improve the model accuracy. SAHO [20] collects data near the current population to raise the accuracy in a marginal area. The characteristics of popular training data selection criteria are introduced in Section III-A.

Despite the dependency of the training data on the model accuracy [2], [5], the training data selection criterion is typically fixed through the search process in most SAEAs.

978-1-6654-8768-9/22/$31.00 ©2022 IEEE

However, a proper selection criterion can change dependent on the problems and/or search situations since surrogates with different criteria have different properties [21]. For example, ESAO [22] and S-JADE [23] combine two different criteria to take advantage of the strength of each surrogate. Accordingly, this paper presents Surrogate-assisted Differential Evolution with Adaptation of Training Data Selection Criterion (SADE-ATDSC), which adapts the training data selection criterion during the run. Our proposal builds multiple RBF models under different selection criteria at every generation, then pre-screens them utilizing the model accuracy at every generation. This idea enables SADE-ATDSC to use a reliable model tailored to problems and search situations, reducing the probability of bad prediction caused by, e.g., overfitting.

Note that SADE-ATDSC does not adapt surrogate settings to reveal the impact of adaptation of the training data selection criterion on the performance of SAEAs. Note also that DDEA-MESS [21] can be the most related work, and it conducts an ensemble of three different criteria. While DDEA-MESS decides the criterion to be used based on predefined scheduling, SADE-ATDSC takes the advantage of using the model accuracy for adaptation. Thus, the adaptation enables SADE-ATDSC to fit its algorithm to problems or search situations more rapidly. To the best of our knowledge, SADE-ATDSC is the first study to adapt the training data selection criterion.

This paper is organized as follows. Section II introduces DE framework and the RBF model. Section III defines the training data selection criteria and then explains the mechanism of SADE-ATDSC. In Section IV, we compare SADE-ATDSC with state-of-the-art SAEAs on CEC 2013 benchmark suite [24]. Section V presents additional insights to confirm the effectiveness of SADE-ATDSC. Finally, Section VI describes our conclusion with future work.

II. Preliminary

This section introduces DE and then the RBF model. This paper considers real-parameter bound-constrained SOPs with each objective function, $f : \mathbb{R}^D \to \mathbb{R}$, where D is the problem dimension.

A. Differential Evolution

DE is a population-based evolutionary algorithm for real-parameter SOPs.

DE initializes its population $\mathcal{P} = \{\boldsymbol{x}_i\}_{i=1}^N$ first. N is the population size as a DE hyper-parameter. Each solution $\boldsymbol{x}_i = [x_{i,1}, \ldots, x_{i,D}]^\mathsf{T}$ is sampled from a uniform distribution with a range $x_{i,j} \in [x_j^l, x_j^u]$, where x_j^l and x_j^u are the lower and upper values for the j-th decision variable, respectively.

In each generation, DE generates a mutant solution, $\boldsymbol{v}_i = [v_{i,1}, \cdots, v_{i,D}]^\mathsf{T}$ for each $\boldsymbol{x}_i$, with an employed mutation strategy. We use the *best/1* mutation strategy as follows;

$$\boldsymbol{v}_i = \boldsymbol{x}_{best} + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2}), \tag{1}$$

where $\boldsymbol{x}_{r_1}$ and $\boldsymbol{x}_{r_2}$ are mutually exclusive solutions randomly selected from the current population $\mathcal{P}$. They also must be different from $\boldsymbol{x}_i$. $\boldsymbol{x}_{best}$ is the solution with the best fitness in $\mathcal{P}$. $F \in [0, 1]$ is a scaling factor as a DE hyper-parameter.

Then, for each $\boldsymbol{x}_i$, DE generates a trial vector, $\boldsymbol{u}_i = [u_{i,1}, \cdots, u_{i,D}]^\mathsf{T}$ with a crossover strategy and a crossover rate as a hyper-parameter, $CR \in [0, 1]$. In this paper, we use the *binomial* crossover strategy as follows;

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand(0,1) \le CR \text{ or } j = j_{rand}, \\ x_{i,j} & \text{otherwise,} \end{cases} \tag{2}$$

where j_{rand} is an integer randomly chosen from 1 to D and $rand(0, 1)$ is a real random value in $(0, 1)$ sampled from a uniform distribution.

Finally, in the solution selection process, after $\boldsymbol{u}_i$ is evaluated, each $\boldsymbol{x}_i$ is replaced with $\boldsymbol{u}_i$ if $f(\boldsymbol{u}_i) \le f(\boldsymbol{x}_i)$ for minimization problems. These procedures from mutation to solution selection are repeated until the termination criteria are met.

B. RBF fitting model

The RBF model is a nonparametric fitting method for scattered point sequences of arbitrary dimensionality using a radial basis function (RBF), $\phi(r)$, also called kernel function. We use the RBF model because it has a scalability of performance to the problem dimension [5] and requires a relatively small execution time [2]. The definition follows OPUS-RBF [25].

Let $\mathcal{X}$ is a finite set of mutually distinct points $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ and $\boldsymbol{f}$ is corresponding function values $\boldsymbol{f} = [f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_n)]^\mathsf{T}$. For any point $\boldsymbol{x}$, the approximation of $f(\boldsymbol{x})$ is defined as follows;

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \phi\left(\|\boldsymbol{x} - \boldsymbol{x}_i\|\right) + p(\boldsymbol{x}), \tag{3}$$

where $\lambda_i \in \mathbb{R}$ is the i-th element of weight coefficient vector $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]^\mathsf{T}$ and $p(\boldsymbol{x}) = \boldsymbol{c}^\mathsf{T}\boldsymbol{x} + c_0$ ($\boldsymbol{c} \in \mathbb{R}^D, c_0 \in \mathbb{R}$) is a linear polynomial function. We use the *cubic* kernel $\phi(r) = r^3$, which is one of the most popular kernels.

Three parameters in Eq. (3), i.e., $\boldsymbol{\lambda}$, $\boldsymbol{c}$, and c_0, can be obtained by solving the following equation;

$$\begin{bmatrix} \Phi & P \\ P^\mathsf{T} & 0_{mat} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{c}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix}, \tag{4}$$

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

III. SADE-ATDSC

This section begins by defining the training data selection criterion and then describes the algorithm of SADE-ATDSC.

To begin with, we use the following notations.

- FE — the number of conducted fitness evaluations so far
- $\mathcal{A}$ — the archive consisting of all evaluated solutions $\boldsymbol{x}$ so far and corresponding fitness values, given by $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^{FE}$
- N — the population size

$\mathcal{P}$	the DE population, given by $\{\boldsymbol{x}_i\}_{i=1}^{N}$
$\mathcal{F}$	a set of corresponding fitness values of solutions $\boldsymbol{x}_i \in \mathcal{P}$, given by $\{f(\boldsymbol{x}_i)\}_{i=1}^{N}$
M	the number of criteria and candidate RBF models
n	a hyper-parameter to define data size
$\mathcal{D}_m$	training data obtained by m-th criterion, given by $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}$, where its size depends on criterion
$\mathcal{S}$	a set of candidate RBF models, i.e., $\{\hat{f}_m\}_{m=1}^{M}$, where a model $\hat{f}_m$ is trained with $\mathcal{D}_m$
$\mathcal{R}$	a set of RMSE r_m for $\hat{f}_m$, given by $\{r_m\}_{m=1}^{M}$

A. Definition of the training data selection criteria

Several training data selection criteria have been proposed. Algorithm 1 shows the detailed procedure to obtain training data as a pseudo function to be used in Section III-B, where m stands for the ID of criterion. In this paper, we introduce four criteria; $m = 1$: *All Data*, $m = 2$: *Current Population*, $m = 3$: *Recent Data*, and $m = 4$: *Neighbor*.

All Data is the most popular criterion to construct the global surrogate using all data in archive $\mathcal{A}$. This criterion is suitable to reflect the entire fitness landscape and thus helps SAEAs select promising solutions [21]. S-JADE, RBFBS, and SACOBRA MQ-Cubic adopt this criterion.

Current Population and *Recent Data* can be considered as the local models because they use a local part of $\mathcal{A}$. The local models are applied to enhance the model accuracy in the corresponding search space [21]. As the number of data increases, the accuracy improves in the area [21]. As shown in **case** 2 in Algorithm 1, *Current Population* constructs the model using the current population $\mathcal{P}$ and corresponding fitness values $\mathcal{F}$ as the training data. iDEaSm and aRBF-NFO build the model with this criterion. *Recent Data* adopts n recently evaluated data as explained in **case** 3 in Algorithm 1. *Recent Data* may chase the search direction as compared to *Current Population*. GPEME [26] is the representative SAEA for this criterion.

Neighbor can be the global-oriented local criterion because it puts the marginal area in perspective relative to *Current Population*. S-JADE, GORS-SSLPSO [27], and SAHO are examples in this class. Note that S-JADE ensembles both *All Data* and *Neighbor*. The definition of the "neighbor" varies for SAEAs. We define *Neighbor* as the criterion that makes the union of selected n data for each solution $\boldsymbol{x}_i$ in the current population $\mathcal{P}$ from $\mathcal{A}$. This follows that of SAHO, where authors argue that the constructed model has higher accuracy since data are closer to the population [20].

According to these papers on SAEAs, the model accuracy depends on the training data selection criterion and it is important to select the criterion appropriately. In this regard, we adapt the criterion based on problems and search situations to enhance the model accuracy and then the optimization performance.

B. Mechanism

SADE-ATDSC preliminarily prepares four criteria mentioned in Section III-A and pre-screens them to improve the reliability of the model In detail, SADE-ATDSC constructs the same number of RBF models with different training data formed by these criteria at every generation. Then SADE-ATDSC employs the best model having the minimum RMSE among the candidates. For simplicity and time-saving, we use the hold-out validation with validation data rate $\delta \in [0, 1]$.

Algorithm 1 *Get-Training-Data*$(\mathcal{A}, \mathcal{P}, \mathcal{F}, FE, N, n, m)$

switch m **do**
  case 1 — *// All Data*
    return $\mathcal{D}_{AD} = \mathcal{A}$

  case 2 — *// Current Population*
    return $\mathcal{D}_{CP} = \{(\mathcal{P}, \mathcal{F})\}$

  case 3 — *// Recent Data*
    return $\mathcal{D}_{RD} = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=FE-n+1}^{FE}$

  case 4 — *// Neighbor*
    $\mathcal{D}_{NB} = \emptyset$
    for $i = 1$ **to** N **do**
      $\mathcal{D}_{temp} \leftarrow$ Select the Nearest n data of $\boldsymbol{x}_i \in \mathcal{P}$ from $\mathcal{A}$
      $\mathcal{D}_{NB} = \mathcal{D}_{NB} \cup \mathcal{D}_{temp}$, $\mathcal{D}_{temp} = \emptyset$
    end for
    return $\mathcal{D}_{NB}$
end switch

Algorithm 2 shows the mechanism of SADE-ATDSC. At the beginning of the search, SADE-ATDSC prepares the population $\mathcal{P} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ using Latin Hypercube Sampling (LHS) and evaluates all solutions in $\mathcal{P}$. Then the initial archive $\mathcal{A}$ is created.

For every generation, SADE-ATDSC recreates $\mathcal{P}$ and $\mathcal{F}$ with top N solutions that have the top N fitness values in $\mathcal{A}$ first. Then pre-screening of the RBF models is performed in the first half of the main loop. After two sets $\mathcal{S}$ and $\mathcal{R}$ for the models and RMSE are prepared as empty, respectively, and for each criterion, the following processes are conducted. First, SADE-ATDSC selects training data $\mathcal{D}_m$ by the pseudo function *Get-Training-Data*$(\mathcal{A}, \mathcal{P}, \mathcal{F}, FE, N, n, m)$ defined in Algorithm 1 under the m-th criterion. Next, $\mathcal{D}_m$ is randomly split into $\mathcal{D}_{m,tra}$ and $\mathcal{D}_{m,val}$ by the ratio of $(1 - \delta) : \delta$ for the hold-out validation. Then the m-th RBF model $\hat{f}_m$ is constructed with $\mathcal{D}_{m,tra}$. SADE-ATDSC validates $\hat{f}_m$ using $\mathcal{D}_{m,val}$, wherein RMSE r_m is obtained. Subsequently, $\hat{f}_m$ and r_m are added to $\mathcal{S}$ and $\mathcal{R}$, respectively. After finishing the validation of all models, the best model $\hat{f}^*$ is determined by selecting model $\hat{f}_m$ that derived the minimum RMSE r_m.

In the latter half of the main loop (from line 14), SADE-ATDSC screens candidate offspring solutions $\{\boldsymbol{u}_i\}_{i=1}^{N}$ using $\hat{f}^*$. Specifically, SADE-ATDSC executes the *best/1* mutation and the *binomial* crossover to every solution $\boldsymbol{x} \in \mathcal{P}$, as introduced in Section II-A. The only different point from the original *DE/best/1/bin* is that SADE-ATDSC does neither evaluation of $\{\boldsymbol{u}_i\}_{i=1}^{N}$ with true function nor solution selection. Instead, SADE-ATDSC calculates pseudo fitness values $\hat{f}^*(\boldsymbol{u}_i)$ for all candidates and chooses one with the best pseudo fitness value. Only selected candidate $\boldsymbol{u}^*$ is evaluated with true function to conserve fitness evaluation resources orienting

Algorithm 2 SADE-ATDSC

Initialize $\mathcal{P} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ by LHS and Evaluate
$\mathcal{F} = \{f(\boldsymbol{x}_i)\}_{i=1}^{N}$, $\mathcal{A} = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^{N}$, $FE = N$
while $FE < FE_{\max}$ **do**
  $\mathcal{P}$, $\mathcal{F} \leftarrow$ Select top N data and their fitness values from $\mathcal{A}$
  $\mathcal{S} = \emptyset$, $\mathcal{R} = \emptyset$
  for $m = 1$ **to** M **do**
    $\mathcal{D}_m \leftarrow$ *Get-Training-Data*$(\mathcal{A}, \mathcal{P}, \mathcal{F}, FE, N, n, m)$
    $\mathcal{D}_{m,tra}$, $\mathcal{D}_{m,val} \leftarrow$ Split $\mathcal{D}_m$ by the ratio of $(1-\delta) : \delta$
    $\hat{f}_m \leftarrow$ Build the RBF model with $\mathcal{D}_{m,tra}$
    $r_m \leftarrow$ Get RMSE in Validation of $\hat{f}_m$ with $\mathcal{D}_{m,val}$
    $\mathcal{S} = \mathcal{S} \cup \{\hat{f}_m\}$, $\mathcal{R} = \mathcal{R} \cup \{r_m\}$
  end for
  $\hat{f}^* \leftarrow \hat{f}_m \in \mathcal{S}$ that derived minimum $r_m \in \mathcal{R}$
  for $i = 1$ **to** N **do**
    $\boldsymbol{v}_i \leftarrow$ *best/1-mutation* with $\mathcal{P}$, F — See Eq. (1)
    $\boldsymbol{u}_i \leftarrow$ *binomial-crossover* with $\boldsymbol{x}_i$, $\boldsymbol{v}_i$, CR — See Eq. (2)
    $\hat{f}^*(\boldsymbol{u}_i) \leftarrow$ Evaluate $\boldsymbol{u}_i$ with $\hat{f}^*$
  end for
  $\boldsymbol{u}^* = \underset{\{\boldsymbol{u}_i\}_{i=1}^{N}}{\arg} \underset{\{\hat{f}^*(\boldsymbol{u}_i)\}_{i=1}^{N}}{\min} \left(\hat{f}^*(\boldsymbol{u}_i)\right)$
  $f(\boldsymbol{u}^*) \leftarrow$ Evaluate $\boldsymbol{u}^*$ with the true function f
  $FE = FE + 1$
  $\mathcal{A} = \mathcal{A} \cup \{(\boldsymbol{u}^*, f(\boldsymbol{u}^*))\}$
end while

EOPs. At the end of the main loop, SADE-ATDSC adds $\boldsymbol{u}^*$ and its fitness value $f(\boldsymbol{u}^*)$ to the archive $\mathcal{A}$.

IV. Experiment

This section evaluates the effect of adaptation of training data selection criterion by comparing performances of state-of-the-art SAEAs and SADE-ATDSC.

A. Experimental Design

We use IEEE CEC2013 real-parameter single-objective benchmark function suite [24]. This benchmark suite consists of uni-modal functions F1-F5, multi-modal functions F6-F20, and composition functions F21-F28 selected some functions from F1-F20. To evaluate the scalability of the performance of SADE-ATDSC against the increase of problem dimension D, we set $D \in \{10, 30, 50, 100\}$. Thus, we conduct 112 cases (28×4) in total. The search range (bound constraint) is $[-100, 100]^D$ for all functions. All experiments are conducted with 4.8 GHz CPU and 16GB RAM.

We compare SADE-ATDSC with S-JADE and SAHO as RBF-based state-of-the-art SAEAs and aRBF-NFO as a representative of adaptive SAEAs. Hyper-parameter settings follow the original papers and their details are as stated below. For S-JADE, $N = 30$, $F_{out} = 0.5$, $CR_{out} = 0.75$, $p_{pbest_{out}} = 0.05$, $F_{in} = 0.5$, $CR_{out} = 0.5$, $p_{pbest_{in}} = 0.1$, $std_F = 0.1$, $std_{CR} = 0.1$, $L = 10$, $\epsilon = 0.01$, $c = 0.1$, $FE_{max_{in}} = 2,000$, kernel = cubic, and $r = rand(0,\ 1.25)$. For SAHO, $N = 5D (D < 50)$ or $100 + \lfloor D/10 \rfloor (D \geq 50)$, $F = 0.9$, $CR = 0.5$, $K = 30$, $neighbor = 5D(D < 50)$ or $D(D \geq 50)$, and kernel = cubic. For aRBF-NFO, $N = 100$, $c = 1.3$, $CR = 0.1$, $|test| = N/2$, $\epsilon = 1$, kernel $\in$ {cubic, multi-quadric, inverse multi-quadric, thin-plate-spline, gaussian }. For SADE-ATDSC, $N = n = 100$, $F = 0.5$, $CR = 0.9$, $\delta = 0.2$, kernel = cubic, criteria $\in$ {*All Data, Current Population, Recent Data, Neighbor*}.

The maximum number of FEs FE_{max} is strictly set to $1,000$, i.e., we forcedly terminate each run when the number of fitness evaluations reaches $1,000$. Each performance is evaluated with the best fitness discovered in $\mathcal{P}$ of each method and reported as the mean value of 21 independent runs.

The Wilcoxon signed-rank test is performed for each function of each dimension D under the significance level of 0.05 to check the statistical significance. The results are reported as "+" (SADE-ATDSC significantly underperforms an algorithm), "−" (SADE-ATDSC significantly outperforms an algorithm), or "∼" (we cannot say that there is a significant difference).

B. Result

TABLE I shows the best fitness values discovered at $1,000$ FEs for $D \in \{10, 30, 50, 100\}$. Note that the best values among methods are highlighted in green and bold.

The number of "−", which stands for the superiority of SADE-ATDSC with statistical significance, is always larger than that of "+", which stands for the inferiority of SADE-ATDSC. In detail, the best fitness values are found by SADE-ATDSC for 17, 14, 17, and 22 functions out of 28 in the order of $D = 10$, 30, 50, and 100, respectively. "−" is obtained from 12 to 23 in every compared pair and only 41 "+" are obtained within 336 cases in sum. SADE-ATDSC especially derives superior results in multi-modal (F6-F28) functions. From these results, we can confirm the effectiveness of the adaptation of the training data selection criterion.

TABLE II summarizes Wilcoxon signed-rank tests at 300, 500, and $1,000$ (baseline) FEs for $D \in \{10, 30, 50, 100\}$ and Fig. 1 presents the average rank across all problems at 200, 300, $\ldots$, $1,000$ (baseline) FEs for $D \in \{10, 30, 50, 100\}$. In general, SADE-ATDSC derives a superior or competitive performance from other SAEAs, judging from the comparison between the number of "−" and "+". From the point of view of the problem dimension D, SADE-ATDSC has a scalability of performance to D. Although the frequency of SADE-ATDSC ranking first is relatively decreasing with the increase of problem dimension D in Fig. 1, the number of "−" is much larger than that of "+", especially at $1,000$ for $D \in \{50, 100\}$ in TABLE II. However, SADE-ATDSC has week scalability to the FEs because SADE-ATDSC ranks second in Fig. 1 and the number of "−" and "+" rapidly decreases and increases, respectively, against S-JADE ($D \in \{50, 100\}$) and SAHO ($D = 100$) in TABLE II, when FEs decrease under 500. Still, the effectiveness of SADE-ATDSC remains intact since the number of "−" surpasses that of "+" in 32 cases out of 36 cases in TABLE II.

V. Discussion

A. Computational Time

We compare computational time among four SAEAs obtained from the experiment conducted in TABLEs I, II, and Fig. 1. Since EOPs can be a case more financially than

TABLE I
THE BEST FITNESS VALUES DISCOVERED AT $1,000$ FITNESS EVALUATIONS (FES) FOR $D \in \{10, 30, 50, 100\}$.

	$D = 10$				$D = 30$			
	S-JADE	SAHO	aRBF-NFO	SADE-ATDSC	S-JADE	SAHO	aRBF-NFO	SADE-ATDSC
F1	7.30E-06 −	**2.18E-28** +	1.74E+03 −	5.09E-20	6.78E+00 ∼	**1.20E-15** +	1.87E+04 −	8.43E+01
F2	2.91E+06 −	**9.66E+05** ∼	1.90E+07 −	1.38E+06	8.81E+07 −	**1.16E+07** +	3.75E+08 −	2.88E+07
F3	5.50E+09 −	6.17E+10 −	4.90E+09 −	**6.03E+03**	1.49E+15 −	2.56E+17 −	8.60E+13 −	**7.39E+09**
F4	**1.88E+04** +	2.56E+04 +	6.16E+04 ∼	4.43E+04	**8.68E+04** +	1.22E+05 +	1.52E+05 +	1.88E+05
F5	4.34E+01 −	2.19E-03 −	1.67E+03 −	**5.21E-04**	3.13E+03 −	**1.79E+02** ∼	2.05E+04 −	2.25E+02
F6	**7.97E+00** ∼	9.33E+00 ∼	1.43E+02 −	8.53E+00	1.04E+02 −	**4.79E+01** ∼	1.02E+03 −	5.28E+01
F7	1.07E+02 −	3.37E+02 −	1.12E+02 −	**2.87E+01**	1.53E+04 −	3.00E+05 −	2.47E+03 −	**2.08E+02**
F8	**2.07E+01** ∼	2.08E+01 ∼	2.08E+01 ∼	**2.07E+01**	**2.12E+01** ∼	**2.12E+01** ∼	**2.12E+01** ∼	**2.12E+01**
F9	6.72E+00 ∼	7.42E+00 −	9.16E+00 −	**5.68E+00**	3.66E+01 −	3.13E+01 −	4.35E+01 −	**2.42E+01**
F10	4.36E-01 ∼	6.72E-01 −	2.86E+02 −	**3.29E-01**	5.71E+01 ∼	**1.46E+00** +	2.96E+03 −	7.53E+01
F11	4.08E+01 −	4.75E+01 −	4.13E+01 −	**1.93E+01**	2.85E+02 −	2.48E+02 −	4.75E+02 −	**1.50E+02**
F12	5.33E+01 −	3.50E+01 −	7.90E+01 −	**2.27E+01**	3.14E+02 −	2.21E+02 −	5.56E+02 −	**1.68E+02**
F13	5.49E+01 ∼	6.26E+01 −	8.63E+01 −	**4.75E+01**	3.18E+02 ∼	3.07E+02 ∼	5.93E+02 −	**2.98E+02**
F14	1.75E+03 −	1.23E+03 ∼	**1.04E+03** ∼	1.06E+03	7.82E+03 ∼	**6.22E+03** +	8.05E+03 ∼	7.73E+03
F15	1.97E+03 −	**1.43E+03** ∼	2.03E+03 −	1.59E+03	8.67E+03 ∼	**6.32E+03** +	8.80E+03 ∼	8.77E+03
F16	2.40E+00 ∼	**2.09E+00** +	2.35E+00 +	2.66E+00	4.59E+00 ∼	4.55E+00 ∼	**4.26E+00** ∼	4.55E+00
F17	5.20E+01 −	**3.31E+01** ∼	7.25E+01 −	3.89E+01	**2.74E+02** +	**2.74E+02** +	1.01E+03 −	3.36E+02
F18	5.77E+01 ∼	**3.86E+01** +	1.11E+02 −	6.18E+01	**2.86E+02** +	2.96E+02 +	1.21E+03 −	3.55E+02
F19	8.33E+00 −	1.88E+02 −	1.77E+01 −	**2.29E+00**	3.58E+04 −	3.75E+05 −	7.74E+03 −	**2.12E+01**
F20	4.37E+00 −	4.56E+00 −	4.37E+00 −	**3.92E+00**	1.50E+01 −	1.50E+01 −	1.50E+01 −	**1.48E+01**
F21	4.26E+02 −	4.46E+02 −	4.97E+02 −	**3.91E+02**	2.46E+03 −	4.24E+03 −	2.65E+03 −	**3.24E+02**
F22	1.94E+03 −	1.59E+03 −	1.32E+03 ∼	**1.22E+03**	8.42E+03 −	**6.24E+03** +	8.67E+03 ∼	7.77E+03
F23	2.24E+03 −	1.77E+03 −	2.17E+03 −	**1.43E+03**	9.20E+03 ∼	**7.14E+03** +	9.51E+03 −	8.98E+03
F24	2.18E+02 ∼	2.16E+02 ∼	2.24E+02 −	**2.15E+02**	2.98E+02 −	2.84E+02 −	3.18E+02 −	**2.67E+02**
F25	**2.17E+02** ∼	**2.17E+02** ∼	2.25E+02 −	2.18E+02	3.14E+02 −	3.02E+02 −	3.36E+02 −	**2.89E+02**
F26	2.05E+02 ∼	2.11E+02 ∼	2.29E+02 ∼	**1.92E+02**	**3.32E+02** ∼	3.48E+02 ∼	3.61E+02 ∼	3.39E+02
F27	**4.30E+02** +	5.22E+02 ∼	6.00E+02 −	4.86E+02	1.17E+03 −	1.04E+03 −	1.41E+03 −	**9.56E+02**
F28	8.28E+02 −	1.41E+03 −	8.74E+02 −	**4.19E+02**	4.63E+03 −	7.39E+03 −	4.45E+03 −	**7.35E+02**
+/ − / ∼	2/16/10	4/14/10	1/22/5		3/15/10	10/12/6	1/21/6	
average rank	2.571	2.571	3.393	**1.464**	2.607	2.071	3.429	**1.893**

	$D = 50$				$D = 100$			
	S-JADE	SAHO	aRBF-NFO	SADE-ATDSC	S-JADE	SAHO	aRBF-NFO	SADE-ATDSC
F1	1.57E+02 ∼	**2.82E-06** +	4.45E+04 −	3.15E+02	3.66E+03 −	6.16E+03 −	3.17E+05 −	**2.66E+03**
F2	1.61E+08 ∼	**5.26E+07** +	1.56E+09 −	1.37E+08	6.39E+08 ∼	9.00E+08 −	1.12E+10 −	**5.62E+08**
F3	7.54E+12 −	3.73E+16 −	9.91E+15 −	**1.24E+11**	5.16E+20 −	9.67E+23 −	4.14E+23 −	**4.69E+19**
F4	**1.26E+05** +	2.07E+05 +	2.36E+05 ∼	2.54E+05	**2.92E+05** +	4.13E+05 ∼	4.61E+05 ∼	4.53E+05
F5	1.13E+04 −	2.42E+03 ∼	5.38E+04 −	**2.32E+03**	2.79E+04 −	2.87E+04 −	3.23E+05 −	**1.59E+04**
F6	2.51E+02 −	**4.74E+01** +	3.92E+03 −	7.16E+01	2.72E+03 −	3.43E+03 −	9.65E+04 −	**1.37E+03**
F7	2.56E+03 −	1.33E+05 −	3.29E+04 −	**2.59E+02**	5.29E+06 −	2.07E+08 −	1.47E+08 −	**9.11E+05**
F8	**2.13E+01** ∼	**2.13E+01** ∼	2.14E+01 ∼	**2.13E+01**	**2.15E+01** ∼	**2.15E+01** ∼	**2.15E+01** ∼	**2.15E+01**
F9	7.01E+01 −	5.78E+01 −	7.97E+01 −	**4.91E+01**	1.60E+02 −	1.54E+02 −	1.71E+02 −	**1.22E+02**
F10	4.69E+02 −	**4.64E+01** +	7.24E+03 −	2.41E+02	3.12E+03 ∼	4.34E+03 −	4.94E+04 −	**2.91E+03**
F11	5.40E+02 −	**2.69E+02** +	1.07E+03 −	3.29E+02	**1.41E+03** ∼	1.73E+03 −	5.00E+03 −	1.45E+03
F12	5.91E+02 −	**2.79E+02** +	1.25E+03 −	3.83E+02	**1.53E+03** ∼	1.67E+03 −	5.21E+03 −	1.54E+03
F13	6.14E+02 +	**6.03E+02** +	1.33E+03 −	6.71E+02	**1.56E+03** +	1.70E+03 ∼	5.30E+03 −	1.70E+03
F14	1.44E+04 −	1.43E+04 −	1.45E+04 −	**1.13E+04**	3.33E+04 −	3.43E+04 −	3.41E+04 −	**3.06E+04**
F15	1.61E+04 ∼	1.62E+04 ∼	1.62E+04 ∼	**1.59E+04**	3.38E+04 ∼	3.40E+04 ∼	3.42E+04 ∼	**3.32E+04**
F16	5.58E+00 ∼	5.66E+00 ∼	**5.22E+00** ∼	5.35E+00	5.86E+00 ∼	5.88E+00 ∼	5.81E+00 ∼	**5.75E+00**
F17	**5.05E+02** +	6.09E+02 +	2.44E+03 −	6.95E+02	**1.35E+03** +	1.89E+03 +	1.01E+04 −	2.21E+03
F18	**5.54E+02** +	6.21E+02 +	2.88E+03 −	7.72E+02	**1.39E+03** +	1.81E+03 +	1.04E+04 −	2.24E+03
F19	4.12E+04 −	3.05E+03 −	2.02E+06 −	**6.79E+01**	1.33E+06 −	8.46E+05 −	5.17E+07 −	**4.51E+04**
F20	2.49E+01 −	2.50E+01 −	2.50E+01 −	**2.48E+01**	**5.00E+01** ∼	**5.00E+01** ∼	**5.00E+01** ∼	**5.00E+01**
F21	4.76E+03 −	7.46E+03 −	7.03E+03 −	**8.62E+02**	9.26E+03 ∼	1.45E+04 −	2.12E+04 −	**8.84E+03**
F22	1.57E+04 −	1.57E+04 −	1.62E+04 −	**1.28E+04**	3.49E+04 −	3.50E+04 −	3.58E+04 −	**2.99E+04**
F23	1.67E+04 ∼	1.68E+04 ∼	1.71E+04 ∼	**1.66E+04**	**3.54E+04** ∼	3.57E+04 ∼	3.59E+04 ∼	**3.54E+04**
F24	3.79E+02 −	3.37E+02 ∼	4.23E+02 −	**3.35E+02**	5.97E+02 −	5.78E+02 −	1.65E+03 −	**5.42E+02**
F25	4.14E+02 −	3.77E+02 ∼	4.65E+02 −	**3.66E+02**	6.82E+02 −	6.42E+02 −	1.11E+03 −	**5.94E+02**
F26	4.59E+02 −	5.15E+02 −	4.97E+02 −	**4.10E+02**	7.22E+02 −	7.49E+02 −	7.65E+02 −	**6.07E+02**
F27	2.05E+03 −	1.73E+03 −	2.38E+03 −	**1.55E+03**	4.23E+03 −	4.14E+03 −	6.27E+03 −	**3.54E+03**
F28	8.39E+03 −	1.24E+04 −	8.70E+03 −	**1.60E+03**	2.17E+04 ∼	3.23E+04 −	3.38E+04 −	**2.10E+04**
+/ − / ∼	4/18/6	10/11/7	0/23/5		4/13/11	2/19/7	0/22/6	
average rank	2.321	2.321	3.679	**1.679**	1.875	2.768	3.804	**1.554**

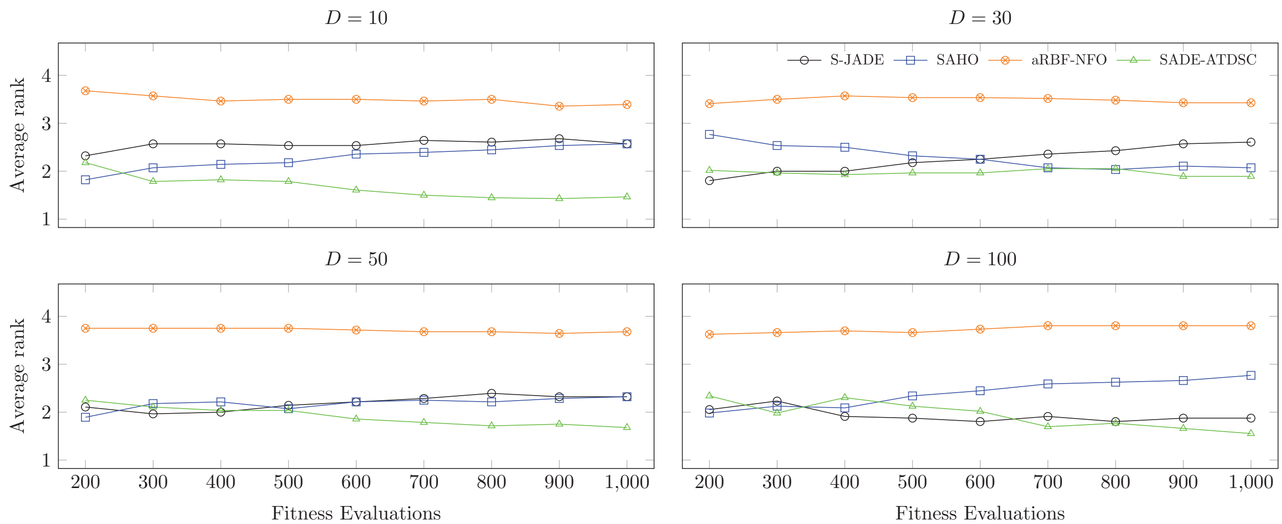

Fig. 1. Average rank across all problems. The results are reported at 200, 300, ..., 1,000 (baseline) fitness evaluations (FEs) for $D \in \{10, 30, 50, 100\}$.

TABLE II
SIGNIFICANT DIFFERENCES REGARDING FINDINGS FOR "+/ − / ∼" BETWEEN SADE-ATDSC AND STATE-OF-THE-ART SAEAS.

D	FE	vs S-JADE	vs SAHO	vs aRBF-NFO
10	300	4/16/ 8	7/ 8/13	1/22/ 5
	500	1/18/ 9	5/10/13	0/23/ 5
	1,000	2/16/10	4/14/10	1/22/ 5
30	300	6/11/11	5/12/11	0/20/ 8
	500	5/14/ 9	7/10/11	1/21/ 6
	1,000	3/15/10	10/12/ 6	1/21/ 6
50	300	11/10/ 7	10/ 8/10	0/21/ 7
	500	8/11/ 9	10/11/ 7	0/23/ 5
	1,000	4/18/ 6	10/11/ 7	0/23/ 5
100	300	6/ 9/13	13/ 6/ 9	0/21/ 7
	500	13/ 8/ 7	9/11/ 8	0/23/ 5
	1,000	4/13/11	2/19/ 7	0/22/ 6

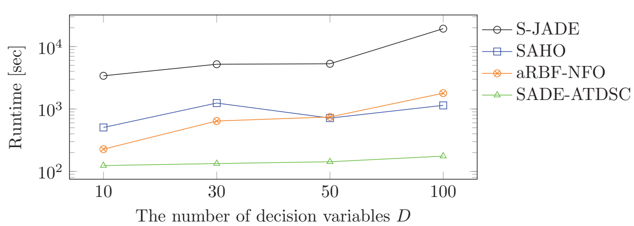

Fig. 2. Averaged computational time obtained at 1,000 FEs from 21 runs of experiments with all 28 functions for $D \in \{10, 30, 50, 100\}$.

computationally expensive, the computational time of the algorithm is crucial. Fig. 2 shows the averaged computational time obtained at 1,000 FEs from 21 runs of experiments with all 28 functions. The results are reported for each dimension $D \in \{10, 30, 50, 100\}$.

From the figure, SADE-ATDSC ranked first for all dimensions. SADE-ATDSC outperforms all the methods at 1,000 FEs, with execution times ranging from approximately 1/2 to 1/110 of the comparative methods. There are two factors at play here. First, SADE-ATDSC uses the RBF model to screen the generated candidate solutions while S-JADE and SAHO perform iterated evolutionary searches on the RBF models. Second, since the number of candidates for adaptation is four and five for SADE-ATDSC and aRBF-NFO, respectively, the difference in the number of the model building increases with the number of fitness evaluations. The computational time of SADE-ATDSC is not so much as get longer against the increase of dimension D, which implies its usefulness in high dimensional problems. Note that SAHO becomes faster at $D = 50$ than at $D = 30$, but this causes by settings of the population size and the neighbor size.

B. Selection Tendency by Adaptation in Total

Next, we analyze the adaptation results obtained in the main experiment, i.e., the one conducted in Section IV. Fig. 3 shows the selection ratio in adaptation obtained by SADE-ATDSC for $D \in \{10, 30, 50, 100\}$ through 21 runs. The stacked bar graph indicates the selected ratio of each candidate criteria in sum through 1,000 FEs and 21 runs.

In general, the selection tendency varies depending on the problems and their dimensions. We can confirm that SADE-ATDSC was able to perform adaptation first. In terms of the modality of the functions, SADE-ATDSC frequently uses *Current Population* in uni-modal functions (F1-F5). This may happen because the local-oriented search is preferred in uni-modal functions.

Here, we introduce another perspective. Fig. 4 presents the average rank obtained in an additional experiment, where four unadapted methods set up with different training data selection criteria are used. Interestingly, the adaptation result by SADE-ATDSC in Fig. 3 tracks the result of the additional experiment in Fig. 4. For example, *Current Population* and *Recent Data*, which got high in rankings for $D \in \{10, 30\}$ in

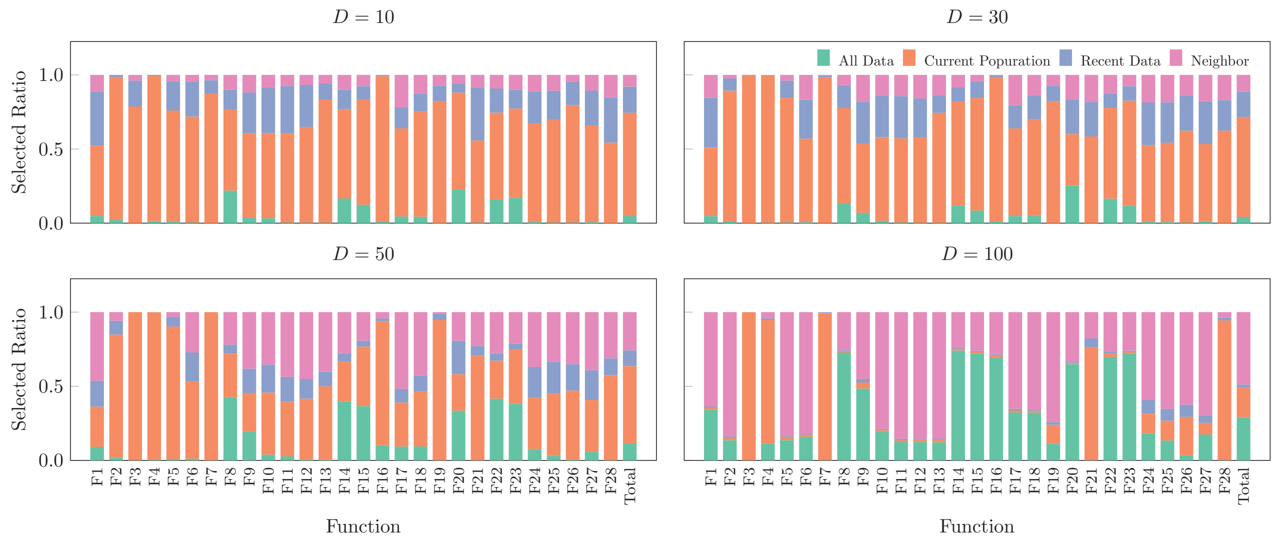

Fig. 3. The selection tendency in adaptation obtained by SADE-ATDSC for $D \in \{10, 30, 50, 100\}$ through 21 runs. Note that the stacked bar graph indicates the selected ratio of each candidate in sum through $1,000$ fitness evaluations (FEs) and 21 runs.

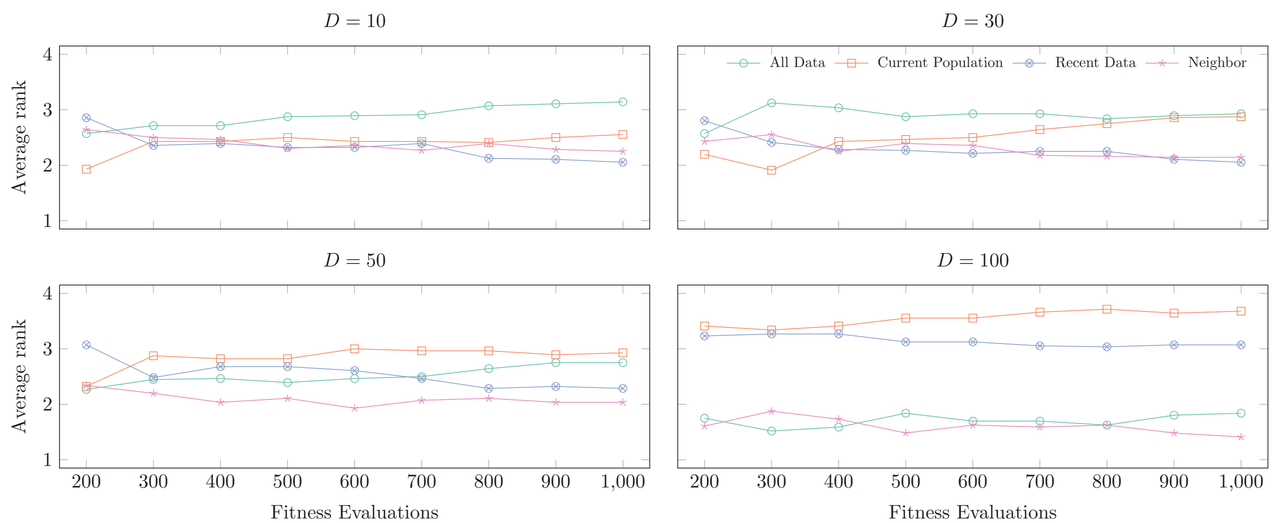

Fig. 4. Average rank across all problems. Four unadapted methods set up with different training data selection criteria are compared.

Fig. 4, are much more used for $D \in \{10, 30\}$, while SADE-ATDSC utilizes *All Data* and *Neighbor* in most cases for $D \in \{50, 100\}$, where these criteria outperformed in Fig. 4. Therefore, the model accuracy-based adaptation by SADE-ATDSC potentially reflects the optimization results and thus we verified the adequacy of adaptation.

C. Selection Tendency by Adaptation in a Certain Problem

Finally, we analyze in terms of FEs whether the adaptation result tracks the training data selection criterion from which the parameter fixed methods derived the best result. As an example, we show Fig. 5, which demonstrates the rank of four unadapted methods set up with different training data selection criteria (top) and the criterion selection tendency in adaptation obtained by SADE-ATDSC with F27 $D = 50$ (bottom). The rank is obtained by the mean value of 21 runs. The adaptation result is reported with stacked curves; each curve indicates the number of the selected criterion in 21 runs.

As a result, the selection tendency (bottom) well tracks the change of rank (top). The selected ratio of *All Data* increases and decreases after 400 and 800 FEs, where *All Data* moves up and down, respectively. Both the rank and the ratio rapidly decrease toward 400 FEs for *Current Population* while *Recent Data* is not adopted very much because it ranks low from first to last. Lastly, *Neighbor* ranks first and consists mostly of the ratio. This tendency also can be evidence to support the validity of the model-accuracy-based adaptation.

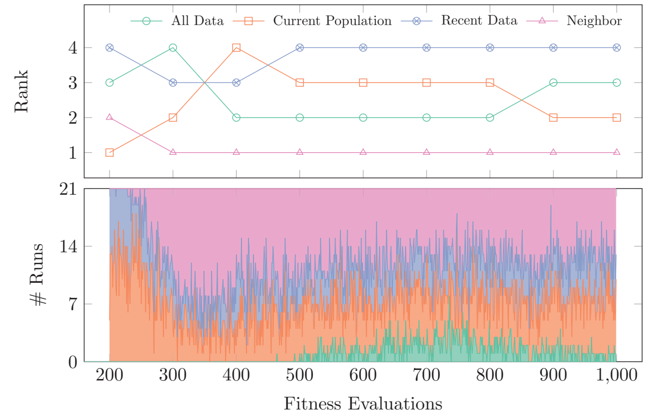

Fig. 5. The rank of four unadapted methods set up with different training data selection criteria (top) and the criterion selection tendency in adaptation obtained by SADE-ATDSC with F27 $D = 50$ (bottom) obtained during 21 runs. Note that the bottom figure is reported with stacked curves; each curve indicates the number of the corresponding criteria. The total number corresponds to the run times.

VI. Conclusion

This paper proposed an adaptive surrogate-assisted differential evolution in a new light. The proposal named SADE-ATDSC adapts the training data selection criterion. The motivation for adaptation was based on the fact that the model accuracy depends on training data. In detail, our proposal pre-screens multiple RBF models trained with different data criteria at every generation. SADE-ATDSC employs the model with the best accuracy and thus pre-screening can select a model considering the problem characteristics and search situations without additional fitness evaluation. Experimental results show that SADE-ATDSC outperforms the state-of-the-art SAEAs up to 100 dimensions. Notably, SADE-ATDSC requires much smaller computational time than others, and this superiority is pronounced in high-dimensional cases.

In our future work, we further investigate the correlation between the stage of search and the effect of adaptation, and then we define the frequency of adaptation. Moreover, our future proposal will also adapt surrogate settings in addition to the training data selection criterion, to improve the performance by finding more effective use of surrogates in SAEAs.

References

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

[2] H. Tong, C. Huang, L. L. Minku, and X. Yao, "Surrogate models in evolutionary single-objective optimization: A new taxonomy and experimental study," *Inf. Sci.*, vol. 562, pp. 414–437, Jul. 2021.

[3] T. Sonoda and M. Nakata, "Multiple Classifiers-Assisted Evolutionary Algorithm Based on Decomposition for High-Dimensional Multi-Objective Problems," *IEEE Trans. Evol. Comput.*, pp. 1–15 (early access), 2022.

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

[5] A. Díaz-Manríquez, G. Toscano, and C. A. Coello Coello, "Comparison of metamodeling techniques in evolutionary algorithms," *Soft Computing*, vol. 21, no. 19, pp. 5647–5663, Oct. 2017.

[6] Gorissen, Dhaene, and De Turck, "Evolutionary model type selection for global surrogate modeling," *J. Mach. Learn. Res.*, 2009.

[7] M. Yu, X. Li, and J. Liang, "A dynamic surrogate-assisted evolutionary algorithm framework for expensive structural optimization," *Struct. Multidiscip. Optim.*, vol. 61, no. 2, pp. 711–729, Feb. 2020.

[8] L. Shi and K. Rasheed, "ASAGA: an adaptive surrogate-assisted genetic algorithm," in *Annu. Conf. Genet. Evol. Comput. (GECCO)*, ser. GECCO '08. New York, NY, USA: ACM, Jul. 2008, pp. 1049–1056.

[9] M. Yu, J. Liang, K. Zhao, and Z. Wu, "An aRBF surrogate-assisted neighborhood field optimizer for expensive problems," *Swarm Evol. Comput.*, p. 100972, Aug. 2021.

[10] S. Bagheri, W. Konen, and T. Bäck, "Online selection of surrogate models for constrained black-box optimization," in *IEEE Symp. Ser. Comput. Intell. (SSCI)*. ieeexplore.ieee.org, Dec. 2016, pp. 1–8.

[11] G. Li, Q. Zhang, J. Sun, and Z. Han, "Radial Basis Function Assisted Optimization Method with Batch Infill Sampling Criterion for Expensive Optimization," in *IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 1664–1671.

[12] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, "DACE: A MATLAB Kriging Toolbox," Informatics and Mathematical Modelling, DTU, Tech. Rep. IMM-REP-2002-12, Aug. 2002.

[13] R. Hooke and T. A. Jeeves, ""Direct Search" Solution of Numerical and Statistical Problems," *J. ACM*, vol. 8, no. 2, pp. 212–229, Apr. 1961.

[14] S. M. Elsayed, T. Ray, and R. A. Sarker, "A surrogate-assisted differential evolution algorithm with dynamic parameters selection for solving expensive optimization problems," in *IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1062–1068.

[15] N. H. Awad, M. Z. Ali, R. Mallipeddi, and P. N. Suganthan, "An improved differential evolution algorithm using efficient adapted surrogate model for numerical optimization," *Inf. Sci.*, vol. 451-452, pp. 326–347, Jul. 2018.

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

[17] 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.

[18] X. Ma, K. Zhang, L. Zhang, Y. Wang, H. Wang, J. Wang, and J. Yao, "A distributed surrogate system assisted differential evolutionary algorithm for computationally expensive history matching problems," *J. Pet. Sci. Eng.*, vol. 210, p. 110029, Mar. 2022.

[19] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Inf. Sci.*, vol. 454-455, pp. 59–72, Jul. 2018.

[20] 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.

[21] F. Yu, W. Gong, and H. Zhen, "A data-driven evolutionary algorithm with multi-evolutionary sampling strategy for expensive optimization," *Knowl. Based. Syst.*, vol. 242, p. 108436, Apr. 2022.

[22] X. Wang, G. G. Wang, B. Song, P. Wang, and Y. Wang, "A Novel Evolutionary Sampling Assisted Optimization Method for High-Dimensional Expensive Problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 815–827, Oct. 2019.

[23] X. Cai, L. Gao, X. Li, and H. Qiu, "Surrogate-guided differential evolution algorithm for high dimensional expensive problems," *Swarm Evol. Comput.*, vol. 48, pp. 288–311, Aug. 2019.

[24] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Comput. Intell. Lab., Zhengzhou Univ., China, and Nanyang Tech. Univ., Singapore, Tech. Rep., 2013.

[25] R. G. Regis, "Particle swarm with radial basis function surrogates for expensive black-box optimization," *J. Comput. Sci.*, vol. 5, no. 1, pp. 12–23, Jan. 2014.

[26] 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.

[27] H. Yu, Y. Tan, C. Sun, and J. Zeng, "A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization," *Knowl. Based. Syst.*, vol. 163, pp. 14–25, Jan. 2019.