Surrogate-Assisted Differential Evolution with Region Division for Expensive Optimization Problems with Discontinuous Responses

Yong Wang, *Senior Member, IEEE,* Jianqing Lin, Jiao Liu, Guangyong Sun, and Tong Pang
,

***Abstract*—A considerable number of surrogate-assisted evolutionary algorithms (SAEAs) have been developed to solve expensive optimization problems (EOPs) with continuous objective functions. However, in the real-world applications, we may face EOPs with discontinuous objective functions, which are also called EOPs with discontinuous responses (EOPDRs). Indeed, EOPDRs pose a great challenge to current SAEAs. In this paper, a surrogate-assisted differential evolution (DE) algorithm with region division is proposed, named ReDSADE. ReDSADE includes three main strategies: the region division strategy, the Kriging-based search, and the radial basis function (RBF)-based local search. In the region division strategy, we define a new distance measure, called the objective-decision distance. Based on this distance, the evaluated solutions are partitioned into several clusters, and several support vector machine (SVM) classifiers are trained to classify them. These SVM classifiers divide the decision space into several subregions, with the aim of making the objective function continuous in them. In the Kriging-based search, a Kriging model is established in each subregion and combined with DE to search for the optimal solution. In the RBF-based local search, DE is coupled with RBF to search around the best solution found so far, thus accelerating the convergence. By combining these three strategies, ReDSADE is able to solve EOPDRs with limited function evaluations. Three set of test problems and a real-world application are utilized to verify the effectiveness of ReDSADE. The results demonstrate that ReDSADE exhibits good convergence accuracy and convergence speed.**

***Index Terms*—Surrogate-assisted evolutionary algorithms, differential evolution, expensive optimization problems, discontinuous response, region division**

I. INTRODUCTION

MANY real-world optimization problems have black-box and time-consuming objective functions and/or constraints [1], [2], which are known as expensive optimization problems (EOPs). Although evolutionary algorithms (EAs) are good at solving black-box optimization problems, they usually require tens of thousands of function evaluations (FEs) to obtain the near-optimal solution [3]–[5]. To overcome this shortcoming, researchers have developed numerous new methods to exploit computationally cheap surrogate models to replace a part of time-consuming exact FEs in the EA community, which are the so-called surrogate-assisted EAs (SAEAs). During the past fifteen years, a considerable number of SAEAs have been proposed to solve different kinds of EOPs, e.g., single-objective EOPs [6]–[14], multi/many-objective EOPs [15]–[23], and combinatorial EOPs [24], [25].

Most of current SAEAs consider that the objective functions of EOPs are continuous. However, in the real world, an EOP may have a discontinuous objective function, which is also termed as an EOP with discontinuous response (EOPDR). As introduced in Section V, the lightweight and crashworthiness design of the front bumper in an automobile is a typical EOPDR. With the change of a design variable, the front bumper may exhibit different deformation modes, which causes that the objective function is discontinuous. The mathematical model of an EOPDR is expressed as follows:

$$\begin{aligned} \min : & \ f(\mathbf{x}) \\ \text{s.t.} & \ L_i \le x_i \le U_i, \ i = 1, \dots, D \\ & \ f(\mathbf{x}) \ \text{ is discontinuous} \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, \dots, x_D)$ is a D-dimensional decision vector, L_i and U_i are the lower and upper bounds of the ith decision variable, respectively, and $f(\mathbf{x})$ is the objective function. Different from common EOPs, $f(\mathbf{x})$ is discontinuous within the decision space of an EOPDR. In this paper, we concentrate mainly on the jump discontinuity since it is the most common situation in the real world [26].

Current SAEAs may not be suitable for EOPDRs. The reason is that most of the surrogate models employed by them, such as polynomial regression [27], support vector machine (SVM) [28], radial basis function (RBF) [14], artificial neural network [29], and Kriging [30], assume that the approximated objective functions are continuous [4], [5], [12]. As a result, the capability of these surrogate models to approximate objective functions with discontinuous responses is limited. Therefore, solving EOPDRs by using these surrogate models directly may mislead the optimization process. Researchers have made some attempts to address this issue. However, they mainly discuss how to establish surrogate models for

Manuscript received February 1, 2021; revised June 13, 2021 and August 13, 2021; accepted October 2, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61976225, and in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant 2018IRS06. *(Corresponding author: Jiao Liu.)*

Y. Wang and J. Liu are with the School of Automation, Central South University, Changsha 410083, China. (Email: ywang@csu.edu.cn; liu_jiao@csu.edu.cn)

J. Lin is with the school of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China. (Email: linjqcn@gmail.com)

G. Sun and T. Pang are with the State Key Laboratory of Advanced Design and Manufacture for Vehicle Body, Hunan University, Changsha 410082, China. (Email: sgy800@126.com; pangtong2011@163.com)

discontinuous objective functions, and rarely consider how to combine surrogate models with EAs to obtain a high-quality solution of an EOPDR by using limited FEs [31]–[33].

Based on the above considerations, a surrogate-assisted differential evolution (DE) algorithm with region division, called ReDSADE, is proposed in this paper. ReDSADE includes three main strategies: 1) the region division strategy, 2) the Kriging-based search, and 3) the RBF-based local search. The main contributions of this paper can be summarized as follows:

- In the region division strategy, a new distance measure is proposed. Based on this distance, the database, which includes the solutions evaluated by the original objective function, are partitioned into several clusters through the density-based spatial clustering of applications with noise (DBSCAN) [34]. Afterwards, a SVM classifier is constructed to distinguish one cluster from the other clusters in the decision space. As a result, the SVM classifiers divide the decision space into several subregions, with the aim of making the objective function continuous in them.
- The Kriging-based search is proposed to search for the optimal solution. At each generation, for each subregion, a Kriging model is established and a subpopulation is produced based on the corresponding cluster. Then, DE is used to optimize the expected improvement (EI) function constructed over each Kriging model. After several iterations, the solution with the best EI value is selected from all the subpopulations and evaluated by the original objective function.
- The RBF-based local search is designed to accelerate the convergence. At each generation, a RBF model is established based on the solutions close to the best solution found so far. Meanwhile, DE is used to optimize the RBF function. Finally, the solution with the best RBF function value is evaluated by the original objective function.

In the experiments, three sets of test problems are designed to systematically study the performance of ReDSADE. The performance of ReDSADE is better than that of three state-of-the-art algorithms. Moreover, ReDSADE is applied to solve a real-world EOPDR, i.e., the lightweight and crashworthiness design of the front bumper in an automobile. Compared with EGO [35], ReDSADE is able to obtain a better design result.

The rest of this paper is organized as follows. Section II introduces DE, SVM, Kriging, RBF, and DBSCAN. The proposed algorithm, i.e., ReDSADE, is elaborated in Section III. The experimental studies are carried out in Section IV. In Section V, ReDSADE is applied to a real-world EOPDR. Finally, Section VI concludes this paper.

II. Related Techniques

A. DE

DE is a popular population-based optimizer [36]–[38]. It consists of the following four operators: initialization, mutation, crossover, and selection.

First, in the initialization operator, ps solutions are randomly produced from the decision space:

$$\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,D}),\ i = 1, \ldots, ps \tag{2}$$

where $\mathbf{x}_i$ is the ith solution.

Then, a mutant vector $\mathbf{v}_i = (v_{i,1}, \ldots, v_{i,D})$ is created for $\mathbf{x}_i$ by a mutation operator. The most commonly used mutation operator is DE/rand/1, which is formulated as:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{3}$$

where r_1, r_2, and r_3 are three different integers randomly selected from $\{1, \ldots, ps\} \setminus \{i\}$, and F is the scaling factor.

Afterwards, by using the binomial crossover, a trial vector $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,D})$ is generated based on $\mathbf{x}_i$ and $\mathbf{v}_i$:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand_j < CR \text{ or } j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{4}$$

where $i = 1, \ldots, ps$, $j = 1, \ldots, D$, $CR \in [0, 1]$ is the crossover control parameter, $rand_j$ is a uniformly distributed random number between 0 and 1, and j_{rand} is an integer randomly selected from $\{1, \ldots, D\}$.

Finally, the selection operator is implemented, in which the better one between $\mathbf{x}_i$ and $\mathbf{u}_i$ is selected into the next generation:

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) \le f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases} \tag{5}$$

B. SVM

SVM aims to find the optimal separating hyperplane to distinguish solutions in two different classes [39], [40]. In principle, SVM maximizes the margin around the separating hyperplane, and the final classification result is determined by a subset of training solutions called support vectors. When facing linear inseparable solutions, a SVM classifier employs a kernel function to convert the input solutions to a linear separable form. Based on a database $\mathcal{D}_1 = \{(\mathbf{x}_i, l_i) | i = 1, \ldots, N\}$ ($l_i \in \{+1, -1\}$ represents the label of a solution $\mathbf{x}_i$), SVM classifies linear inseparable solutions as follows:

$$\hat{f}_{svm}(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i l_i \phi(\mathbf{x}_i, \mathbf{x}) + \sum_{i=1}^{N} \alpha_i l_i\right) \tag{6}$$

where $\hat{f}_{svm}(\mathbf{x})$ is the predicted label, $\phi(\cdot, \cdot)$ is the kernel function[1], and α_i ($i \in \{1, \ldots, N\}$) is the parameter that can be obtained by solving the following optimization problem:

$$\begin{aligned} \max:\ & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j l_i l_j \phi(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t. } & \alpha_i \ge 0, i = 1, \ldots, N \\ & \sum_{i=1}^{N} \alpha_i l_i = 0 \end{aligned} \tag{7}$$

(7) can be solved via the sequence minimum optimization [41].

[1] In this paper, the Gaussian kernel function is adopted.

C. Kriging

The Kriging model is a surrogate model which can not only predict the objective function value of a solution, but also provide the uncertainty of the predicted value. It assumes that the objective function has the following form:

$$f(\mathbf{x}) = \mu + \mathcal{N}(0, s^2) \tag{8}$$

where μ is the mean of the Gaussian process, and $\mathcal{N}(0, s^2)$ is a Gaussian distribution with mean zero and variance s^2.

For two solutions $\mathbf{x}_i$ and $\mathbf{x}_j$, the correlation between them is defined as:

$$Corr(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{k=1}^{D} \theta_k |x_{i,k} - x_{j,k}|^2\right) \tag{9}$$

where θ_k ($0 \leq \theta_k < \infty$ and $k \in \{1, ..., D\}$) is the weight coefficient. Based on a database $\mathcal{D}_2 = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, N\}$ (y_i is a real-valued number and represents the objective function value of a solution $\mathbf{x}_i$), values of θ_k can be obtained by the maximum likelihood estimation:

$$\max_{\boldsymbol{\theta}} \left(-\frac{\ln \hat{\sigma}^2 + \ln |\mathbf{A}|}{2}\right) \tag{10}$$

where

$$\hat{\mu} = \frac{\mathbf{1}^T \mathbf{A}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{A}^{-1} \mathbf{1}} \tag{11}$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{A}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{N} \tag{12}$$

Note that $\mathbf{A}$ is a $N \times N$ matrix with entry $A_{ij} = Corr(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{y} = (y_1, \ldots, y_N)^T$, and $\mathbf{1}$ is a N-dimensional vector of ones. In this paper, DE is used to optimize (10). DE's two parameters, i.e., F and CR, are set to 0.5 and 0.9, respectively, and the maximum number of FEs (denoted as L) is set to 500.

Subsequently, the predicted value and uncertainty provided by a Kriging model are calculated as follows [35]:

$$\hat{f}_{kri}(\mathbf{x}) = \hat{\mu} + \mathbf{a}^T \mathbf{A}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{13}$$

$$\hat{s}^2_{kri}(\mathbf{x}) = \hat{\sigma}^2 \left(1 - \mathbf{a}^T \mathbf{A}^{-1} \mathbf{a} + \frac{(1 - \mathbf{1}^T \mathbf{A}^{-1} \mathbf{a})^2}{\mathbf{1}^T \mathbf{A}^{-1} \mathbf{1}}\right) \tag{14}$$

where $\mathbf{a}$ is a N-dimensional vector with entry $a_i = Corr(\mathbf{x}, \mathbf{x}_i)$.

Based on (13) and (14), the EI value of a solution $\mathbf{x}$ can be calculated as [19], [20], [42]:

$$\begin{aligned} EI(\mathbf{x}) = (f_{min} - \hat{f}_{kri}(\mathbf{x}))\Phi\left(\frac{f_{min} - \hat{f}_{kri}(\mathbf{x})}{\hat{s}_{kri}(\mathbf{x})}\right) \\ + \hat{s}_{kri}(\mathbf{x})\phi\left(\frac{f_{min} - \hat{f}_{kri}(\mathbf{x})}{\hat{s}_{kri}(\mathbf{x})}\right) \end{aligned} \tag{15}$$

where f_{min} is the objective function value of the best solution in $\mathcal{D}_2$, and $\Phi(\cdot)$ and $\phi(\cdot)$ are the Gaussian cumulative distribution function and probability density function, respectively.

D. RBF

RBF is a simple and widely used surrogate model. In essence, it is composed of multiple kernel functions. It can also be treated as a single-layer neural network. On the basis of a database $\mathcal{D}_3 = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, N\}$ (y_i is a real-valued number and represents the objective function value of a solution $\mathbf{x}_i$), it approximates a continuous function as follows:

$$\hat{f}_{rbf}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\mathbf{x}_i, \mathbf{x}) \tag{16}$$

where w_i and $\phi(\cdot, \cdot)$ are the weight coefficient and kernel function, respectively. The weight vector $\mathbf{w} = (w_1, ..., w_N)$ is calculated as follows:

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y} \tag{17}$$

where $\mathbf{y} = (y_1, \ldots, y_N)^T$ and Φ is the matrix computed as follows:

$$\Phi = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N, \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{18}$$

E. DBSCAN

DBSCAN is a density-based clustering method, the idea of which is to form clusters that are dense enough and partition different clusters by the sparse regions [34]. DBSCAN is implemented based on two kinds of parameters: ϵ and $MinPts$. Actually, different clusters can have different ϵ and $MinPts$ values, such as in [43]. In DBSCAN, the following four concepts are introduced:

Definition 1: *ϵ-neighborhood*: The ϵ-neighborhood of a solution $\mathbf{x}$ is defined as: $N_\epsilon(\mathbf{x}) = \{\mathbf{y} | dist(\mathbf{x}, \mathbf{y}) \leq \epsilon\}$, where $dist(\cdot, \cdot)$ is the distance function.

Definition 2: *Directly density-reachable*: A solution $\mathbf{x}$ is directly density-reachable from another solution $\mathbf{y}$, if 1) $\mathbf{x} \in N_\epsilon(\mathbf{y})$ and 2) $|N_\epsilon(\mathbf{y})| \geq MinPts$, where $MinPts$ is a parameter and $|N_\epsilon(\mathbf{y})|$ is the size of $N_\epsilon(\mathbf{y})$.

Definition 3: *Density-reachable*: A solution $\mathbf{x}$ is density-reachable from another solution $\mathbf{y}$, if there is a chain of solutions (denoted as $\mathbf{x}_1, \ldots, \mathbf{x}_z$), $\mathbf{x} = \mathbf{x}_1$, $\mathbf{y} = \mathbf{x}_z$, and $\mathbf{x}_i$ is directly density-reachable from $\mathbf{x}_{i+1}$.

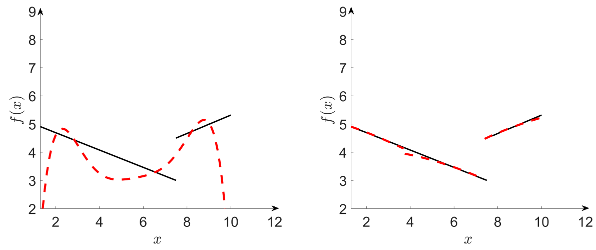
Definition 4: *Density-connected*: A solution $\mathbf{x}$ is density-connected to another solution $\mathbf{y}$, if there is a solution $\mathbf{z}$ such that both $\mathbf{x}$ and $\mathbf{y}$ are density-reachable from $\mathbf{z}$.

Based on the above concepts, DBSCAN defines the cluster and noise as follows:

Definition 5: *Cluster*: Let $\mathcal{D}_4 = \{\mathbf{x}_i | i = 1, \ldots, N\}$ be a database. A cluster C is defined as: 1) $\forall$ $\mathbf{x}$, $\mathbf{y}$: if $\mathbf{x} \in C$ and $\mathbf{y}$ is density-reachable from $\mathbf{x}$, then $\mathbf{y} \in C$, and 2) $\forall$ $\mathbf{x}$, $\mathbf{y}$: if $\mathbf{x} \in C$ and $\mathbf{y}$ is density-connected to $\mathbf{x}$, then $\mathbf{y} \in C$.

Definition 6: *Noise*: Let $C_1, C_2, \ldots$ be the clusters of $\mathcal{D}_4$. Noise is defined as: $noise = \{\mathbf{x} \in \mathcal{D}_4 | \forall i : \mathbf{x} \notin C_i\}$.

According to the above definitions, DBSCAN traverses all the solutions in $\mathcal{D}_4$, and partitions them into different clusters. In this paper, we utilize the self-adaptive strategy in [43] to adjust the parameters of DBSCAN. In this strategy, multiple



(a) (b)

Fig. 1. Approximation of the objective function of an EOPDR by RBF. (a) Approximation of the objective function of an EOPDR in the whole decision space. (b) Approximation of the objective function of an EOPDR in the two subregions, respectively.

Algorithm 1 *ReDSADE*

Input: H and K

Output: the best solution in $\mathcal{D}$

1: Initialize $\mathcal{D}$, which contains N solutions produced from the decision space by Latin hypercube design and their original objective function values;
2: t = N; // t is the number of FEs
3: **while** $t < MaxFEs$ **do**
4: $\quad$ [$\mathcal{C}$, $\mathcal{S}$] $\leftarrow$ *ReD*($\mathcal{D}$);
5: $\quad$ $\mathcal{D} \leftarrow$ *Kriging-S*($\mathcal{D}$, $\mathcal{C}$, $\mathcal{S}$, H);
6: $\quad$ $t = t + 1$;
7: $\quad$ $\mathcal{D} \leftarrow$ *RBF-LS*($\mathcal{D}$, K);
8: $\quad$ $t = t + 1$;
9: **end while**

---

parameters, i.e., ϵ_j and $MinPts_j$ ($j = 1, 2, \ldots$), are set as follows:

- The values of ϵ_j ($j = 1, 2, \ldots$) are determined through the following steps. Firstly, for each solution in $\mathcal{D}_4$, the average distance from it to its k nearest neighbors (denoted as KD) is calculated. Afterwards, the KD values of all solutions are sorted in the ascending order. Next, the slopes of the first pair of neighbors and the second pair of neighbors are calculated, and if there is a large change between these two neighboring slopes[2], ϵ_1 is equal to the second KD value. Subsequently, the same process is implemented for the remaining pairs of neighbors. As a result, $\{\epsilon_j | j = 1, 2, \ldots\}$ can be obtained.
- $MinPts_j$ is calculated as follows:

$$MinPts_j = \frac{\sum_{n=1}^{N} P_n I_n}{\sum_{n=1}^{N} I_n} \tag{19}$$

$$I_n = \begin{cases} 1, & \text{if } \epsilon_{j-1} \leq KD_n \leq \epsilon_j \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

where P_n is the number of solutions in the ϵ_j-neighborhood of the nth solution, and KD_n is the KD value of the nth solution.

After obtaining the parameters of DBSCAN, we firstly cluster the solutions in $\mathcal{D}_4$ by using ϵ_1 and $MinPts_1$. Afterwards, the remaining solutions in $\mathcal{D}_4$ are clustered by using $\epsilon_2, \epsilon_3, \ldots$ and $MinPts_2, MinPts_3, \ldots$ sequentially.

III. Proposed Method

A. Motivation

In general, even though the objective function of an EOPDR is discontinuous within the whole decision space, we could find several subregions and make the objective function continuous in them. Compared with approximating the objective function in the whole decision space, approximating it in these subregions separately may provide more accurate predicted values. An example in Fig. 1 is used to illustrate this phenomenon. In Fig. 1, we employ RBF to approximate the objective function of an EOPDR. In Fig. 1(a), RBF is directly used to approximate it in the whole decision space. It can be observed that, under this condition, RBF cannot reflect the landscape of the objective function well. In contrast, as shown in Fig. 1(b), when establishing RBF in two subregions (i.e., $[0, 7.8]$ and $(7.8, 10]$) respectively, the landscape of the objective function can be approximated well. Therefore, if we could divide the decision space into several subregions reasonably and establish surrogate models in these subregions respectively, the prediction capability could be greatly improved. Motivated by this, ReDSADE is proposed in this paper.

B. General Framework of ReDSADE

The symbols used in ReDSADE are introduced as follows:

- $\mathcal{D}$: the database containing the evaluated solutions and their original objective function values. Inspired by [20], to save the time consumed by establishing surrogate models, $\mathcal{D}$ contains at most 300 solutions. Once the size of $\mathcal{D}$ is bigger than 300, the worst solutions in terms of the original objective function value in $\mathcal{D}$ are deleted;
- $\mathcal{C}$: the set containing all the clusters, i.e., $\mathcal{C} = \{C_1, \ldots, C_c\}$, where c is the number of clusters;
- $\mathcal{S}$: the set containing all the SVM classifiers, i.e., $\mathcal{S} = \{SVM_1, \ldots, SVM_c\}$;
- t: the number of FEs.

The parameters used in ReDSADE are summarized as follows:

- H: the maximum population size in the Kriging-based search;
- K: the population size in the RBF-assisted local search.

The framework of ReDSADE is given in **Algorithm 1**. At first, N solutions are produced from the decision space by Latin hypercube design: $\mathbf{x}_1, \ldots, \mathbf{x}_N$. These solutions are evaluated by the original objective function. Subsequently, these solutions and their original objective function values are saved in $\mathcal{D}$. During the evolution, three strategies are implemented iteratively until the maximum number of FEs (denoted as $MaxFEs$) is reached: 1) the region division strategy (line 4), 2) the Kriging-based search (line 5), and 3) the RBF-based local search (line 7).

[2]When the change between these two neighboring slopes is greater than 0.2, it is considered that there is a large change.

Algorithm 2 *ReD*

Input: $\mathcal{D}$
Output: $\mathcal{C}$ and $\mathcal{S}$

1: Normalize all the solutions in $\mathcal{D}$ according to (21) and (22);
2: Partition the solutions in $\mathcal{D}$ into c clusters based on ODD and DBSCAN: $\mathcal{C} = \{C_1, \ldots, C_c\}$;
3: **for** $i = 1, \ldots, c$ **do**
4: Set the labels of the solutions in C_i to $+1$;
5: Set the labels of the solutions in $\{C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_c\}$ to -1;
6: Train a SVM classifier that can distinguish the solutions with the label of $+1$ from the solutions with the label of -1, and record this classifier as SVM_i;
7: **end for**
8: Reserve all the SVM classifiers into $\mathcal{S}$: $\mathcal{S} = \{SVM_1, \ldots, SVM_c\}$.

In the region division strategy, the solutions in $\mathcal{D}$ are firstly partitioned into c clusters (i.e., $\mathcal{C} = \{C_1, \ldots, C_c\}$), and then c SVM classifiers (i.e., $\mathcal{S} = \{SVM_1, \ldots, SVM_c\}$) are trained according to these c clusters. Each of the SVM classifiers can determine a subregion. The Kriging-based search aims to search for the optimal solution in the subregions determined by the SVM classifiers. In the Kriging-based search, DE is employed as the search engine, and a parameter (i.e., H) determines how DE is implemented. The purpose of the RBF-based local search is to accelerate the convergence. In the RBF-based local search, we also employ DE as the search engine, and a parameter (i.e., K) determines the implementation of DE. Note that both the Kriging model and the RBF model are established based on the solutions in $\mathcal{D}$. Once $\mathcal{D}$ has been updated, both of them are also reestablished. At each generation, both the Kriging-based search and the RBF-based local search consume one FE since only one solution is evaluated by the original objective function. Next, we will introduce these three strategies in detail.

C. Region Division Strategy

The process of the region division strategy is described in **Algorithm 2**. Firstly, both the decision variables and the objective function values of the solutions in $\mathcal{D}$ are normalized as follows:

$$x'_{i,j} = \frac{x_{i,j} - L_j}{U_j - L_j} \tag{21}$$

$$f'_i = \frac{f_i - f_{min}}{f_{max} - f_{min}} \tag{22}$$

where $x_{i,j}$ and $x'_{i,j}$ are the jth decision variable of the ith solution before and after normalization, respectively, f_i and f'_i are the objective function values of the ith solution before and after normalization, respectively, and f_{min} and f_{max} are the minimum and maximum objective function values in $\mathcal{D}$, respectively. Then, a new distance measure, called the objective-decision distance (denoted as ODD), is proposed.

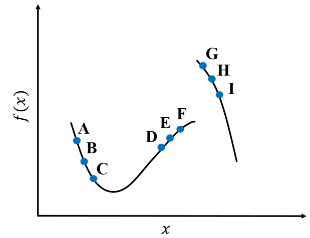
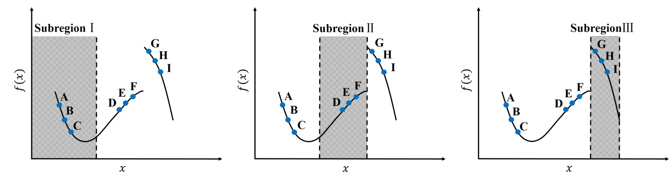


Fig. 2. An example to illustrate the working principle of the region division strategy. $f(x)$ is an one-dimensional discontinuous objective function and A-I are nine solutions.



(a) (b) (c)

Fig. 3. Subregions determined by the three SVM classifiers. (a) The subregion determined by SVM_1. (b) The subregion determined by SVM_2. (c) The subregion determined by SVM_3.

ODD between any two normalized solutions $(\mathbf{x}'_{r1}, f'_{r1})$ and $(\mathbf{x}'_{r2}, f'_{r2})$ is calculated as follows:

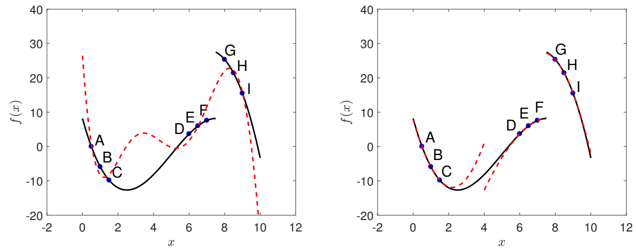
$$\begin{aligned} &ODD((\mathbf{x}'_{r1}, f'_{r1}), (\mathbf{x}'_{r2}, f'_{r2})) \\ &\qquad = \sqrt{(||\mathbf{x}'_{r1} - \mathbf{x}'_{r2}||)^2 + (f'_{r1} - f'_{r2})^2} \end{aligned} \tag{23}$$

where $||\cdot||$ is the vector norm. It can be observed from (23) that ODD considers the difference of two solutions in both the decision space and the objective space. The characteristics of ODD are summarized as follows:

- If two solutions are close to each other in both the decision space and the objective space, they tend to have a small ODD value.
- For a discontinuous objective function, it is possible that two solutions are close to each other in the decision space, but their objective function values are quite different. Under this condition, they may have a big ODD value. Thus, ODD has the capability to distinguish them.

Based on ODD, DBSCAN is used to partition the solutions in $\mathcal{D}$ into c clusters: $C_1, \ldots, C_c$. Afterwards, c SVM classifiers are trained (line 3 to line 7). Note that each SVM classifier can determine a subregion.

An example is designed to illustrate the working principle of the region division strategy. As shown in Fig. 2, $f(x)$ is an one-dimensional discontinuous objective function. Suppose that $\mathcal{D}$ contains nine solutions: A, B, C, D, E, F, G, H, and I. From Fig. 2, A, B, and C are close to each other in both the decision space and the objective space. Therefore, they form a high-density area. Despite D, E, F, G, H, and I are close to each other in the decision space, the objective function values of {D, E, F} and {G, H, I} are quite different. Therefore, these six solutions form two other high-density areas. As a result, based on ODD, DBSCAN partitions these nine solutions into three



(a) (b)

Fig. 4. Approximation of $f(x)$ by RBF. (a) Approximation of $f(x)$ in the whole decision space. (b) Approximation of $f(x)$ in the three subregions determined by the three SVM classifiers respectively.

Algorithm 3 *Kriging-S*

Input: $\mathcal{D}$, $\mathcal{C}$, $\mathcal{S}$, and H

Output: $\mathcal{D}$

1: **for** i = $1 : c$ **do**
2: Establish a Kriging model based on the solutions in C_i;
3: Select the best $m = \min(H, |C_i|)$ solutions from C_i to form a subpopulation: $SP_i = \{\mathbf{sx}_{i,1}, \ldots, \mathbf{sx}_{i,m}\}$;
4: **for** $iter = 1 : max\_iter$ **do**
5: **for** $j = 1 : m$ **do**
6: Generate an offspring solution $\mathbf{gx}_{i,j}$ for $\mathbf{sx}_{i,j}$ via the mutation and crossover operators of DE;
7: **while** $\mathbf{gx}_{i,j}$ is not in the subregion determined by SVM_i **do**
8: Regenerate an offspring solution $\mathbf{gx}_{i,j}$;
9: **end while**
10: Evaluate the EI function values of $\mathbf{sx}_{i,j}$ and $\mathbf{gx}_{i,j}$ based on (15);
11: If $EI(\mathbf{gx}_{i,j}) > EI(\mathbf{sx}_{i,j})$, replace $\mathbf{sx}_{i,j}$ with $\mathbf{gx}_{i,j}$;
12: **end for**
13: **end for**
14: **end for**
15: Select the solution with the best EI value from $\{SP_1, \ldots, SP_c\}$ and record it as $\mathbf{x}_s$;
16: Evaluate $\mathbf{x}_s$ by the original objective function, and reserve $\mathbf{x}_s$ and its original objective function value into $\mathcal{D}$.

Algorithm 4 *RBF-LS*

Input: $\mathcal{D}$

Output: $\mathcal{D}$

Select K solutions closest to $\mathbf{x}_{best}$ in $\mathcal{D}$ based on ODD;
Establish a RBF model based on these K solutions;
Employ these K solutions as the initial population, and optimize the RBF function in (16) by using DE;
Record the best solution obtained by DE as $\mathbf{x}_{ls}$;
Evaluate $\mathbf{x}_{ls}$ by the original objective function, and reserve $\mathbf{x}_{ls}$ and its original objective function value into $\mathcal{D}$.

---

clusters: cluster I (A, B, and C), cluster II (D, E, and F), and cluster III (G, H, and I). Subsequently, three SVM classifiers (i.e., SVM_1, SVM_2, and SVM_3) are trained according to these three clusters, respectively. Among them, SVM_1 can distinguish cluster I from the others, SVM_2 can distinguish cluster II from the others, and SVM_3 can distinguish cluster III from the others. These three classifiers determine three subregions (i.e., subregion I, subregion II, and subregion III), as shown in Fig. 3. Note that, $f(x)$ is continuous within these three subregions. Therefore, approximating $f(x)$ in them separately is more accurate than approximating $f(x)$ in the whole decision space, as shown in Fig. 4.

D. Kriging-Based Search

Algorithm 3 describes the process of the Kriging-based search. In this strategy, for the ith ($i \in \{1, \ldots, c\}$) subregion determined by the ith SVM classifier, a Kriging model is established based on the solutions in C_i (line 2). Subsequently, the best m solutions are selected from C_i: $\mathbf{sx}_{i,1}, \ldots, \mathbf{sx}_{i,m}$, where $m = \min(H, |C_i|)$ and $|C_i|$ is the size of C_i. These solutions form a subpopulation: $SP_i = \{\mathbf{sx}_{i,1}, \ldots, \mathbf{sx}_{i,m}\}$ (line 3). Afterwards, for the ith subregion, based on SP_i, DE is used to optimize the EI function constructed over the ith Kriging model in (15) (line 4 to line 13). After several iterations, we select the solution with the best EI value (denoted as $\mathbf{x}_s$) from all the subpopulations (line 15). Then, $\mathbf{x}_s$ is evaluated by the original objective function, and $\mathbf{x}_s$ and its objective function value are reserved into $\mathcal{D}$ (line 16).

E. RBF-Based Local Search

The RBF-based local search aims to find a more promising solution around the current best solution in $\mathcal{D}$ (denoted as $\mathbf{x}_{best}$), thus accelerating the convergence. The process of the RBF-based local search is shown in **Algorithm 4**. Firstly, based on ODD, K solutions closest to $\mathbf{x}_{best}$ are selected from $\mathcal{D}$. Then, by employing these K solutions as the initial population, a RBF model is established and DE is used to optimize the RBF function in (16). As a result, the solution with the best RBF function value is obtained, denoted as $\mathbf{x}_{ls}$. Finally, $\mathbf{x}_{ls}$ is evaluated by the original objective function, and $\mathbf{x}_{ls}$ and its original objective function value are reserved into $\mathcal{D}$.

F. Computational Complexity of ReDSADE

In ReDSADE, the region division strategy adopts DBSCAN to cluster the solutions in $\mathcal{D}$ and trains c SVM classifiers. Since the computational complexity of both DBSCAN and SVM is $O(DN^2)$ [44], [45], the computational complexity of the region division strategy is $O((c+1) \cdot DN^2)$. In the Kriging-based search, c Kriging models are established to approximate the objective function. Assume that N_i ($i \in \{1, \ldots, c\}$) solutions are used to establish the ith Kriging model, the computational complexity of establishing all the c Kriging models is $O(\sum_{i=1}^{c} LDN_i^3) \leq O(LD(\sum_{i=1}^{c} N_i)^3) = O(LDN^3)$ [46]. Moreover, the computational complexity of DE used in the Kriging-based search is negligible. Overall, the computational complex of the Kriging-based search is $O(LDN^3)$. Similarly, since the computational complexity of the RBF model is $O(DK^3)$, the computational complexity of the RBF-based local search is also $O(DK^3)$ [46]. As a result, the computational complexity of ReDSADE is $O(LDN^3 + DK^3)$.

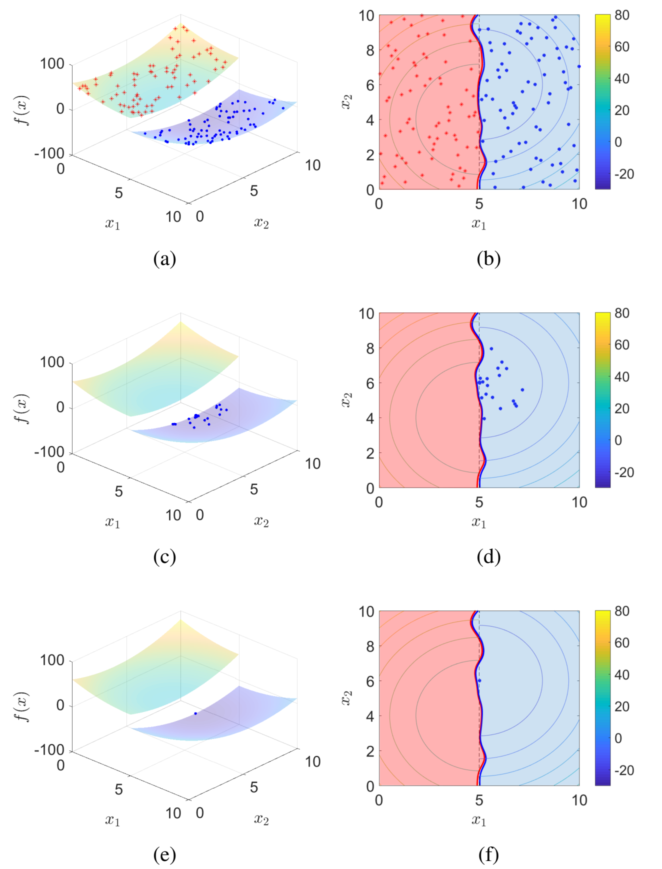


Fig. 5. Evolution of ReDSADE over a typical run on the constructed artificial test problem. (a) The 1st generation in a 3D view. (b) The 1st generation in the decision space. (c) The 120th generation in a 3D view. (d) The 120th generation in the decision space. (e) The 200th generation in a 3D view. (f) The 200th generation in the decision space.

IV. Experimental Studies

A. Proof-of-Principle Results

An artificial test problem[3] is constructed to explain the principle of ReDSADE:

$$\min : f(x) = \begin{cases} (x_1 - 5)^2 + (x_2 - 4)^2 + 20, & x_1 < 5 \\ (x_1 - 5)^2 + (x_2 - 6)^2 - 30, & x_1 \geq 5 \end{cases} \quad (24)$$
$$x_1, x_2 \in [0, 10]$$

The objective function of this artificial test problem is discontinuous at $x_1 = 5$. Its optimal solution and optimal objective function value are (5, 6) and -30, respectively.

Fig. 5 provides a typical run derived from ReDSADE on solving this artificial test problem. At first, ReDSADE generates 150 solutions in the whole decision space. Then, DBSCAN partitions these solutions into two clusters, as shown in red '$*$' and blue '$\cdot$' in Fig. 5(a) and Fig. 5(b), respectively. Based on these two clusters, two SVM classifiers (i.e., the red line and the blue line in Fig. 5(b) are trained, and the whole decision space is divided into two subregions (i.e., the red subregion and the blue subregion in Fig. 5(b). Subsequently, two Kriging models are trained in these two subregions, respectively. Afterwards, DE is utilized to optimize the EI functions in these two subregions. Although a solution has a high variance could also have a high EI value, during the evolution, the population will be gradually guided toward the subregion surrounding the optimal solution by optimizing the EI functions. As a result, the search will be gradually carried out on the blue subregion, since the optimal solution is located in it. To better illustrate this phenomenon, in Fig. 5(c), Fig. 5(d), Fig. 5(e), and Fig. 5(f), we exhibit the best 150 solutions in terms of the original objective function value in $\mathcal{D}$ at the 120th generation and the 200th generation, respectively. As shown in Fig. 5(c) and Fig. 5(d), ReDSADE focuses on finding the optimal solution in the blue subregion. Finally, at the 200th generation, ReDSADE converges to the optimal solution, as shown in Fig. 5(e) and Fig. 5(f).

B. Test Problems and Parameter Settings

In this paper, three sets of test problems were adopted to test the performance of ReDSADE. The first set consists of ten test problems (denoted as S1-F1–S1-F10). They were constructed based on the simple test problems collected in IEEE CEC2005 [47]. The second set also consists of ten test problems (denoted as S2-F1–S2-F10). They were constructed based on the complex test problems collected in IEEE CEC2005. Compared with the first set of test problems, the test problems in the second set have smaller gap sizes in the objective space, and the discontinuity of them depends on more decision variables. All the test problems in the first and second sets have discontinuous objective functions. The third set is the BBOB test problems developed in [48]. It consists of 24 test problems. The details of these three sets of test problems are summarized in the supplementary file.

In the experimental study, the size of the initial $\mathcal{D}$ (i.e., N) was set to 150 and $MaxFEs$ was set to 600. For DE, F was set to 0.5 and CR was set to 0.9. In the Kriging-based search, H was set to 50 and max_iter was set to 20. In the RBF-based local search, K was set to 50.

To evaluate the performance of different algorithms, the following three statistical values were calculated:

- Mean Function Error Value (*MFEV*): The mean of the function error value $|f(\mathbf{x}_{best}) - f(\mathbf{x}^*)|$ provided by an algorithm over 20 independent runs, where $\mathbf{x}_{best}$ is the best solution provided by the algorithm and $\mathbf{x}^*$ is the optimal solution. When the optimal solution of a test problem is known, this value was calculated. Note that *MFEV* is used for the first and third sets of test problems.
- Mean Objective Function Value (*MOFV*): The mean of the objective function value provided by an algorithm over 20 independent runs. When the optimal solution of a test problem is unknown, this value was calculated. Note that *MOFV* is used for the second set of test problems.
- Standard Deviation (*Std Dev*): The standard deviation of *MFEV*/*MOFV* values provided by an algorithm over 20 independent runs.

[3]In the community of evolutionary computation, to test the performance of an algorithm designed for EOPs, a common way is to use some inexpensive test problems and limit the number of FEs while ignoring the time cost. By this way, the performance of an algorithm designed for EOPs can be evaluated to a certain degree.

TABLE I

RESULTS OF EGO, CAL-SAPSO, GLoSADE, AND ReDSADE ON THE FIRST SET OF TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN ReDSADE AND EACH OF EGO, CAL-SAPSO, AND GLoSADE.

Problem	EGO *MOFV* ± *Std Dev*		CAL-SAPSO *MOFV* ± *Std Dev*		GLoSADE *MOFV* ± *Std Dev*		ReDSADE *MOFV* ± *Std Dev*
S1-F1	8.96E+02±4.47E+02	+	1.63E+03±8.34E+02	+	5.71E+01±1.20E+01	+	**1.08E-04±2.05E-04**
S1-F2	2.14E+03±8.28E+02	+	7.95E+03±3.06E+03	+	8.99E+02±7.09E+02	+	**1.09E+00±7.44E-01**
S1-F3	4.60E+01±2.47E+01	+	7.34E+01±5.14E+01	+	3.18E+00±3.28E+00	+	**3.01E-09±3.72E-09**
S1-F4	1.05E+01±7.96E+00	+	6.23E+01±4.74E+01	+	5.59E-01±4.05E-01	+	**3.72E-10±7.44E-10**
S1-F5	4.31E+02±3.27E+02	+	9.70E+02±2.24E+02	+	9.32E+01±4.40E+01	+	**1.81E-04±3.60E-04**
S1-F6	9.58E-03±1.00E-02	+	2.60E-02±1.19E-02	+	2.33E-03±2.57E-03	+	**7.87E-09±1.16E-08**
S1-F7	1.96E-02±9.41E-03	+	7.44E-02±4.92E-02	+	9.27E-03±9.93E-03	+	**1.52E-04±1.11E-04**
S1-F8	2.58E-02±7.57E-03	+	6.43E-02±8.79E-03	+	7.48E-03±5.86E-03	+	**1.21E-04±1.25E-04**
S1-F9	1.66E+02±7.96E+01	+	7.98E+02±2.79E+02	+	9.73E+00±1.36E+01	+	**3.16E-07±5.29E-07**
S1-F10	5.88E+02±2.83E+02	+	9.49E+02±1.66E+02	+	6.31E+01±3.19E+01	+	**2.00E-02±2.96E-02**
$+/-/\approx$	10/0/0		10/0/0		10/0/0		

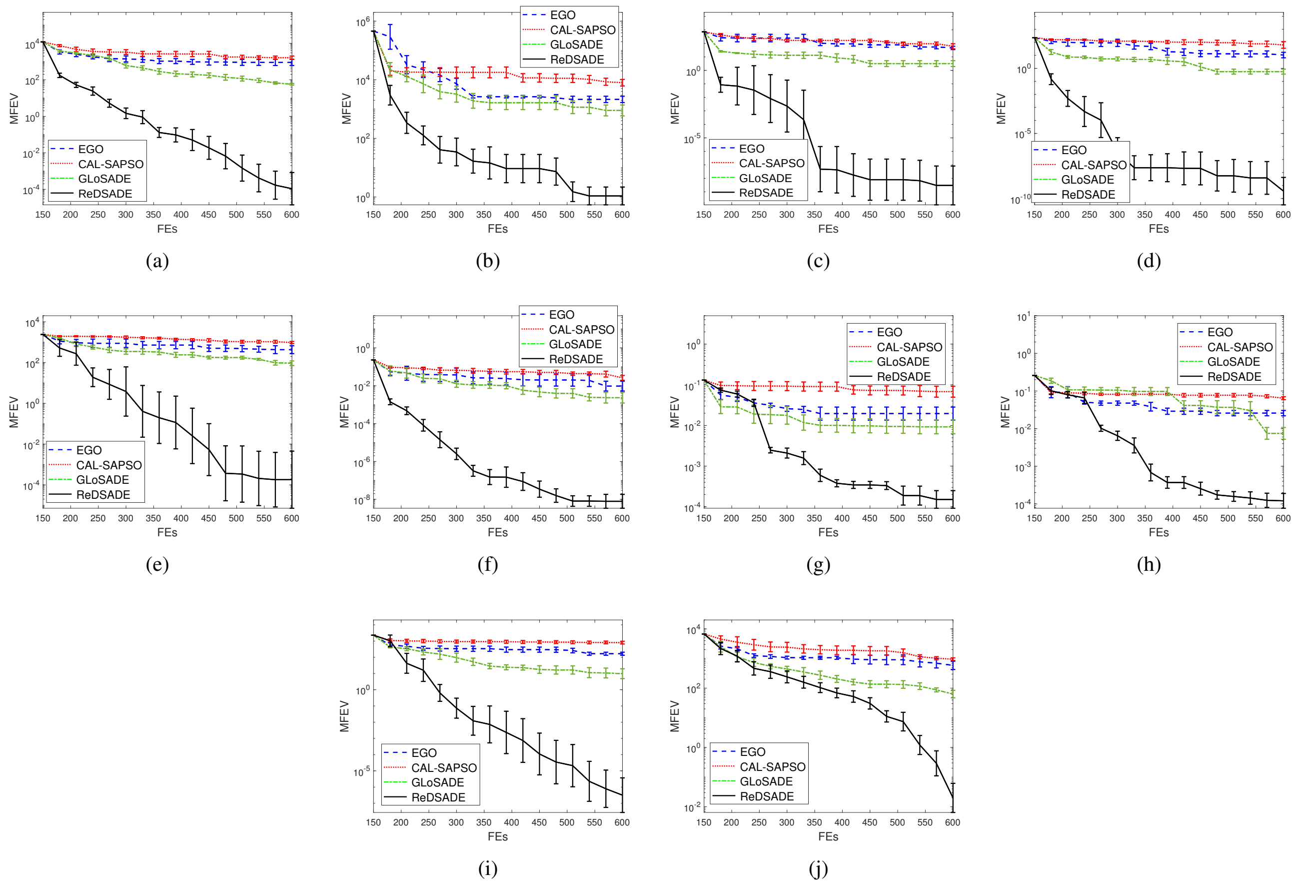


Fig. 6. Evolution of *MFEV* provided by the four compared algorithms on the first set of test problems over 20 independent runs. (a) S1-F1. (b) S1-F2. (c) S1-F3. (d) S1-F4. (e) S1-F5. (f) S1-F6. (g) S1-F7. (h) S1-F8. (i) S1-F9. (j) S1-F10.

In the experimental studies, the Wilcoxon's rank-sum test at a 0.05 significance level was implemented between ReDSADE and each of its competitors to test the statistical significance. In the following tables, "+", "−", and "≈" denote that ReDSADE performs better than, worse than, and similar to its competitor, respectively.

C. Comparison with Three State-of-the-Art SAEAs on the First Set of Test Problems

ReDSADE was compared with three state-of-the-art SAEAs: EGO [35], CAL-SAPSO [5], and GLoSADE [11]. EGO is a classical algorithm which has been applied to solve many EOPs in the real world, CAL-SAPSO is a SAEA which shows excellent performance on solving EOPs with continuous responses, and GLoSADE is a SAEA which employs both global and local surrogate models to deal with EOPs.

The results of EGO, CAL-SAPSO, GLoSADE, and ReDSADE on the first set of test problems are summarized in Table I. It can be observed that ReDSADE achieves significantly better results than its three competitors. Specifically, the *MFEV* values provided by ReDSADE are at least two orders of magnitude smaller than those produced by EGO and

CAL-SAPSO, and at least one order of magnitude smaller than those resulting from GLoSADE. According to the Wilcoxon's rank-sum test, ReDSADE surpasses EGO, CAL-SAPSO, and GLoSADE on all the ten test problems. In Fig. 6, we also depict the evolution of *MFEV* provided by the four compared algorithms on the ten test problems. From Fig. 6, ReDSADE consistently has faster convergence speed than the three competitors. The superiority of ReDSADE against EGO and CAL-SAPSO can be attributed to the fact that these two competitors establish surrogate models in the whole decision space and, as a result, cannot approximate the objective function of an EOPDR well. Although GLoSADE also combines a global surrogate model with a local surrogate model to assist the optimization, similar to EGO and CAL-SAPSO, GLoSADE establishes the global surrogate model in the whole decision space. Therefore, ReDSADE also obtains better results than GLoSADE.

D. Comparison with Three State-of-the-Art SAEAs on the Second Set of Test Problems with 10 and 30 Dimensions

Subsequently, the second set of test problems was utilized to test the performance of ReDSADE. Considering that one may be interested in the performance of ReDSADE on high-dimensional EOPDRs, we also tested the performance of ReDSADE on 30-dimensional test problems. The performance of ReDSADE was compared with that of EGO, CAL-SAPSO, and GLoSADE. The results of these four compared algorithms are given in Table S-I and Table S-II of the supplementary file.

From Table S-I and Table S-II, overall, the performance of ReDSADE is superior to that of the three competitors. To be specific, for the 10-dimensional test problems, ReDSADE obtains the best *MOFV* values on seven test problems. However, EGO and CAL-SAPSO produce better *MOFV* values than ReDSADE on only one and two test problems, respectively. Moreover, GLoSADE cannot provide better *MOFV* value than ReDSADE on any test problem. For the 30-dimensional test problems, ReDSADE achieves the best *MOFV* values on seven test problems. It provides worse *MOFV* values than EGO and CAL-SAPSO on only one and two test problems, respectively. Similarly, GLoSADE fails to obtain better *MOFV* value than ReDSADE on any test problem. According to the Wilcoxon's rank-sum test, for the 10-dimensional test problems, ReDSADE performs better than EGO, CAL-SAPSO, and GLoSADE on four, eight, and ten test problems, respectively. In addition, for the 30-dimensional test problems, ReDSADE beats EGO, CAL-SAPSO, and GLoSADE on six, eight, and eight test problems, respectively.

E. Comparison with Three State-of-the-Art SAEAs on the 10-Dimensional and 30-Dimensional BBOB Test Problems

Finally, the experiments were executed on the 10-dimensional and 30-dimensional BBOB test problems, in which we did not set any discontinuity, with the aim of investigating the generality of ReDSADE. The results of EGO, CAL-SAPSO, GLoSADE, and ReDSADE are presented in Table S-III and Table S-IV of the supplementary file.

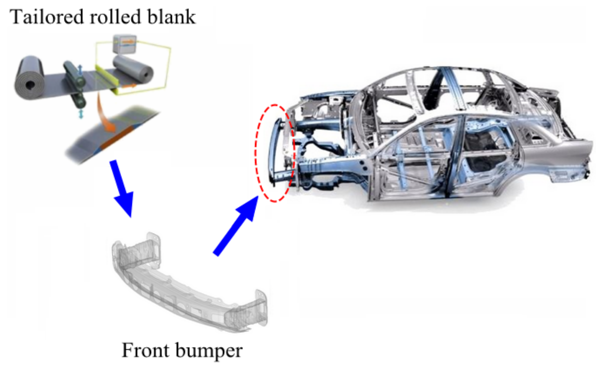


Fig. 7. TRB-based front bumper in an automobile [26].

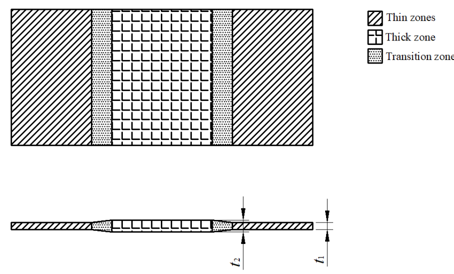


Fig. 8. Thin zones, thick zones, and transition zones in a TRB.

From Table S-III and Table S-IV, ReDSADE also exhibits better performance on solving the BBOB test problems. For the 10-dimensional BBOB test problems, ReDSADE obtains the best *MFEV* values on 14 test problems. With respect to the 30-dimensional BBOB test problems, ReDSADE produces the best results on 14 test problems. In Fig. S-I and Fig. S-II of the supplementary file, we also depicted the evolution of *MFEV* provided by the four compared algorithms on the 10-dimensional and 30-dimensional BBOB test problems, respectively. According to the Wilcoxon's rank-sum test, for the 10-dimensional BBOB test problems, ReDSADE performs better than EGO, CAL-SAPSO, and GLoSADE on 17, 17, and 21 test problems, respectively. In addition, for the 30-dimensional BBOB test problems, ReDSADE outperforms EGO, CAL-SAPSO, and GLoSADE on 13, 13, and 21 test problems, respectively.

V. Real-World Application

To reduce the fuel consumption and improve the safety of an automobile, it is desirable to design an automobile body with low weight and high crashworthiness. The front bumper is an important component to protect passenger from injury and damage induced by severe collapse [49]. Commonly, a front bumper is made by the equal constant thickness blanks. In this paper, we try to use tailor rolled blanks (TRBs) instead of the equal constant thickness blanks, thus further reducing the weight and enhancing the crashworthiness of the front bumper (as shown in Fig. 7). Such a lightweight and crashworthiness design problem is a typical EOPDR. In Fig. 8, according to the thicknesses of different zones, a TRB can be divided into

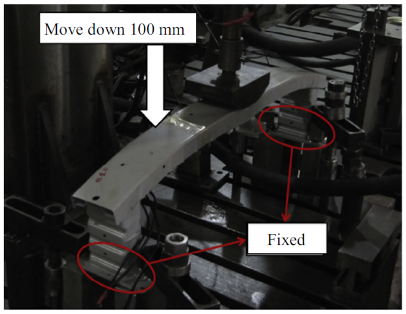
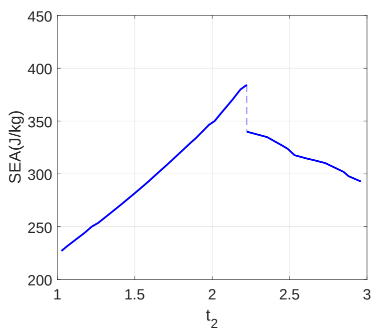
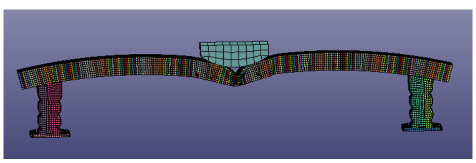


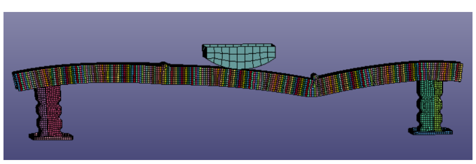
Fig. 9. Three-point bending test [49].



(a)

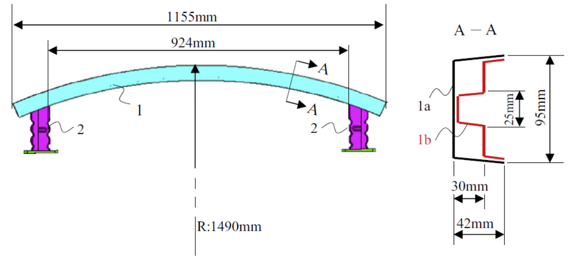


(b)

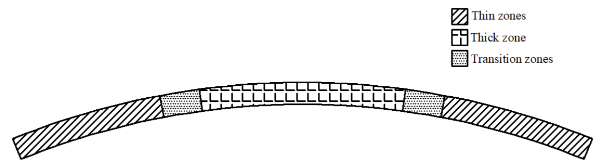


(c)

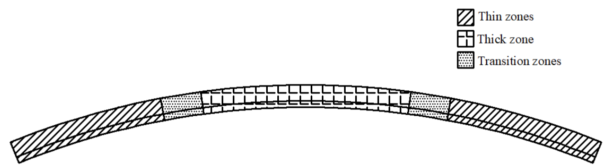
Fig. 10. Change of SEA according to t_2 and two kinds of deformation modes. (a) The response of SEA. (b) The first kind of deformation mode. (c) The second kind of deformation mode.



(a)



(b)



(c)

Fig. 11. Structure of the considered TRB-based front bumper. (a) The total structure. (b) The outer sheet (i.e., 1a). (c) The inner sheet (i.e., 1b).

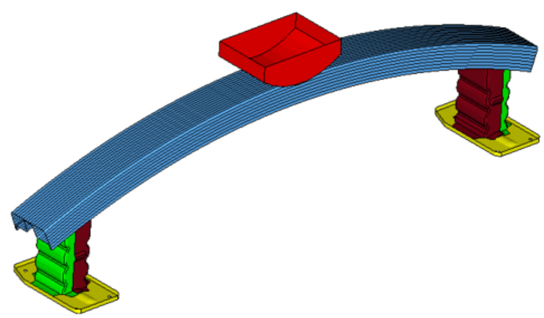


Fig. 12. Finite element analysis model of the three-point bending test.

thin zones, thick zones, and transition zones. By adjusting the thicknesses of different zones, we can maximize the specific energy absorption (SEA), i.e., the energy absorption per unit weight, of a TRB-based front bumper in the three-point bending test (as shown in Fig. 9). It should be noted that, with the change of thicknesses, a TRB-based front bumper may exhibit different deformation modes, which causes that the response of SEA is discontinuous. For example, in Fig. 10, we describe the change of SEA according to the thickness of thick zones (i.e., t_2). From Fig. 10(a), it can be observed that the response of SEA is discontinuous at $t_2 = 2.2mm$. This is because when $t_2 \leq 2.2mm$, the deformation mode in Fig. 10(b) happens, and when $t_2 > 2.2mm$, the deformation mode in Fig. 10(c) occurs.

The structure of the considered TRB-based front bumper is shown in Fig. 11. It consists of a curved front bumper and two crash boxes. The curved front bumper includes an outer sheet (i.e., 1a in Fig 11(a)) and an inner sheet (i.e., 1b in Fig 11(a)). Both of these two sheets are made by TRBs. As shown in Fig 11(b) and Fig 11(c), each sheet contains two thin zones and one thick zone. The thicknesses of these zones are considered as the decision variables, denoted as $x_{o,thin}$, $x_{o,thick}$, $x_{i,thin}$, and $x_{i,thick}$. Specifically, $x_{o,thin}$ and $x_{o,thick}$ are the thicknesses of the thin zone and the thick zones of the outer sheet, respectively, and $x_{i,thin}$ and $x_{i,thick}$ are the thicknesses of the thin zone and the thick zones of the inner sheet, respectively. The finite element analysis model is shown in Fig. 12. To optimize both the weight and crashworthiness of the front bumper, SEA is employed as the indicator of

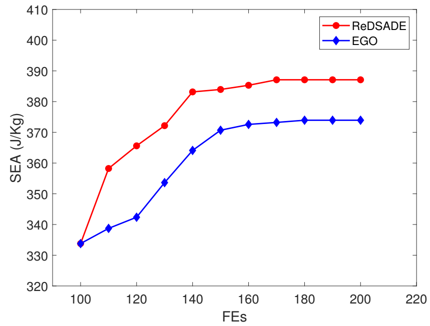


Fig. 13. Evolution of the average SEA values derived from EGO and ReDSADE.

TABLE II
THE ORIGINAL DESIGN AND THE BEST DESIGNS PROVIDED BY EGO AND REDSADE.

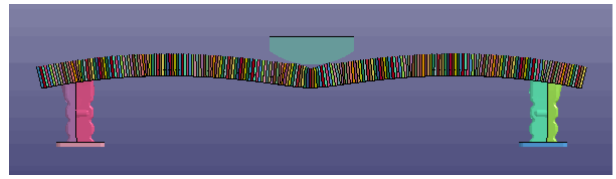
Status	Original Design	EGO	ReDSADE
$x_{o,thin}$	1.50 *mm*	1.50 *mm*	1.50 *mm*
$x_{o,thick}$	1.50 *mm*	0.78 *mm*	0.77 *mm*
$x_{i,thin}$	1.50 *mm*	1.50*mm*	1.50 *mm*
$x_{i,thick}$	1.50 *mm*	1.14 *mm*	1.07 *mm*
SEA	295.73 *J/kg*	383.79 *J/kg*	391.17 *J/kg*

lightweight and crashworthiness design [49]. The function of SEA is formulated as: $SEA(\mathbf{x}) = EA(\mathbf{x})/M(\mathbf{x})$, where $EA(\mathbf{x})$ and $M(\mathbf{x})$ are the energy absorption and mass of the TRB-based front bumper, respectively. Both $EA(\mathbf{x})$ and $M(\mathbf{x})$ are obtained by the finite element analysis. Thus, this design problem can be described as follows:

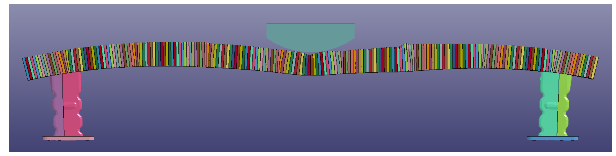
$$\begin{aligned} \max:\ & SEA(\mathbf{x}) \\ & \mathbf{x} = (x_{o,thin},\ x_{o,thick},\ x_{i,thin},\ x_{i,thick}) \\ & x_{o,thin},\ x_{o,thick},\ x_{i,thin},\ x_{i,thick} \in [0.5, 1.5] \end{aligned} \tag{25}$$

ReDSADE was applied to solve this design problem, and $MaxFEs$ was set to 200. The optimization was independently implemented ten times[4]. The average SEA value over ten independent runs provided by ReDSADE is 387.10 J/kg, which is better than the original design, i.e., 295.73 J/kg. For comparison, we also used EGO to cope with this design problem. It provides an average SEA value of 373.93 J/kg over ten independent runs. Obviously, ReDSADE achieves the best average design performance. Meanwhile, Fig. 13 plots the evolution of the average SEA values derived from EGO and ReDSADE. From Fig. 13, ReDSADE has better convergence performance. In addition, in Table II, we listed the original design and the best designs provided by ReDSADE and EGO. The corresponding deformations are shown in Fig. 14. Compared with the two competitors, ReDSADE provides the design result with the maximum SEA value and minimum degree of deformation. The above results verify the effectiveness of ReDSADE in this application.

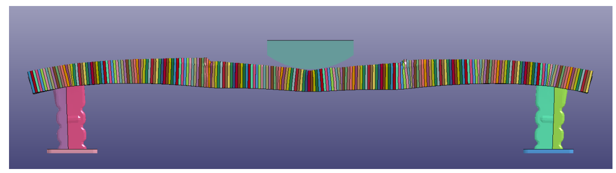
[4]It needs about 30 minutes to complete a simulation. Therefore, it takes about 30 $min * 200 * 10/(60 * 24) \approx 41.6$ days to complete the ten runs.



(a)



(b)



(c)

Fig. 14. Deformations of the original design and the best designs provided by EGO and ReDSADE. (a) The original design. (b) The best design provided by EGO. (c) The best design provided by ReDSADE.

VI. CONCLUSION

Real-world optimization problems may have discontinuous and expensive objective functions, which are the so-called EOPDRs. Few attempts have been made to solve this kind of optimization problem in the EA community. In this paper, a surrogate-assisted DE algorithm with region division, called ReDSADE, was proposed to solve EOPDRs. ReDSADE included three main strategies: the region division strategy, the Kriging-based search, and the RBF-based local search. The region division strategy divided the decision space into several subregions, with the aim of making the objective function continuous in them. Then, in the Kriging-based search, DE was combined with Kriging models to guide the evolution. Finally, in the RBF-based local search, DE was combined with RBF to search for a more potential solution around the current best solution. The comparative studies on the three sets of test problems showed the good performance of ReDSADE. We also applied ReDSADE to solve the lightweight and crashworthiness design of the TRB-based front bumper. The results verified the effectiveness of ReDSADE in this real-world application.

In the future, we will try to solve large-scale EOPDRs (e.g., EOPDRs with more than 50 dimensions). For a large-scale EOP, it is very hard to establish an accurate surrogate model with limited solutions. When a large-scale EOP is with discontinuous response, it is more difficult to approximate the objective function accurately by making use of surrogate models. Therefore, solving large-scale EOPDRs deserves in-depth research in the future.

References

[1] G. Zhu, G. Sun, Q. Liu, G. Li, and Q. Li, "On crushing characteristics of different configurations of metal-composites hybrid tubes," *Composite Structures*, vol. 175, pp. 58–69, 2017.

[2] H. Zhang, G. Sun, X. Zhi, G. Li, and Q. Li, "Bending characteristics of top-hat structures through tailor rolled blank (TRB) process," *Thin-Walled Structures*, vol. 123, pp. 420–440, 2018.

[3] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 644–660, 2017.

[4] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

[5] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2664–2677, 2017.

[6] M. N. Le, Y. S. Ong, S. Menzel, Y. Jin, and B. Sendhoff, "Evolution by adapting surrogates," *Evolutionary Computation*, vol. 21, no. 2, pp. 313–340, 2013.

[7] X. Lu, T. Sun, and K. Tang, "Evolutionary optimization with hierarchical surrogates," *Swarm and Evolutionary Computation*, vol. 47, pp. 21–32, 2019.

[8] F. Li, X. Cai, and L. Gao, "Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems," *Applied Soft Computing*, vol. 74, pp. 291–305, 2019.

[9] X. Sun, D. Gong, Y. Jin, and S. Chen, "A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 685–698, 2013.

[10] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Information Sciences*, vol. 454, pp. 59–72, 2018.

[11] Y. Wang, D.-Q. Yin, S. Yang, and G. Sun, "Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1642–1656, 2018.

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

[13] X. Cai, L. Gao, and X. Li, "Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 365–379, 2020.

[14] 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 Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 815–827, 2019.

[15] 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.

[16] 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.

[17] N. Namura, K. Shimoyama, and S. Obayashi, "Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 898–913, 2017.

[18] D. Zhan, Y. Cheng, and J. Liu, "Expected improvement matrix-based infill criteria for expensive multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 956–975, 2017.

[19] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, 2006.

[20] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 459–472, 2018.

[21] 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.

[22] 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.

[23] 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.

[24] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 536–549, 2020.

[25] 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.

[26] G. Sun, H. Zhang, J. Fang, G. Li, and Q. Li, "Multi-objective and multi-case reliability-based design optimization for tailor rolled blank (TRB) structures," *Structural & Multidisciplinary Optimization*, vol. 55, no. 5, pp. 1899–1916, 2017.

[27] Y. Lian and M. S. Liou, "Multiobjective optimization using coupled response surface model and evolutionary algorithm," *European Radiology*, vol. 43, no. 6, pp. 1316–1325, 2005.

[28] M. Herrera, A. Guglielmetti, M. Xiao, and R. F. Coelho, "Metamodel-assisted optimization based on multiple kernel regression for mixed variables," *Structural & Multidisciplinary Optimization*, vol. 49, no. 6, pp. 979–991, 2014.

[29] 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.

[30] B. Liu, Q. Zhang, and G. G. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2013.

[31] M. Meckesheimer, R. R. Barton, T. W. Simpson, F. Limayem, and B. Yannou, "Metamodeling of combined discrete/continuous responses," *AIAA Journal*, vol. 39, no. 10, pp. 1950–1959, 2001.

[32] A. A. Gorodetsky and Y. M. Marzouk, "Efficient localization of discontinuities in complex computational simulations," *SIAM Journal on Scientific Computing*, vol. 36, no. 6, pp. A2584–A2610, 2014.

[33] A. Basudhar, S. Missoum, and A. H. Sanchez, "Limit state function identification using support vector machines for discontinuous responses and disjoint failure domains," *Probabilistic Engineering Mechanics*, vol. 23, no. 1, pp. 1–11, 2008.

[34] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231.

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

[36] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[37] Y. Wang, H. Liu, H. Long, Z. Zhang, and S. Yang, "Differential evolution with a new encoding mechanism for optimizing wind farm layout," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1040–1054, 2018.

[38] Y. Wang, Z.-Z. Liu, J. Li, H.-X. Li, and J. Wang, "On the selection of solutions for mutation in differential evolution," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 297–315, 2018.

[39] Z. Y. Luo, W. F. Zhang, B. Y. Ye, and L. Q. Cai, "Wavelet SVM ensemble for pattern classification with quantum-inspired evolutionary algorithm," *2008 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 2, pp. 485 – 490, 2008.

[40] T. Yeoh, S. Zapotecas-Martnez, Y. Akimoto, H. Aguirre, and K. Tanaka, "Genetic algorithm assisted by a SVM for feature selection in gait classification," *2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 191–195, 2014.

[41] C. Saunders, M. O. Stitson, J. Weston, R. Holloway, L. Bottou, B. Scholkopf, and A. Smola, "Support Vector Machine," *Computer Science*, vol. 1, no. 4, pp. 1–28, 2002.

[42] J. Močkus, "On bayesian methods for seeking the extremum," in *Optimization techniques IFIP technical conference*. Springer, 1975, pp. 400–404.

[43] W. Wang, Y. Wu, C. Tang, and M. Hor, "Adaptive density-based spatial clustering of applications with noise (DBSCAN) according to data," in

2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 1, 2015, pp. 445–451.

[44] L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, *Training a Support Vector Machine in the Primal*, 2007, pp. 29–50.

[45] S. F. Galn, "Comparative evaluation of region query strategies for dbscan clustering," *Information Sciences*, vol. 502, pp. 76–90, 2019.

[46] Y. Wang, D. Q. Yin, S. Yang, and G. Sun, "Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1642–1656, 2019.

[47] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Natural Computing*, pp. 341–357, 2005.

[48] N. Hansen, A. Auger, S. Finck, and R. Ros, "Real-parameter black-box optimization benchmarking: Experimental setup," *Orsay, France: Université Paris Sud, Institut National de Recherche en Informatique et en Automatique (INRIA) Futurs, Équipe TAO, Tech. Rep*, 2014.

[49] Z. Xiao, J. Fang, G. Sun, and Q. Li, "Crashworthiness design for functionally graded foam-filled bumper beam," *Advances in Engineering Software*, vol. 85, pp. 81–95, 2015.

Yong Wang (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

He is a Professor with the School of Automation, Central South University, Changsha, China. His current research interests include the theory, algorithm design, and interdisciplinary applications of computational intelligence.

Dr. Wang is an Associate Editor of the *IEEE Transactions on Evolutionary Computation* and the *Swarm and Evolutionary Computation*. He was a recipient of Cheung Kong Young Scholar by the Ministry of Education, China, in 2018, and a Web of Science highly cited researcher in Computer Science in 2017 and 2018.

Jianqing Lin received the B.S. degree in automation from Tiangong University, Tianjin, China, in 2017, and the M.S. degree in control engineering from Central South University, Changsha, China, in 2020. He is currently pursuing the Ph.D. degree in artificial intelligence, Huazhong University of Science and Technology, Wuhan, China.

His current research interests include surrogate-assisted evolutionary optimization, large-scale optimization, and their applications.

Jiao Liu received the B.S. degree in process equipment and control engineering and the M.S. degree in power engineering and engineering thermophysics both from the Taiyuan University of Technology, Taiyuan, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in control science and engineering, Central South University, Changsha, China. His current research interests include evolutionary computation, mixed-integer programming, and lightweight design of automobile.

Guangyong Sun received the Ph.D. degree in Mechanical Engineering from the Hunan University, Changsha, China, in 2011.

He is a Professor with the State Key Laboratory of Advanced Design and Manufacture for Vehicle Body, Hunan University, Changsha, China. His current research interests include lightweight design, structural optimization, and automotive safety.

Dr. Sun is an Associate Editor of the *Thin-Walled Structures* and the *Applied Composite Materials*. He was a Web of Science highly cited researcher in Engineering in 2020.

Tong Pang received the Ph.D. degree in mechanical engineering from the Hunan University, Changsha, China, in 2019.

He is currently a research associate with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China. His current research interests include lightweight design, structural optimization, and automotive crashworthiness.

Supplementary File of "Surrogate-Assisted Differential Evolution with Region Division for Expensive Optimization Problems with Discontinuous Responses"

S-I. Discussions

A. Effectiveness of the RBF-Based Local Search

As introduced in Section III-E, ReDSADE employed the RBF-based local search to accelerate the convergence. One may be interested in the effectiveness of this strategy. To this end, a variant of ReDSADE, called ReDSADE-WoLocal, was devised. In this variant, the RBF-based local search was removed. The results of ReDSADE and ReDSADE-WoLocal on the first set of test problems are summarized in Table S-V. As shown in Table S-V, ReDSADE provides better *MFEV* values than ReDSADE-WoLocal on all the ten test problems. According to the Wilcoxon's rank-sum test, ReDSADE outperforms ReDSADE-WoLocal on all the ten test problems. From the above results, we can conclude that the RBF-based local search is able to enhance the convergence speed of ReDSADE.

B. Effectiveness of ODD

As mentioned in Section III-C, in the region division strategy, *ODD* was employed as the distance measure for DBSCAN. To investigate its effectiveness, we designed a variant of ReDSADE, called ReDSADE-ED. In ReDSADE-ED, the Euclidean distance of solutions in the decision space was used for DBSCAN. We used the first set of test problems to investigate the performance of these two algorithms. The results are summarized in Table S-VI. From Table S-VI, ReDSADE provides better *MFEV* values than ReDSADE-ED on all the ten test problems. According to the Wilcoxon's rank-sum test, ReDSADE performs better than ReDSADE-ED on nine test problems.

In addition, considering that the gap size in the objective space, i.e., the Δ value of the first and second sets of test problems in Section S-III of the supplementary file, may influence the effectiveness of *ODD*, we also investigated the impact of the Δ value. We selected S1-F2 as the test problem, and the Δ value changes from 10 to 10^6. ReDSADE and ReDSADE-ED were used to solve S1-F2 with different Δ values, and the results are summarized in Table S-VII. It can be observed that, when the value of Δ is relatively large (i.e., $\Delta = 10^6$, $\Delta = 10^5$ or $\Delta = 10^4$), ReDSADE can provide significantly better results than ReDSADE-ED. It is because ReDSADE has the capability to distinguish the solutions in different subregions based on *ODD*. When the value of Δ is relatively small (i.e., $\Delta = 10^3$, $\Delta = 10^2$, or $\Delta = 10$), distinguishing solutions in different subregions based on *ODD* is not easy. Note, however, that when the value of Δ becomes small, the difficulty of approximating the objective function is also reduced. Therefore, under this condition, both ReDSADE and ReDSADE-ED provide similar results.

The above results demonstrate the effectiveness of *ODD*.

C. Effectiveness of Optimizing EI Functions in the Kriging-Based Search

In the Kriging-based search, we adopted DE to optimize the EI function based on each subpopulation. After several iterations, the solution with the best EI function value is chosen from all the subpopulations and evaluated by the original objective function. Instead of optimizing the EI functions by using DE, we also investigated the following two cases: 1) generate offspring solutions for each subpopulation by using Monte Carlo sampling, select the solution with the best EI function value from all offspring solutions, and evaluate this solution by the original objective function, and 2) generate offspring solutions for each subpopulation by using DE, select the solution with the best EI function value from all offspring solutions, and evaluate this solution by the original objective function. In principle, these two cases use pre-selection rather than optimization to update the database. As a result, two variants of ReDSADE, i.e., ReDSADE-MC and ReDSADE-WoO, were designed. The first set of test problems and the 10-dimensional BBOB test problems were utilized to investigate the performance of ReDSADE-MC, ReDSADE-WoO, and ReDSADE. The results are given in Table S-VIII and Table S-IX. From Table S-VIII, for seven test problems, ReDSADE achieves the best results. From Table S-IX, for 19 test problems, ReDSADE obtains the best results. According to the Wilcoxon's rank-sum test, ReDSADE is better than ReDSADE-MC and ReDSADE-WoO on ten and seven test problems in terms of the first set of test problems, respectively, and 15 and 22 test problems in terms of the 10-dimensional BBOB test problems, respectively. The above results verify the effectiveness of optimizing EI functions in the Kriging-based search.

D. Effectiveness of DBSCAN

In this paper, DBSCAN was employed as the clustering method to partition the evaluated solutions into different clusters. In Section II-E, we have pointed out the reason why we choose DBSCAN as the clustering method: it does not need to predefine the number of clusters. However, some other clustering methods also have this characteristic. One may be interested in whether other clustering methods can also be used in ReDSAEA. To this end, we designed a variant of ReDSAEA (named as ReDSADE-KM). In ReDSADE-KM, an adaptive K-means algorithm [**?**] was used as the clustering method. The results of ReDSADE and ReDSADE-KM on the first set of test problems are provided in Table S-X. From Table S-X, ReDSADE and ReDSADE-KM exhibit the similar overall performance. Therefore, the influence of the clustering method is not obvious and the adaptive K-means can also be used in ReDSADE.

E. Running Time of ReDSADE

We used the second set of test problems with 10 dimension and 30 dimension to investigate the runtime of EGO, CAL-SAPSO, GLoSADE, and ReDSADE. All the algorithms were conducted in Windows 7 Professional Operating System environment with an Intel(R) Core(TM) i5-7500 CPU @ 3.4GHz 3.4GHz and 8GB RAM, and the programs were implemented in Matlab 2017a. Table S-XI summarizes the average runtime provided by the four compared algorithms over 20 independent runs. From Table S-XI, GLoSADE has the shortest running time. It is because GLoSADE does not adopt Kriging as the surrogate model. Although all of CAL-SAPSO, GLoSADE, and ReDSADE make use of Kriging as the surrogate model, the running time of ReDSADE is still shorter than that of EGO and CAL-SAPSO. The reason is that ReDSADE establishes a surrogate model in each subregion based on a cluster rather than the database.

F. Study of the Parameter Settings in ReDSADE

In ReDSADE, two parameters, i,e, K and H, were introduced. We also investigated the influence of these two parameters. Two test problems, i.e., S1-F5 and S2-F8, were employed to test the influence of different parameters settings. K and H were selected from the following two sets: $\{20,50,80\}$ and $\{20,50,80\}$, respectively. Fig. S-3 records the *MFEV*/*MOFV* values provided by the nine different combinations of K and H. It can be observed from Fig. S-3 that $H \leq 50$ and $K \leq 50$ are the best choice for ReDSADE.

S-II. Results

TABLE S-I

Results of EGO, CAL-SAPSO, GLoSADE, and ReDSADE on the Second Set of Test Problems With 10 Dimensions. The Wilcoxon's Rank-sum Test at a 0.05 Significance Level Was Performed Between ReDSADE and Each of EGO, CAL-SAPSO, and GLoSADE.

Problem	EGO		CAL-SAPSO		GLoSADE		ReDSADE
	MOFV ± *Std Dev*		*MOFV* ± *Std Dev*		*MOFV* ± *Std Dev*		*MOFV* ± *Std Dev*
S2-F1	**6.32E+02±1.00E+02**	−	9.62E+02±6.02E+01	+	8.51E+02±1.20E+02	+	6.90E+02±2.35E+02
S2-F2	5.86E+02±1.34E+02	+	**3.57E+02±5.55E+01**	−	6.87E+02±5.95E+01	+	5.47E+02±1.10E+02
S2-F3	6.00E+02±9.56E+01	≈	**4.78E+02±4.45E+01**	−	6.49E+02±1.10E+02	+	5.84E+02±7.09E+01
S2-F4	9.85E+02±6.11E+01	+	1.20E+03±6.28E+01	+	1.21E+03±6.49E+01	+	**9.49E+02±1.11E+02**
S2-F5	9.52E+02±8.07E+01	≈	1.19E+03±9.25E+01	+	1.18E+03±7.76E+01	+	**8.58E+02±1.65E+02**
S2-F6	1.12E+03±5.73E+01	+	1.18E+03±9.29E+01	+	1.12E+03±1.03E+02	+	**9.96E+02±1.33E+02**
S2-F7	1.57E+03±8.05E+01	≈	1.66E+03±4.75E+01	+	1.74E+03±3.84E+01	+	**1.55E+03±1.52E+02**
S2-F8	1.46E+03±3.19E+01	≈	1.53E+03±1.09E+02	+	2.11E+03±5.84E+02	+	**1.42E+03±1.35E+02**
S2-F9	1.56E+03±8.88E+01	≈	1.64E+03±3.40E+01	+	1.71E+03±2.62E+01	+	**1.53E+03±2.03E+02**
S2-F10	1.29E+03±1.36E+01	+	1.25E+03±1.41E+01	+	1.41E+03±1.11E+02	+	**1.18E+03±2.99E+02**
+/−/≈	4/1/5		8/2/0		10/0/0		

TABLE S-II

Results of EGO, CAL-SAPSO, GLoSADE, and ReDSADE on the Second Set of Test Problems With 30 Dimensions. The Wilcoxon's Rank-sum Test at a 0.05 Significance Level Was Performed Between ReDSADE and Each of EGO, CAL-SAPSO, and GLoSADE.

Problem	EGO		CAL-SAPSO		GLoSADE		ReDSADE
	MOFV ± *Std Dev*		*MOFV* ± *Std Dev*		*MOFV* ± *Std Dev*		*MOFV* ± *Std Dev*
S2-F1	**7.28E+02±8.72E+01**	≈	1.01E+03±3.45E+01	+	1.10E+03±1.08E+02	+	8.05E+02±1.20E+02
S2-F2	7.95E+02±1.51E+02	≈	8.98E+02±9.80E+01	+	9.26E+02±1.65E+02	+	**7.47E+02±6.34E+01**
S2-F3	1.14E+03±1.68E+02	≈	**1.00E+03±1.57E+01**	≈	1.18E+03±2.94E+02	≈	1.10E+03±1.64E+02
S2-F4	1.13E+03±1.53E+02	+	1.22E+03±1.36E+02	+	1.13E+03±5.12E+01	+	**1.04E+03±6.71E+01**
S2-F5	1.38E+03±4.15E+01	+	1.22E+03±1.15E+02	+	1.16E+03±4.11E+01	≈	**1.14E+03±1.09E+02**
S2-F6	1.38E+03±7.09E+01	+	1.41E+03±2.41E+01	+	1.20E+03±1.34E+02	+	**9.60E+02±2.73E+01**
S2-F7	1.54E+03±1.75E+01	+	**1.25E+03±2.04E+02**	≈	1.60E+03±4.07E+01	+	1.30E+03±1.78E+02
S2-F8	2.10E+03±2.26E+02	+	2.37E+03±9.58E+01	+	3.13E+03±1.14E+03	+	**1.70E+03±1.42E+02**
S2-F9	1.54E+03±1.75E+01	≈	1.65E+03±1.23E+02	+	1.64E+03±1.09E+02	+	**1.46E+03±3.58E+01**
S2-F10	1.64E+03±6.41E+01	+	1.59E+03±9.07E+01	+	1.68E+03±1.26E+02	+	**1.34E+03±6.20E+01**
+/−/≈	6/0/4		8/0/2		8/0/2		

TABLE S-III

RESULTS OF EGO, CAL-SAPSO, GLOSADE, AND REDSADE ON THE 10-DIMENSIONAL BBOB TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN REDSADE AND EACH OF EGO, CAL-SAPSO, AND GLOSADE.

Problem	EGO		CAL-SAPSO		GLoSADE		ReDSADE
	MOFV ± Std Dev		*MOFV ± Std Dev*		*MOFV ± Std Dev*		*MOFV ± Std Dev*
BBOB1	2.49E-02±2.24E-02	+	**7.74E-09±6.33E-09**	−	1.29E+01±3.90E+00	+	3.33E-08±3.36E-08
BBOB2	6.43E+03±2.69E+03	+	6.79E+02±2.96E+02	+	2.47E+05±7.63E+04	+	**4.63E+02±4.79E+02**
BBOB3	7.37E+01±1.67E+01	+	6.15E+01±9.09E+00	+	1.74E+02±3.88E+01	+	**4.29E+01±6.23E+00**
BBOB4	1.88E+02±2.15E+01	+	2.33E+02±5.18E+01	+	2.32E+02±5.37E+01	+	**8.37E+01±1.34E+01**
BBOB5	**1.00E-08±0.00E+00**	≈	3.73E-02±3.42E-02	+	2.54E+00±2.03E+00	+	**1.00E-08±0.00E+00**
BBOB6	8.15E+01±2.57E+01	+	9.54E+01±1.88E+01	+	1.07E+02±5.80E+01	+	**6.15E+01±8.23E+00**
BBOB7	6.22E+00±3.58E+00	≈	**3.47E+01±3.60E+00**	≈	7.78E+01±3.91E+01	≈	7.81E+00±3.09E+00
BBOB8	4.92E+02±2.13E+02	+	1.59E+02±5.50E+01	+	1.27E+03±1.02E+03	+	**1.82E+01±1.76E+01**
BBOB9	5.01E+02±1.58E+02	+	2.82E+01±1.98E+01	+	8.70E+03±7.79E+03	+	**8.85E+00±1.69E+00**
BBOB10	2.65E+04±9.92E+03	+	**1.69E+04±2.65E+04**	−	6.51E+04±2.74E+04	+	2.13E+04±7.84E+03
BBOB11	1.24E+02±3.05E+01	≈	1.04E+02±3.47E+01	+	**7.79E+01±4.23E+01**	≈	9.95E+01±2.66E+01
BBOB12	2.14E+07±6.19E+06	+	2.40E+07±0.00E+00	+	1.67E+07±1.14E+07	+	**1.67E+01±1.12E+01**
BBOB13	6.86E+01±1.63E+01	+	1.77E+02±1.38E+02	+	6.85E+02±2.50E+02	+	**7.92E-01±8.13E-01**
BBOB14	3.23E+00±7.46E-01	+	5.07E+00±5.41E+00	+	5.44E+00±3.60E+00	+	**3.41E-03±1.24E-03**
BBOB15	8.41E+01±1.25E+01	+	1.16E+02±3.20E+01	+	1.48E+02±2.99E+01	+	**4.83E+01±1.81E+01**
BBOB16	**7.36E+00±2.38E+00**	≈	1.74E+01±3.81E+00	+	1.94E+01±7.68E+00	+	1.23E+01±2.23E+00
BBOB17	4.56E+00±1.33E+00	+	6.36E+00±0.00E+00	+	5.68E+00±1.89E+00	+	**2.18E+00±2.91E-01**
BBOB18	2.29E+01±3.60E+00	+	2.83E+01±6.33E+00	+	2.48E+01±4.33E+00	+	**5.50E+00±2.22E+00**
BBOB19	5.67E+00±9.52E-01	+	**4.61E+00±6.15E-01**	≈	6.32E+00±7.01E-01	+	4.84E+00±6.95E-01
BBOB20	3.52E+00±4.58E-01	+	3.20E+00±2.14E-01	+	9.34E+02±1.19E+03	+	**2.81E+00±4.00E-01**
BBOB21	**2.28E+00±1.63E+00**	≈	2.72E+00±1.24E+00	≈	5.04E+01±2.72E+00	+	4.99E+00±2.88E+00
BBOB22	3.34E+00±3.96E+00	≈	**2.71E+00±2.33E+00**	≈	5.27E+01±1.60E+01	+	4.31E+00±2.68E+00
BBOB23	**2.40E+00±7.56E-01**	≈	2.68E+00±7.67E-01	≈	3.68E+00±8.55E-01	+	2.42E+00±5.11E-01
BBOB24	6.35E+01±5.88E+00	+	6.42E+01±6.33E+00	+	1.06E+02±2.59E+01	+	**4.49E+01±1.21E+01**
+/−/≈	17/0/7		17/2/5		22/0/2		

TABLE S-IV

RESULTS OF EGO, CAL-SAPSO, GLOSADE, AND REDSADE ON THE 30-DIMENSIONAL BBOB TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN REDSADE AND EACH OF EGO, CAL-SAPSO, AND GLOSADE.

Problem	EGO		CAL-SAPSO		GLoSADE		ReDSADE
	MOFV ± Std Dev		*MOFV ± Std Dev*		*MOFV ± Std Dev*		*MOFV ± Std Dev*
BBOB1	7.00E+00±1.95E+00	+	**6.06E-01±2.58E-01**	−	6.21E+01±1.64E+01	+	3.24E+00±1.24E+00
BBOB2	1.91E+06±4.83E+05	+	3.43E+06±1.22E+06	+	1.55E+06±4.25E+05	+	**3.51E+05±1.31E+05**
BBOB3	4.84E+02±3.94E+01	≈	**3.82E+02±3.42E+01**	≈	6.13E+02±7.10E+01	+	4.37E+02±1.14E+02
BBOB4	9.44E+02±1.56E+02	+	8.55E+02±5.71E+01	+	1.08E+03±1.77E+02	+	**7.40E+02±1.23E+02**
BBOB5	1.26E+00±1.86E+00	+	7.08E+00±2.26E+00	+	1.26E+01±1.56E+01	+	**1.00E-08±0.00E+00**
BBOB6	2.43E+03±1.91E+03	+	**1.98E+02±7.59E+00**	−	2.53E+04±1.42E+04	+	4.54E+02±9.27E+01
BBOB7	2.34E+02±4.50E+01	+	3.61E+02±1.26E+02	+	4.27E+02±1.40E+02	+	**1.74E+02±5.14E+01**
BBOB8	1.34E+04±7.38E+03	+	**4.47E+02±2.08E+02**	−	9.63E+04±6.77E+04	+	7.99E+03±2.24E+03
BBOB9	4.67E+03±1.69E+03	−	**3.74E+02±9.13E+01**	−	7.81E+04±1.54E+04	+	7.28E+03±1.90E+03
BBOB10	**1.05E+06±2.05E+05**	≈	2.12E+06±2.03E+06	+	2.57E+06±1.16E+06	+	1.51E+06±3.04E+05
BBOB11	3.58E+02±5.97E+01	≈	6.49E+02±2.86E+01	+	3.64E+02±8.86E+01	≈	**3.32E+02±9.49E+01**
BBOB12	1.30E+08±3.66E+07	≈	3.30E+08±5.49E+07	+	1.72E+08±7.27E+07	+	**1.12E+08±2.02E+07**
BBOB13	4.90E+02±5.72E+01	+	6.63E+02±4.54E+01	+	1.59E+03±3.54E+02	+	**4.78E+02±6.30E+01**
BBOB14	1.64E+01±1.35E+00	≈	**5.85E+00±1.32E+00**	−	3.54E+01±7.74E+00	+	1.46E+01±2.83E+00
BBOB15	5.25E+02±6.59E+01	+	4.51E+02±5.83E+01	+	7.53E+02±4.58E+01	+	**3.87E+02±3.44E+01**
BBOB16	3.53E+01±5.81E+00	≈	3.57E+01±2.66E+00	+	3.41E+01±4.86E+00	≈	**3.31E+01±2.63E+00**
BBOB17	9.17E+00±1.21E+00	≈	7.51E+00±2.41E-01	≈	1.15E+01±2.02E+00	+	**7.15E+00±5.91E+00**
BBOB18	3.60E+01±5.39E+00	≈	3.02E+01±6.52E+00	≈	4.14E+01±5.61E+00	+	**2.75E+01±1.76E+00**
BBOB19	8.10E+00±5.23E-01	≈	**7.06E+00±2.02E-01**	−	1.79E+01±3.36E+00	+	9.09E+00±5.95E-01
BBOB20	6.41E+02±7.59E+02	−	**6.19E+02±8.70E+02**	−	2.56E+04±8.84E+03	+	3.73E+03±2.27E+03
BBOB21	2.50E+01±2.97E+01	+	5.43E+01±9.00E+00	+	6.94E+01±2.12E+01	+	**3.43E+00±1.01E+00**
BBOB22	1.36E+01±5.21E+00	+	6.76E+01±1.60E+00	+	7.63E+01±6.09E+00	+	**3.97E+00±2.23E+00**
BBOB23	4.87E+00±6.08E-01	+	5.02E+00±8.83E-01	+	5.25E+00±7.67E-01	≈	**4.08E+00±3.50E-01**
BBOB24	4.02E+02±1.43E+02	+	**2.93E+02±7.41E+00**	≈	6.81E+02±3.75E+01	+	3.34E+02±3.00E+01
+/−/≈	13/2/9		13/7/4		21/0/3		

TABLE S-V

RESULTS OF REDSADE-WOLOCAL AND REDSADE ON THE FIRST SET OF TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN REDSADE AND REDSADE-WOLOCAL.

Problem	ReDSADE-WoLocal *MFEV ± Std Dev*		ReDSADE *MFEV ± Std Dev*
S1-F1	2.05E+02 ± 1.54E+02	+	**1.08E-04±2.29E-04**
S1-F2	1.35E+02 ± 2.62E+02	+	**1.09E+00±8.32E-01**
S1-F3	4.24E−03 ± 7.61E−03	+	**3.01E-09±4.16E-09**
S1-F4	9.11E−04 ± 1.43E−03	+	**3.72E-10±8.32E-10**
S1-F5	3.27E+00 ± 3.62E+00	+	**1.81E-04±4.02E-04**
S1-F6	3.52E−04 ± 1.03E−03	+	**7.87E-09±1.30E-08**
S1-F7	1.10E−03 ± 9.17E−04	+	**1.52E-04±1.24E-04**
S1-F8	4.33E−03 ± 3.53E−03	+	**1.21E-04±1.40E-04**
S1-F9	1.88E−01 ± 2.33E−01	+	**3.16E-07±5.91E-07**
S1-F10	9.77E+01 ± 1.07E+02	+	**2.00E-02±2.96E-02**
$+/-/\approx$	10/0/0		

TABLE S-VI

RESULTS OF REDSADE-ED AND REDSADE ON THE FIRST SET OF TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN REDSADE AND REDSADE-ED.

Problem	ReDSADE-ED *MFEV ± Std Dev*		ReDSADE *MFEV ± Std Dev*
S1-F1	1.01E+00±8.50E-01	+	**1.08E-04±2.29E-04**
S1-F2	1.15E+01±2.32E+00	+	**1.09E+00±8.32E-01**
S1-F3	6.53E-03±5.11E-03	+	**3.01E-09±4.16E-09**
S1-F4	3.26E-04±3.43E-04	+	**3.72E-10±8.32E-10**
S1-F5	7.51E-02±5.32E-02	+	**1.81E-04±4.02E-04**
S1-F6	6.78E-07±1.45E-07	+	**7.87E-09±1.30E-08**
S1-F7	7.42E-04±8.06E-04	+	**1.52E-04±1.24E-04**
S1-F8	1.98E-04±3.35E-04	$\approx$	**1.21E-04±1.40E-04**
S1-F9	2.10E-02±2.11E-02	+	**3.16E-07±5.91E-07**
S1-F10	6.20E+01±7.80E+01	+	**2.00E-02±2.96E-02**
$+/-/\approx$	9/0/1		

TABLE S-VII

RESULTS OF REDSADE-ED AND REDSADE ON S1-F2 WITH DIFFERENT Δ VALUES. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN REDSADE AND REDSADE-ED.

Problem	Value of Δ	ReDSADE-ED *MFEV ± Std Dev*		ReDSADE *MFEV ± Std Dev*
S1-F2	$\Delta = 10^6$	2.67E+01 ±5.63E+01	+	**6.83E+00±8.42E+00**
	$\Delta = 10^5$	1.14E+00 ±2.27E+00	+	**6.49E-01±5.56E-01**
	$\Delta = 10^4$	1.03E+00 ±1.92E+00	+	**1.59E-01±4.34E-01**
	$\Delta = 10^3$	1.21E-01 ±9.97E-02	$\approx$	**8.02E-02±7.25E-02**
	$\Delta = 10^2$	**6.14E-03±1.09E-03**	$\approx$	7.18E-03±1.29E-02
	$\Delta = 10$	6.86E-05±1.10E-04	$\approx$	**4.45E-05±2.99E-02**

TABLE S-VIII

RESULTS OF REDSADE-MC, REDSADE-WOO, AND REDSADE ON THE FIRST SET OF TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN REDSADE AND EACH OF REDSADE-MC AND REDSADE-WOO.

Problem	ReDSADE-MC *MFEV ± Std Dev*		ReDSADE-WoO *MFEV ± Std Dev*		ReDSADE *MFEV ± Std Dev*
S1-F1	3.37E-02±5.13E-02	+	4.34E-01±5.32E-01	+	**1.08E-04±2.29E-04**
S1-F2	2.49E+01±2.61E+01	+	4.54E+00±4.72E+00	+	**1.09E+00±8.32E-01**
S1-F3	2.11E+01±2.96E+01	+	8.18E-05±8.90E-05	+	**3.01E-09±4.16E-09**
S1-F4	5.79E-07±1.23E-06	+	4.34E-05±1.95E-05	+	**3.72E-10±8.32E-10**
S1-F5	1.07E-03±9.55E-02	+	1.50E-03±9.51E-03	+	**1.81E-04±4.02E-04**
S1-F6	5.27E-02±4.28E-02	+	4.82E-08±3.24E-08	+	**7.87E-09±1.30E-08**
S1-F7	9.98E-03±9.99E-03	+	**7.53E-08±7.18E-08**	−	1.52E-04±1.24E-04
S1-F8	1.97E-03±1.96E-03	+	**3.21E-06±1.34.E-06**	−	1.21E-04±1.40E-04
S1-F9	3.98E-02±8.82E-02	+	6.16E-03±3.85E-03	+	**3.16E-07±5.91E-07**
S1-F10	1.15E+02±1.32E+02	+	**7.96E-03±5.75E-03**	≈	2.00E-02±2.96E-02
+/−/≈	10/0/0		7/2/1		

TABLE S-IX

RESULTS OF REDSADE-MC, REDSADE-WOO, AND REDSADE ON THE 10-DIMENSIONAL BBOB TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN REDSADE AND EACH OF REDSADE-MC AND REDSADE-WOO.

Problem	ReDSADE-MC *MFEV ± Std Dev*		ReDSADE-WoO *MFEV ± Std Dev*		ReDSADE *MFEV ± Std Dev*
BBOB1	9.97E-08±1.51E-08	+	1.26E-02±1.40E-02	+	**3.33E-08±3.36E-08**
BBOB2	**3.96E+02±1.41E+01**	≈	1.89E+04±3.52E+04	+	4.63E+02±4.79E+02
BBOB3	8.61E+02±1.58E+01	+	8.29E+02±2.61E+01	+	**4.29E+01±6.23E+00**
BBOB4	8.15E+02±2.49E+01	+	7.90E+02±3.31E+01	+	**8.37E+01±1.34E+01**
BBOB5	**1.00E-08±0.00E+00**	≈	1.83E+01±2.78E-01	+	**1.00E-08±0.00E+00**
BBOB6	8.83E+01±1.72E+01	≈	1.85E+02±6.62E+01	+	**6.15E+01±8.23E+00**
BBOB7	1.39E+01±1.12E+01	+	2.95E+01±1.57E+01	+	**7.81E+00±3.09E+00**
BBOB8	**1.57E+01±1.41E+01**	≈	1.83E+02±2.68E+02	+	1.82E+01±1.76E+01
BBOB9	4.21E+01±6.63E+01	+	2.77E+02±2.22E+02	+	**8.85E+00±1.69E+00**
BBOB10	4.82E+04±2.22E+04	+	9.10E+04±1.30E+05	+	**2.13E+04±7.84E+03**
BBOB11	2.06E+02±1.05E+02	+	1.59E+02±9.37E+01	+	**9.95E+01±2.66E+01**
BBOB12	2.37E+03±2.61E+03	+	2.68E+04±3.40E+04	+	**1.67E+01±1.12E+01**
BBOB13	6.14E+00±6.11E+00	+	4.80E+01±5.64E+01	+	**7.92E-01±8.13E-01**
BBOB14	1.05E+02±1.52E-02	+	1.05E+02±8.36E-02	+	**3.41E-03±1.24E-03**
BBOB15	9.23E+01±8.87E+00	+	1.20E+02±8.75E+00	+	**4.83E+01±1.81E+01**
BBOB16	**5.43E+00±3.37E+00**	−	1.41E+01±2.52E+00	≈	1.23E+01±2.23E+00
BBOB17	3.03E+01±1.61E+00	≈	3.03E+01±1.63E+00	+	**2.18E+00±2.91E-01**
BBOB18	**1.95E+01±2.35E+00**	≈	2.36E+01±4.75E+00	≈	2.10E+01±2.22E+00
BBOB19	2.00E+02±9.87E-01	+	1.96E+02±1.67E+00	+	**4.84E+00±6.95E-01**
BBOB20	1.09E+03±2.20E-01	+	1.09E+03±4.83E-01	+	**2.81E+00±4.00E-01**
BBOB21	**4.81E+00±5.31E+00**	≈	1.10E+01±1.48E+01	+	4.99E+00±2.88E+00
BBOB22	1.99E+03±8.72E+00	+	1.98E+03±1.46E+01	+	**4.31E+00±2.68E+00**
BBOB23	2.75E+00±6.71E-01	≈	2.90E+00±7.07E-01	+	**2.42E+00±5.11E-01**
BBOB24	6.45E+01±4.89E+00	+	1.15E+02±2.54E+01	+	**4.49E+01±1.21E+01**
+/−/≈	15/1/8		22/0/2		

TABLE S-X
RESULTS OF REDSADE-KM AND REDSADE ON THE FIRST SET OF TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN REDSADE AND REDSADE-KM.

Problem	ReDSADE-KM *MFEV ± Std Dev*		ReDSADE *MFEV ± Std Dev*
S1-F1	9.24E+01±7.77E+01	+	**1.08E-04±2.29E-04**
S1-F2	1.39E+01±2.09E+00	+	**1.09E+00±8.32E-01**
S1-F3	**2.59E-09±5.19E-09**	≈	3.01E-09±4.16E-09
S1-F4	1.41E-09±7.73E-09	≈	**3.72E-10±8.32E-10**
S1-F5	**1.70E-04±3.30E-04**	≈	1.81E-04±4.02E-04
S1-F6	**1.02E-09±3.11E-09**	≈	7.87E-09±1.30E-08
S1-F7	2.21E-04±1.43E-04	≈	**1.52E-04±1.24E-04**
S1-F8	**1.63E-05±2.29E-05**	≈	1.21E-04±1.40E-04
S1-F9	2.83E-06±4.70E-06	≈	**3.16E-07±5.91E-07**
S1-F10	**5.02E-06±2.10E-06**	−	2.00E-02±2.96E-02
$+/-/\approx$	$2/1/7$		

TABLE S-XI
RUNTIME (IN SECOND) CONSUMED BY EGO, CAL-SAPSO, GLOSADE, AND REDSADE ON THE SECOND SET OF TEST PROBLEMS.

Problem	EGO	CAL-SAPSO	GLoSADE	ReDSADE	Problem	EGO	CAL-SAPSO	GLoSADE	ReDSADE
	10 Dimension					30 Dimension			
S2-F1	1096.83	886.40	78.27	539.45	S2-F1	3212.19	2717.56	107.45	1754.02
S2-F2	979.42	809.77	77.66	516.50	S2-F2	3098.74	2528.80	107.63	1997.82
S2-F3	1108.88	938.34	78.06	569.53	S2-F3	3121.22	2684.04	107.37	2425.80
S2-F4	804.32	660.92	77.27	715.98	S2-F4	2787.87	2087.49	105.78	2378.99
S2-F5	995.39	810.89	78.68	710.68	S2-F5	2765.12	2057.90	103.87	2097.76
S2-F6	812.41	664.38	78.75	586.05	S2-F6	2798.47	2098.96	103.89	1807.56
S2-F7	1398.39	1153.25	78.33	716.46	S2-F7	3121.04	2628.49	105.97	1919.84
S2-F8	1386.08	1117.58	78.79	684.75	S2-F8	2686.35	1981.13	105.63	1649.37
S2-F9	1388.55	1116.59	79.25	708.52	S2-F9	2989.02	2465.47	105.51	1797.10
S2-F10	1242.70	1075.88	79.53	900.48	S2-F10	3102.90	2604.76	105.62	1939.58

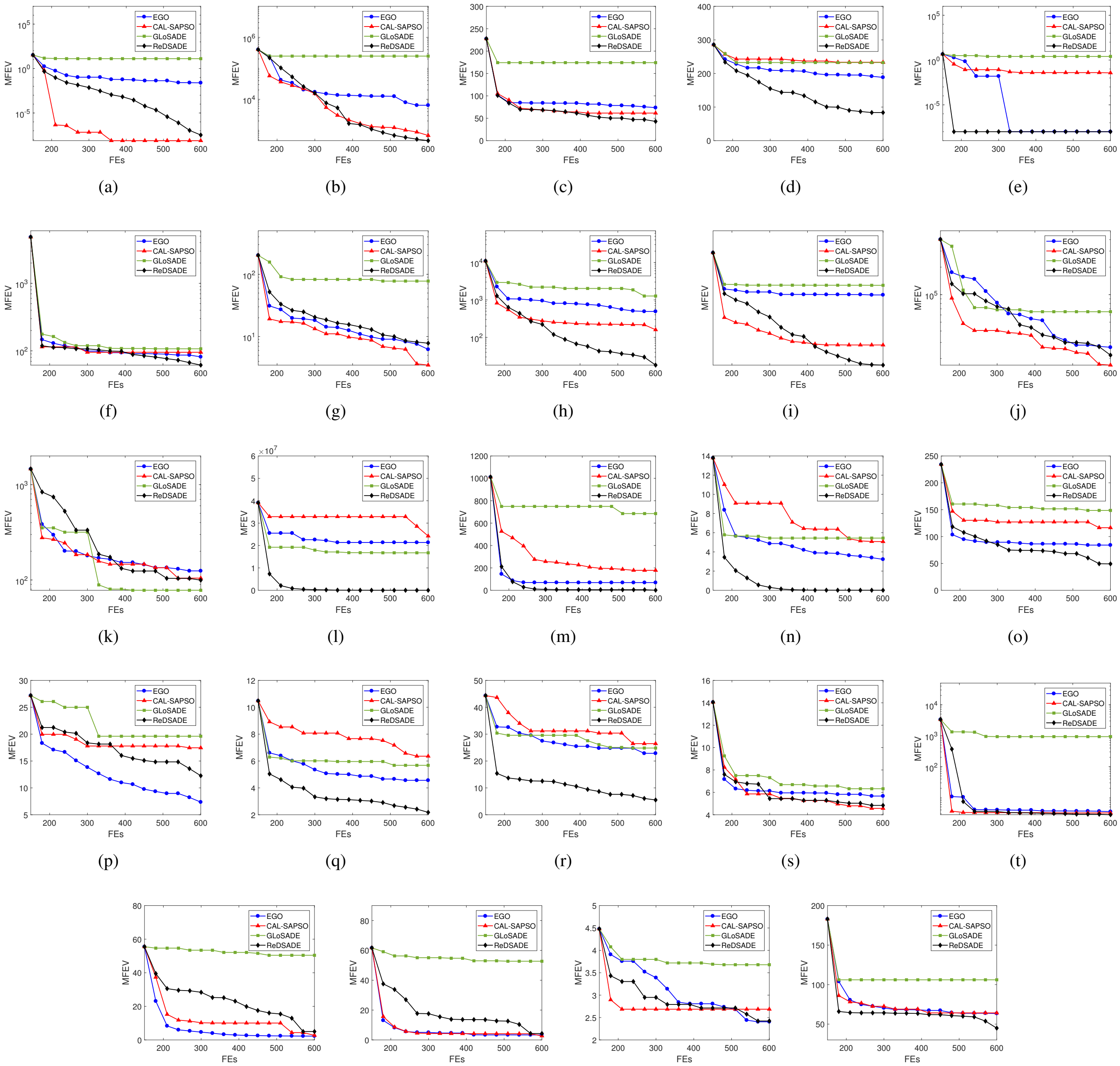


Fig. S-1. Evolution of *MFEV* provided by the four compared algorithms on 10-dimensional BBOB test problems over 20 independent runs. (a) BBOB1. (b) BBOB2. (c) BBOB3. (d) BBOB4. (e) BBOB5. (f) BBOB6. (g) BBOB7. (h) BBOB8. (i) BBOB9. (j) BBOB10. (k) BBOB11. (l) BBOB12. (m) BBOB13. (n) BBOB14. (o) BBOB15. (p) BBOB16. (q) BBOB17. (r) BBOB18. (s) BBOB19. (t) BBOB20. (u) BBOB21. (v) BBOB22. (w) BBOB23. (x) BBOB24.

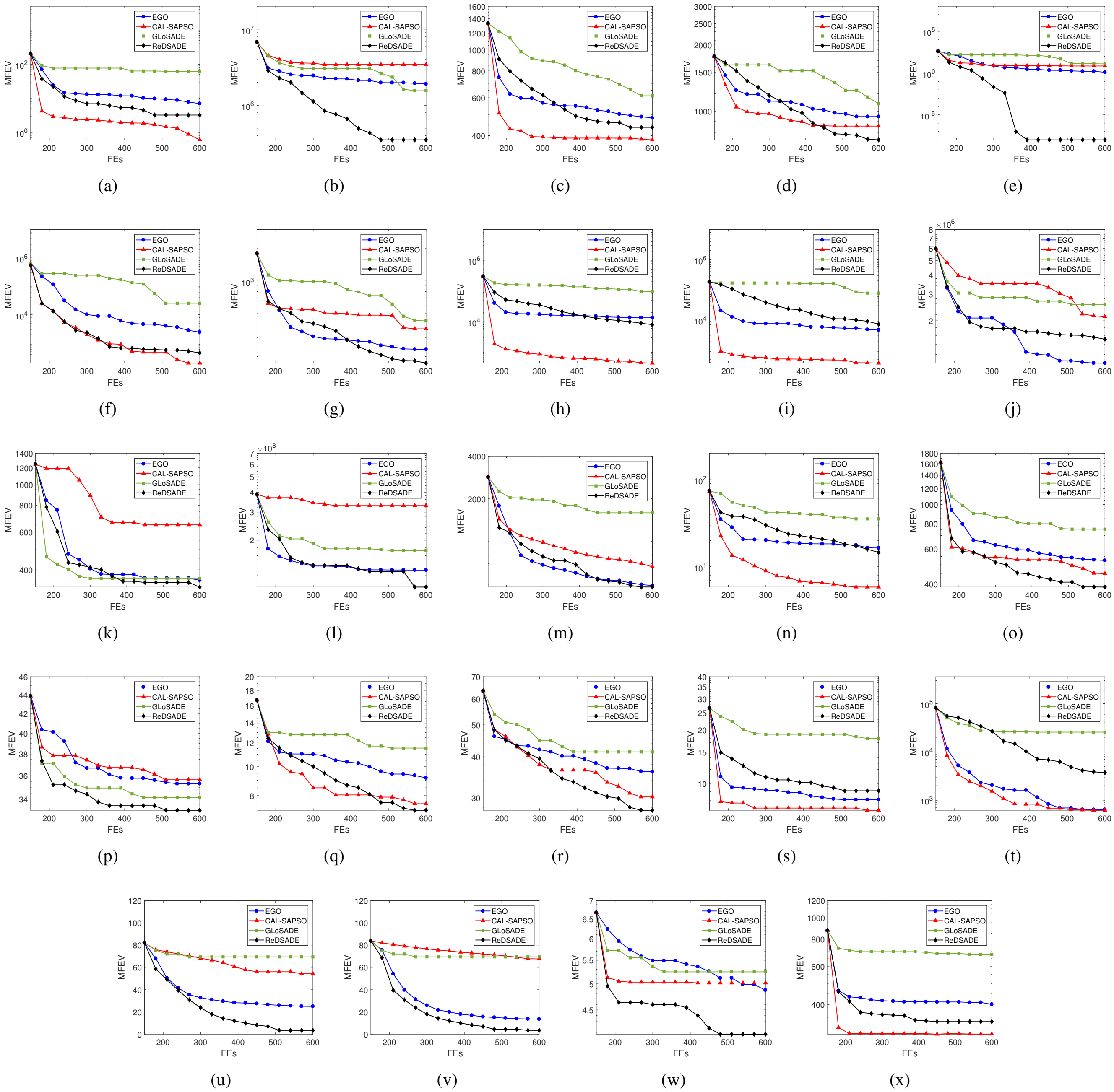
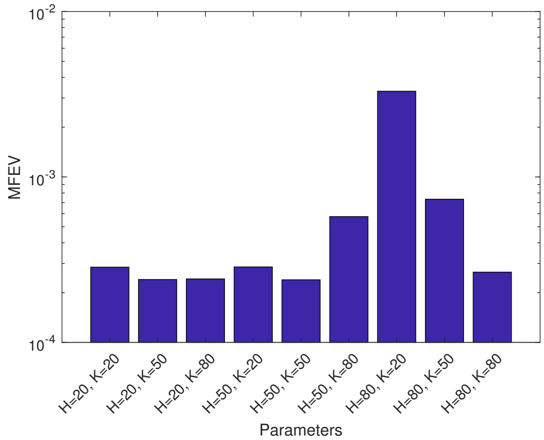
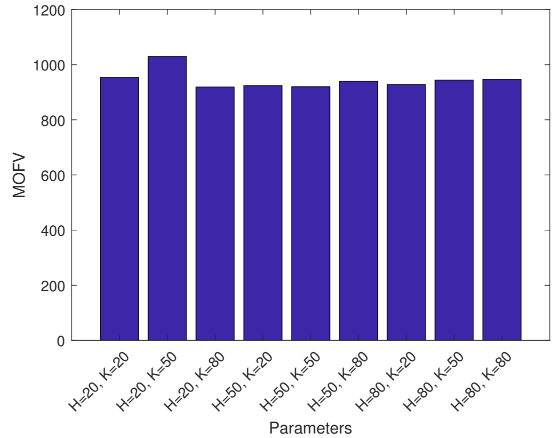


Fig. S-2. Evolution of *MFEV* provided by the four compared algorithms on 30-dimensional BBOB test problems over 20 independent runs. (a) BBOB1. (b) BBOB2. (c) BBOB3. (d) BBOB4. (e) BBOB5. (f) BBOB6. (g) BBOB7. (h) BBOB8. (i) BBOB9. (j) BBOB10. (k) BBOB11. (l) BBOB12. (m) BBOB13. (n) BBOB14. (o) BBOB15. (p) BBOB16. (q) BBOB17. (r) BBOB18. (s) BBOB19. (t) BBOB20. (u) BBOB21. (v) BBOB22. (w) BBOB23. (x) BBOB24.



(a)



(b)

Fig. S-3. Influence of the parameter settings. (a) S1-F5. (b) S2-F8.

S-III. Test Problems

A. The First Set of Test Problems

$$\textbf{S1-F1}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{01} + \Delta, & x_1 > -35 \ \& \ x_2 > 59 \\ CEC2005_{01}, & \text{otherwise} \end{cases} \\ x_1, \ldots, x_8 \in [-100, 100] \tag{S-1}$$

where $CEC2005_{01}$ is the first benchmark problem of IEEE CEC2005 [?], $\Delta = 10^4$, the optimal solution is $(-39.3119, 58.8999, -46.3224, -74.6515, -16.7997, -80.5441, -10.5935, 24.9694)$, and the optimal objective function value is -450.

$$\textbf{S1-F2}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{03} + \Delta, & x_1 > -33 \ \& \ x_2 > 66.5 \\ CEC2005_{03}, & \text{otherwise} \end{cases} \\ x_1, x_2 \in [-100, 100] \tag{S-2}$$

where $CEC2005_{03}$ is the third benchmark problem of IEEE CEC2005, $\Delta = 10^7$, the optimal solution is $(-32.2013, 64.9776)$, and the optimal objective function value is -450.

$$\textbf{S1-F3}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{01}, & x_1 > -40 \ \& \ x_2 > 59 \\ CEC2005_{02} + \Delta, & \text{otherwise} \end{cases} \\ x_1, x_2 \in [-100, 100] \tag{S-3}$$

where $CEC2005_{01}$ and $CEC2005_{02}$ are the first and second benchmark problems of IEEE CEC2005, respectively, $\Delta = 10^4$, the optimal solution is $(-39.3119, 58.8999)$, and the optimal objective function value is -450.

$$\textbf{S1-F4}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{02}, & x_1 < 36 \\ CEC2005_{01} + \Delta, & \text{otherwise} \end{cases} \\ x_1, \ldots, x_4 \in [-100, 100] \tag{S-4}$$

where $CEC2005_{01}$ and $CEC2005_{02}$ are the first and second benchmark problems of IEEE CEC2005, respectively, $\Delta = 10^4$, the optimal solution is $(35.6267, -82.9123, -10.6423, -83.5815)$, and the optimal objective function value is -450.

$$\textbf{S1-F5}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{02} + \Delta, & x_1 > 36 \ \& \ x_2 < -11 \\ CEC2005_{02}, & \text{otherwise} \end{cases} \\ x_1, \ldots, x_4 \in [-100, 100] \tag{S-5}$$

where $CEC2005_{02}$ is the second benchmark problem of IEEE CEC2005, $\Delta = 10^4$, the optimal solution is $(35.6267, -82.9123, -10.6423, -83.5815)$, and the optimal objective function value is -450.

$$\textbf{S1-F6}: \ \min: \ f(\mathbf{x}) = \begin{cases} \sum_{i=1}^{4}(|x_i|^{i+1}), & (x_1 - 1)^2 + (x_4 - 1) \leq 2 \\ \sum_{i=1}^{4}(|x_i|^{i+1}) + \Delta, & \text{otherwise} \end{cases} \\ x_1, \ldots, x_4 \in [-1, 1] \tag{S-6}$$

where $\Delta = 15$, the optimal solution is $(0, 0, 0, 0)$, and the optimal objective function value is 0.

$$\textbf{S1-F7}: \ \min: \ f(\mathbf{x}) = \begin{cases} \sum_{i=1}^{6}(|x_i|^{i+1}), & (x_1 - 1)^2 + (x_2 - 1) \leq 2 \ \& \ (x_3 - 1)^2 + (x_4 - 1) \leq 2 \\ \sum_{i=1}^{6}(|x_i|^{i+1}) + \Delta, & \text{otherwise} \end{cases} \\ x_1, \ldots, x_6 \in [-1, 1] \tag{S-7}$$

where $\Delta = 15$, the optimal solution is $(0, 0, 0, 0, 0, 0)$, and the optimal objective function value is 0.

$$\textbf{S1-F8}: \ \min: \ f(\mathbf{x}) = \begin{cases} \sum_{i=1}^{8}(|x_i|^{i+1}), & x_1 \leq 0 \\ \sum_{i=1}^{8}(|x_i|^{i+1}) + \Delta, & \text{otherwise} \end{cases} \\ x_1, \ldots, x_8 \in [-1, 1] \tag{S-8}$$

where $\Delta = 20$, the optimal solution is $(0,0,0,0,0,0,0,0)$, and the optimal objective function value is 0.

$$\textbf{S1-F9}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{01}, & x_1 > 15 \ \& \ x_2 > 30 \\ CEC2005_{01}, & x_1 < -35 \ \& \ x_2 < 60 \\ CEC2005_{01} + \Delta, & \text{otherwise} \end{cases} \quad x_1,\ldots,x_4 \in [-100,100] \tag{S-9}$$

where $CEC2005_{02}$ is the second benchmark problem of IEEE CEC2005, $\Delta = 10^5$, the optimal solution is $(-39.3119, 58.8999, -46.3224, -74.6515)$, and the optimal objective function value is -450.

$$\textbf{S1-F10}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{02} + \Delta_1, & x_1 < -50 \ \& \ x_2 > 50 \\ CEC2005_{02}, & x_1 > 34 \ \& \ x_2 < -80 \\ CEC2005_{02} + \Delta_2, & \text{otherwise} \end{cases} \quad x_1,\ldots,x_6 \in [-100,100] \tag{S-10}$$

where $CEC2005_{01}$ is the first benchmark problem of IEEE CEC2005, $\Delta_1 = 10^5$, $\Delta_2 = 2 \times 10^5$, the optimal solution is $(35.6267, -82.9123, -10.6423, -83.5815, 83.1552, 47.048)$, and the optimal objective function value is -450.

B. The Second Set of Test Problems

$$\begin{aligned} &\textbf{S2-F1}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{15}, & g(x) > 0 \\ CEC2005_{15} + \Delta, & \text{otherwise} \end{cases} \\ &g(x) = \sum_{i=1}^{D} x_i \\ &x_i \in [-5,5], \qquad i = 1,\ldots,D \end{aligned} \tag{S-11}$$

where $CEC2005_{15}$ is the 15th benchmark problem of IEEE CEC2005 [**?**] and $\Delta = 300$.

$$\begin{aligned} &\textbf{S2-F2}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{16}, & g(x) > 0 \\ CEC2005_{16} + \Delta, & \text{otherwise} \end{cases} \\ &g(x) = \sum_{i=1}^{D} x_i \\ &x_i \in [-5,5], \qquad i = 1,\ldots,D \end{aligned} \tag{S-12}$$

where $CEC2005_{16}$ is the 16th benchmark problem of IEEE CEC2005 and $\Delta = 600$.

$$\begin{aligned} &\textbf{S2-F3}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{17}, & g(x) > 0 \\ CEC2005_{17} + \Delta, & \text{otherwise} \end{cases} \\ &g(x) = \sum_{i=1}^{D} x_i \\ &x_i \in [-5,5], \qquad i = 1,\ldots,D \end{aligned} \tag{S-13}$$

where $CEC2005_{17}$ is the 17th benchmark problems of IEEE CEC2005 and $\Delta = 900$.

$$\begin{aligned} &\textbf{S2-F4}: \ \min: \ f(\mathbf{x}) = \begin{cases} CEC2005_{18}, & g(x) > 9D \\ CEC2005_{18} + \Delta, & \text{otherwise} \end{cases} \\ &g(x) = \sum_{i=1}^{D} x_i^2 \\ &x_i \in [-5,5], \qquad i = 1,\ldots,D \end{aligned} \tag{S-14}$$

where $CEC2005_{18}$ is the 18th benchmark problems of IEEE CEC2005 and $\Delta = 300$.

$$\begin{aligned}
\textbf{S2-F5}: \ \min: \ & f(\mathbf{x}) = \begin{cases} CEC2005_{19}, & g(x) > 9D \\ CEC2005_{19} + \Delta, & \text{otherwise} \end{cases} \\
& g(x) = \sum_{i=1}^{D} x_i^2 \\
& x_i \in [-5, 5], \qquad i = 1, \ldots, D
\end{aligned} \tag{S-15}$$

where $CEC2005_{19}$ is the 19th benchmark problem of IEEE CEC2005 and $\Delta = 700$.

$$\begin{aligned}
\textbf{S2-F6}: \ \min: \ & f(\mathbf{x}) = \begin{cases} CEC2005_{20}, & g(x) > 9D \\ CEC2005_{20} + \Delta, & \text{otherwise} \end{cases} \\
& g(x) = \sum_{i=1}^{D} x_i^2 \\
& x_i \in [-5, 5], \qquad i = 1, \ldots, D
\end{aligned} \tag{S-16}$$

where $CEC2005_{20}$ is the 20th benchmark problem of IEEE CEC2005 and $\Delta = 1500$.

$$\begin{aligned}
\textbf{S2-F7}: \ \min: \ & f(\mathbf{x}) = \begin{cases} CEC2005_{21}, & g(x) > 12000D \\ CEC2005_{21} + \Delta, & \text{otherwise} \end{cases} \\
& g(x) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \\
& x_i \in [-5, 5], \qquad i = 1, \ldots, D
\end{aligned} \tag{S-17}$$

where $CEC2005_{21}$ is the 21st benchmark problem of IEEE CEC2005 and $\Delta = 200$.

$$\begin{aligned}
\textbf{S2-F8}: \ \min: \ & f(\mathbf{x}) = \begin{cases} CEC2005_{22}, & g(x) > 12000D \\ CEC2005_{22} + \Delta, & \text{otherwise} \end{cases} \\
& g(x) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \\
& x_i \in [-5, 5], \qquad i = 1, \ldots, D
\end{aligned} \tag{S-18}$$

where $CEC2005_{22}$ is the 22nd benchmark problem of IEEE CEC2005 and $\Delta = 900$.

$$\begin{aligned}
\textbf{S2-F9}: \ \min: \ & f(\mathbf{x}) = \begin{cases} CEC2005_{23}, & g(x) > 120000D \\ CEC2005_{23} + \Delta, & \text{otherwise} \end{cases} \\
& g(x) = \sum_{i=1}^{D-1} \left[1 + \frac{\left[100 \times (x_i^2 - x_{i+1})^2 + (1 - x_i)^2\right]^2}{4000} - \cos[100 \times (x_i^2 - x_{i+1})^2 + (1 - x_i)^2]\right] \\
& \quad + \left[1 + \frac{\left[100 \times (x_D^2 - x_1)^2 + (1 - x_D)^2\right]^2}{4000} - \cos[100 \times (x_D^2 - x_1)^2 + (1 - x_D)^2]\right] \\
& x_i \in [-5, 5], \qquad i = 1, \ldots, D
\end{aligned} \tag{S-19}$$

where $CEC2005_{23}$ is the 23rd benchmark problem of IEEE CEC2005 and $\Delta = 200$.

$$\begin{aligned}
\textbf{S2-F10}: \ \min: \ & f(\mathbf{x}) = \begin{cases} CEC2005_{24}, & g(x) > 120000D \\ CEC2005_{24} + \Delta, & \text{otherwise} \end{cases} \\
& g(x) = \sum_{i=1}^{D-1} \left[1 + \frac{\left[100 \times (x_i^2 - x_{i+1})^2 + (1 - x_i)^2\right]^2}{4000} - \cos[100 \times (x_i^2 - x_{i+1})^2 + (1 - x_i)^2]\right] \\
& \quad + \left[1 + \frac{\left[100 \times (x_D^2 - x_1)^2 + (1 - x_D)^2\right]^2}{4000} - \cos[100 \times (x_D^2 - x_1)^2 + (1 - x_D)^2]\right] \\
& x_i \in [-5, 5], \qquad i = 1, \ldots, D
\end{aligned} \tag{S-20}$$

where $CEC2005_{24}$ is the 24th benchmark problem of IEEE CEC2005 and $\Delta = 300$.

C. The Third Set of Test Problems

The third set of test problems is the BBOB test suite, which is directly taken from [1].

References

[1] N. Hansen, A. Auger, S. Finck, and R. Ros, "Real-parameter black-box optimization benchmarking: Experimental setup," *Orsay, France: Université Paris Sud, Institut National de Recherche en Informatique et en Automatique (INRIA) Futurs, Équipe TAO, Tech. Rep*, 2014.