A Surrogate-Assisted Evolutionary Algorithm for Seeking Multiple Solutions of Expensive Multimodal Optimization Problems

Jing-Yu Ji, Zusheng Tan, Sanyou Zeng, Eric W. K. See-To, and Man-Leung Wong, *Member, IEEE*

***Abstract*—Surrogate-assisted evolutionary algorithms for expensive optimization problems have gained considerable attention in recent years. In many real-world optimization problems, we may face expensive optimization problems with multiple optimal solutions. Locating multiple optima for such expensive problems is qualitatively challenging. This study proposes a surrogate-assisted differential evolution based on region decomposition to seek multiple optima for expensive multimodal optimization problems. In this study, we have designed three major components: 1) the adaptive region decomposition, 2) the multilayer perceptron-based global surrogate, and 3) the self-adaptive gradient descent-based local search. First, the improved adaptive region decomposition detects promising subregions at the beginning phase, and continuously discards inferior subregions successively. Second, the multilayer perceptron-based surrogate and self-adaptive gradient-based mutation work in a collaborative manner on distinct sub-populations to seek multiple optimal solutions. Overall, an attempt has been made to solve expensive multimodal optimization problems. Systematic experiments on 20 test functions show the encouraging and promising performance of our proposed approach.**

***Index Terms*—Surrogate-assisted evolutionary algorithm, expensive multimodal optimization, region decomposition, multilayer perceptron, self-adaptive gradient-based local search.**

I. Introduction

OPTIMIZATION problems that need costly physical experiments to evaluate candidate solutions are known as expensive optimization problems (EOPs) [1]. Classic evolutionary algorithms (EAs) would lose feasibility and effectiveness for such problems since they usually require hundreds of thousands of exact function evaluations (FEs) [2], [3], [4], [5]. For expensive optimization, such a significant number of exact function evaluations (FEs) is usually unaffordable. To bridge the gap between cheap and expensive optimizations, a wide variety of surrogate-assisted EAs (SAEAs) [6], [7], [8], [9] have been proposed and developed to tackle EOPs. Compared with using expensive exact FEs, SAEAs prescreen candidate solutions through surrogate models which are cheap approximations of the fitness function. Frequently adopted surrogate models include polynomial regression [10], Gaussian process (GP) [11], [12], support vector machine (SVM) [13], [14], and radial basis function (RBF) [15], [16]. Recently, some hybrid surrogates and the ensemble of surrogates have also been developed for different benefits. Since building surrogate models require much less computational overhead cost, expensive exact FEs can be lessened. Therefore, with the assistance of surrogate models, SAEAs can evolve population and finally obtain the near-optimal solution of EOPs under the limited exact FE budget.

Commonly, the design of SAEAs is based on the traditional EA of search engines and the surrogate of fitness approximation. Different from inexpensive optimization problems, EAs are often naturally devised with surrogate models for population evolution. Related work has been associated with evolutionary operators from various EAs, such as genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO). For example, a linkage-learning-based GA with surrogate assistance was presented to solve a set of pseudo-Boolean benchmark functions [17]. Different from using a surrogate model as an approximation of a fitness function, the proposed GA is used to directly optimize the correctly learned surrogate which perfectly matches the exact fitness function. In [18], a surrogate-assisted differential evolution was devised to solve expensive constrained optimization problems involving mixed-integer variables, and a modified mutation strategy of DE is used to balance the local and global characteristics of the candidate solutions which are then prescreened by surrogate models. Moreover, a classifier-assisted level-based learning swarm optimizer was designed to solve large-scale EOPs in [19], where better candidate solutions are classified rather than evaluated by the surrogate model.

Surrogates [20] can be roughly categorized into two types: 1) global surrogate models and 2) local surrogate models. The former aim to approximate the landscape of a given problem, and while the latter are aimed at achieving a high approximation accuracy within a limited decision region. For example, Yang, et al. [21] utilized a global surrogate built by all the samples in a database to obtain the most potential offspring solution. A surrogate-guided DE approach, proposed by Cai et al. [22], uses the optima predicted by the global and local surrogates to

Manuscript received 3 December 2022; revised 29 April 2023; accepted 22 June 2023. Date of publication 15 August 2023; date of current version 23 January 2024. This work was supported by the LEO Dr. David P. Chan Institute of Data Science. *(Corresponding author: Sanyou Zeng.)*

Jing-Yu Ji, Zusheng Tan, Eric W. K. See-To, and Man-Leung Wong are with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong (e-mail: im@jijingyu.com; allentan@ln.hk; ericseeto@ln.edu.hk; mlwong@acm.org).

Sanyou Zeng is with the School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China (e-mail: sanyouzeng@qq.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TETCI.2023.3301794, provided by the authors.

Digital Object Identifier 10.1109/TETCI.2023.3301794

guide the mutation direction of the DE algorithm, and thus the proposed approach can achieve rapid converge. Liu et al. [23] have designed two RBF models. One is to guide the global learning strategy, and the other one is for the trust region local search. Afterward, an improved version based on the decision space partition is proposed in [24]. Kůdela et al. [25] developed a novel Lipschitz-based surrogate that combines a standard global RBF surrogate and a local optimization procedure to enhance the search capability. In addition, some surrogates can be further used in the prescreening strategies. For example, Cai et al. [26] proposed a prescreening strategy based on the expected improvement infilling criterion of a simplified GP to obtain promising candidate offspring produced by the basic GA, and Fu et al. [27] utilized the same criterion to balance the potential and uncertainty of the candidate solutions for the optimization of antenna synthesis.

So far, SAEAs have attracted much attention to many kinds of expensive optimization, such as high dimensional EOPs [16], [28], [29], large-scale EOPs [30], [31], multiobjective EOPs [32], [33], [34], and constrained EOPs [35], [36], [37]. Compared with the aforementioned EOPs, few attempts have been made to solve expensive multimodal optimization problems (EMMOPs) characterized by multiple global solutions [38]. The mathematical model of such a problem can be formulated as follows:

$$\begin{aligned} &\text{minimize} f(\mathbf{x}) \\ &\text{subeject to} \mathbf{x} \in \Re \end{aligned} \tag{1}$$

where $f(\mathbf{x})$ is the objective function, $\mathbf{x} = (x_1, \ldots, x_D)$ denotes a decision vector with D variables, and $\Re$ is the decision space defined as

$$\Re = \prod_{i=1}^{D} [LB_i, UB_i] \tag{2}$$

where LB_i and UB_i are the lower and upper bounds of x_i, respectively. Different from solving EOPs which have a similar formulation but only one optimal solution, it is very preferable for EMMOPs to seek as many optimal solutions as possible, such that decision-makers can be well informed to make a final decision.

EMMOPs commonly arise in scientific and engineering optimization. For example, the design of antenna arrays can be transformed into solving an EMMOP [27], [39]. In this case, evaluating a candidate solution through physical simulation is extremely expensive and requires the use of precious metals and alloys. In contrast, one virtual simulation for estimation is cheap, which can just be achieved by a high-performance computer. Recently, three approaches have been proposed to solve EMMOPs, namely DSCPSO-EMM [38], MaMPSO [40] and D/REM [2]. DSCPSO-EMM adopts a modal-guided dual-layer cooperative surrogate model to assist PSO with the purpose of reducing evaluation costs. And a clustering and peak-valley-based hybrid strategy is proposed to detect new modalities. MaMPSO uses a multi-surrogate-assisted multitasking PSO to solve EMMOPs. The authors employ various surrogate models to transform an EMMOP into a multitasking optimization problem, and then design a multitasking niche particle swarm algorithm to solve the transformed problem. D/REM locates the multiple global solutions in two stages, wherein promising regions are first detected and then local searches are performed in these identified areas.

In the presence of multiple optima, solving EMMOPs is quantitatively challenging for SAEAs [41]. Generally, there are two major difficulties. First, SAEAs are desirable to locate multiple optimal solutions rather than one exclusively optimal solution. Thus, the multimodality where different optima have identical objective values requires special treatment. Second, for surrogate models themselves, the global regression ability is required to approximate multiple attraction basins simultaneously. Since surrogate models are often used to prescreen new candidate solutions, population convergence and diversity can be directly biased at the surrogate level. However, fitting the quantitatively complex landscape of EMMOPs is not a trivial work. If a surrogate has poor generalization, population diversity can be impaired, which leads the population to converge prematurely to one or two dominant solutions while neglecting others. In addition, EMMOPs also require high-precision solutions. Therefore, the fitting function of surrogate models is expected to be closely aligned with the training data. In this case, surrogate models for the objective function of EMMOPs are prone to over-fitting due to the quantitatively global and local optima. It is well known that unreliable predictions of new data would frequently occur in such a situation [42], which makes the model useless. As a result, an over-fitting surrogate model does not assist EAs, but directly impairs the population evolution.

Based on the above considerations, this study proposes a decomposition-based SAEA, which is termed DSADE, to seek multiple optimal solutions for EMMOPs. Specifically, the main components of DSADE are: 1) the adaptive region decomposition (ARD); 2) the multilayer perceptron-based global surrogate (MLPGS); and 3) the self-adaptive gradient descent-based local search (SaGDLS). Accordingly, the contributions of this article are threefold.

1) In this study, we have developed the ARD, an improved version of density clustering [43], [44] to handle multimodality. ARD adaptively decomposes the decision space into multiple subregions and allocates an independent population for each subregion to explore its potential optimal solutions. To avoid wasting exact FEs, once a subregion has been detected that its corresponding population fails to search for the optimal solution or it has been explored by the other sub-populations, such a subregion will be discarded to save the expensive cost of exact FEs.
2) A MLPGS is proposed to evolve each sub-population rather than prescreen offspring. If the objective function is learned properly, the obtained solutions by optimizing the surrogate can be close to the optimal solutions of the original objective function. To this end, an attempt has been made to simultaneously approximate the multiple attraction basins of EMMOPs by the multilayer perceptron (MLP) [45]. Theoretically, MLP with a back-propagation training technique is better to avoid over-fitting and to keep good generalization for complex fitness landscapes due to

its characteristics of variable lengths of hidden layers and neurons. However, the MLP-based surrogate has attracted less attention than the mainstream surrogate models, such as RBF, GP, and SVM.

3) A SaGDLS is developed to obtain highly precise candidate solutions required by EMMOPs. By using gradient information, a fast convergence speed can be achieved. Meanwhile, we utilize the coexistence of multiple optimal solutions to adaptively calculate the increment instead of setting step size for gradient descent (GD). Therefore, the parameter sensitivity issue of GD can be largely alleviated.

In the experimental study, the performance of DSADE is evaluated on 20 EMMOPs to demonstrate the feasibility of DSADE. In terms of expensive optimization, DSADE is compared with five state-of-the-art algorithms to demonstrate satisfactory performance and high efficiency.

The remainder of this article is organized as follows. Section II presents preliminaries. Section III systematically elaborates on our proposed approach DSADE. Empirical studies are reported in Section IV. Section V concludes this article.

II. Preliminaries

As the basis of DSADE, MLP and the standard GD algorithm are briefly introduced in this section.

A. MLP

MLP is a multilayered feed-forward neural network and utilizes back-propagation [46] for training and deep learning, which is highly capable of fitting complex non-linear functions. An MLP has three types of layers which are the input, the hidden, and the output layers. Each layer has its elementary nodes. Especially, the nodes with nonlinear activation functions in hidden and output layers are called neurons. To construct an MLP, user-defined hyperparameters are required, such as the number of hidden layers, learning rate, and activation function. Besides hyperparameters, MLP also contains learnable parameters, namely weights and bias, which transform received data within hidden layers. The learning progress to obtain the most suitable values for hyper-parameters is called neural network training. Concerning MLP, the most popular optimization method is a supervised learning technique called back-propagation.

A simple example is given in Fig. 1. In this case, the input layer first receives a decision vector with four variables. Then, two hidden layers, each with eight neurons, apply the given activation functions to their received values. Finally, the output layer sums all connections from hidden layers and returns the predicted objective function value corresponding to the input decision vector. An MLP with more hidden layers and neurons are usually constructed to fit more complex problems. Correspondingly, the deep learning technique with sophisticated approaches is introduced to train a large neural network.

B. Gradient Descent

GD-based optimization algorithm iteratively uses the first-order derivative information to find the minimum solution. Due to its efficacy, efficiency, and simplicity, a wide variety of developed GD-based optimizers [47], [48], [49] are commonly used in the areas of numerical optimization, control system, machine learning, and deep learning.

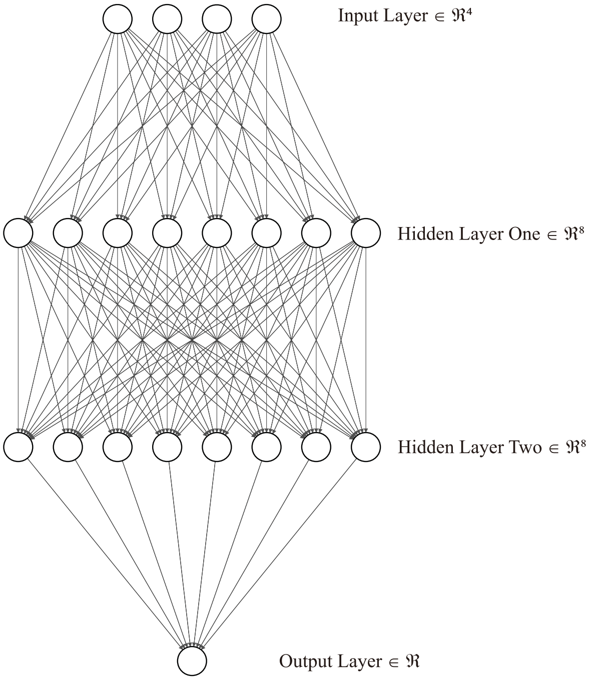

Fig. 1. Illustration of a simple feed-forward MLP with only two hidden layers. The large circle represents an artificial neuron with a certain kind of activation functions and the arc is associated with a weight at each neuron.

The gradient of a differentiable function $f(\mathbf{x})$ at a given D-dimensional vector $\mathbf{v} = (v_1, \ldots, v_D)$ is defined as follows:

$$\nabla f(\mathbf{v}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{v}) \\ \vdots \\ \frac{\partial f}{\partial x_D}(\mathbf{v}) \end{bmatrix} \tag{3}$$

where ∂ and ∇ denotes a partial derivative and the vector differential operator, respectively. $\nabla f(\mathbf{x})$ indicates the direction and the rate of the fastest increase of function $f(\mathbf{x})$, and therefore, the opposite direction of the gradient, defined as the negative of the gradient $-\nabla f(\mathbf{x})$, is the direction of the steepest descent. In numerical optimization, if the objective function is provided by a black-box form, the gradient can be estimated by the linear approximation in which the first-order partial derivative at the ith variable of $\mathbf{v}$ can be numerically calculated [50] as follows:

$$\frac{\partial f}{\partial x_i}(\mathbf{v}) = \frac{f(v_1, \ldots, v_i + \Delta x, \ldots, v_D) - f(v_1, \ldots, v_i, \ldots, v_D)}{\Delta x} \tag{4}$$

where Δx converges to 0.

GD-based optimization algorithm minimizes the objective function by iteratively calculating the next decision vector as follows:

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \eta \nabla f(\mathbf{v}_n) \tag{5}$$

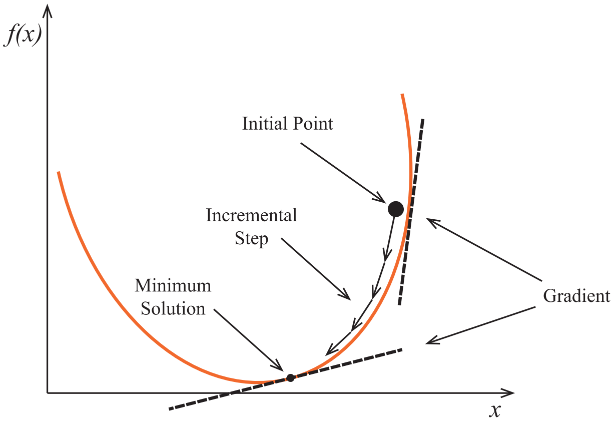

Fig. 2. Illustration of the standard GD algorithm. The black point is an initial point. Stepping in the opposite direction of gradient iteratively leads to the initial point toward the minimum solution.

where n denotes the iteration number and η is the step size that scales the gradient. As the gradient specifies the steepest direction and maximum increasing rate on a certain decision, the GD-based optimization algorithm usually has a fast and stable convergence speed. Fig. 2 graphically illustrates a common situation in which the black initial point repeatedly takes incremental steps toward the lowest point on the smooth orange curve.

The classic GD-based optimization algorithms have several well-known drawbacks that are 1) difficult to apply to non-differentiable functions; 2) η-parameter dependent; 3) easy to converge toward local minima if the objective function is non-convex; and 4) sensitive to the initial point. Although there are some drawbacks, GD-based optimization algorithms are still the most attractive optimizers for machine learning due to their computational efficiency, rapid convergence performance, and simple working principle.

III. Proposed Algorithm

As EMMOP is an emerging research topic in expensive optimization, few attempts have been made to develop SAEAs for seeking multiple optimal solutions for such an EOP. To have a better understanding of our contributions, we first introduce the motivation of this study, and then elaborate on each component of DSADE.

A. Motivation

In this study, an attempt has been made to fill the gap of SAEAs in seeking multiple optimal solutions of EMMOPs. Although such a kind of EOPs commonly arises in many real-world applications, fewer research efforts have been devoted to meeting the challenge of EMMOPs. Thus, there is a need for an efficient approach to handle EMMOPs.

With a limited number of FEs, how to explore the potential attraction basins and simultaneously exploit multiple promising solutions are two challenges for EMMOP optimizers. With respect to the first challenge, the generalization ability of the surrogate model is key to exploring as many potential attraction basins as possible. Among various regression models, MLP is a powerful neural network [45] that utilizes a back-propagation learning technique for training. Furthermore, its structure can be expanded into a larger one and then trained by deep learning techniques. This capacity allows the approximation to complex fitness landscapes, which is more potential to fit multiple peaks of EMMOPs. We thus develop an MLP-based surrogate model to predict the attraction basins associated with multiple optimal solutions. Considering the second challenge, a novel SaGDLS is proposed to exploit high-precision candidate solutions at a rapid speed. The identical minimum objective function value of multiple optimal solutions of EMMOPs enables the automation of the step size η for the decision vector by utilizing objective function information.

B. ARD

To solve EMMOPs, the population-based EA should converge toward different optima simultaneously. To this end, we divide the decision spaces into several promising areas and each of these areas is allocated an independent sub-population to search. Among various decision space division techniques, the decomposition method derived from clustering [43] is an efficient one. To better suit the property of EMMOPs, we propose an improved version to avoid parameter learning progress which requires a large number of exact FEs.

The decomposition technique is shown in **Algorithm 1**. The seed as centroid for each sub-population is identified by a small value f_s and a large distance γ_l from the individuals of which objective function values are also small. Given PS initial individuals $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{PS}\}$, they are sorted in ascending order according to their objective function values. Then, the minimal distance between $\mathbf{x}_i$ and the individuals ahead of $\mathbf{x}_i$ can be achieved as follows:

$$\gamma_i = \min_{j<i} dist(\mathbf{x}_i, \mathbf{x}_j) \tag{6}$$

where $dist(\mathbf{x}_i, \mathbf{x}_j)$ denotes the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. For the $\mathbf{x}_1$ with the minimum objective function value, its γ_1 can be specially defined

$$\gamma_1 = \max_{j=2,\ldots,PS} \gamma_j + 0.1 \tag{7}$$

where 0.1 is added to make sure that the best individual $\mathbf{x}_1$ has the largest distance among the PS initial individuals.

Afterward, γ and $f(\mathbf{x})$ are used together to identify the seeds of sub-populations. If individuals have small objective function values and large γ values, they are considered to be the seeds. In the original decomposition technique, two predefined thresholds f_{ref} and γ_{ref} are set. If individual $\mathbf{x}_i, i = 1, \ldots, PS$ satisfies $f(\mathbf{x}_i) \leq f_{ref}$ and $\gamma_i \geq \gamma_{ref}$, $\mathbf{x}_i$ is considered to be a seed. Once all seeds have been determined, each seed and its $PN - 1$ nearest neighbors are assigned to form the corresponding sub-population.

Fig. 4 shows a simple example of ARD, which is the F_{11} test function from the CEC'2020 multimodal competition [51]. The black points represent 200 initial individuals which are randomly generated in the two-dimensional decision space $\Re$. The red

Algorithm 1: ARD.

```
Input:
  • $PS$: the number of initial individuals;
  • $D$: the number of decision variables;
  • $PN$: the sub-population size.
Initialization:
  • Randomly generate $PS$ individuals (Set $P$) in $\Re$;
  • Evaluate the $PS$ individuals by exact FEs;
  • Sort the $PS$ individuals by their $f(\mathbf{x})$ values in ascending order;
  • Calculate the $\gamma$ value for each individual by (6);
  • Calculate the sequence number $r = \lfloor \frac{PS}{2D} \rfloor$;
  • $f_{ref} = f(\mathbf{x}_r)$ and $\gamma_{ref} = \gamma_r$.
Seed Identification
  Set $S = \{\}$;
  for i=1 to $PS$ do
    if $f(\mathbf{x}_i) \leq f_{ref}$ and $\gamma_i \geq \gamma_{ref}$
      Move $\mathbf{x}_i$ from $P$ into $S$.
Sub-population Allocation:
  Set $N = |S|$;
  Set $SP_i = \{\}, i = 1, \dots, N$;
  for i=1 to $N$ do
    For the $i$th seed in $S$, find its $PN - 1$ nearest neighbors from $P$;
    Copy the $i$th seed and its $PN - 1$ neighbors into $SP_i$.
Output:
  $N$ sub-populations $SP_i, i = 1, \dots, N$.
```

Algorithm 2: SaGDLS.

```
Input:
  • $\mathbf{x}$: the given individual to exploit;
  • $f_{\min}$: the estimated best objective value;
  • $D$: the number of decision variables;
  • $XS$ and $YS$: archives for exactly evaluated individuals;
  • $tfe$: the number of used exact FEs.
Initialization:
  • Calculate $\nabla f(\mathbf{x})$ by (4);
  • $tfe = tfe + D$;
  • Calculate $\nabla f(\mathbf{x})^{-1}$;
Self-adaptive GD:
  while any improvement has been found do
    Produce a new individual by (10);
    Evaluate $\mathbf{x}_{new}$ by exact FE;
    $tfe = tfe + 1$;
    Archive $\mathbf{x}_{new}$ and $f(\mathbf{x}_{new})$ into $XS$ and $YS$, respectively;
    if $f(\mathbf{x}_{new}) < f(\mathbf{x})$ then
      Set $\mathbf{x} = \mathbf{x}_{new}$;
    else
      Break the while loop.
Output:
  $\mathbf{x}$, $tfe$, $XS$ and $YS$.
```

points are eight seeds identified by ARD. The blue circles are the six optimal solutions that need to be located simultaneously. It can be observed that although black points are randomly distributed, the promising areas can be detected by these blue circles which are close to the red points.

The decomposition technique [44] can ideally divide individuals into a number of sub-populations without introducing new parameters. All the input parameters are also required by SAEAs. With respect to ARD, the number of sub-population is fully self-determined, which is more practical for different EMMOPs. In **Algorithm 1**, there are two major differences between our developed ARD and the original version.

- In [44], the sequence number of r is set to $\lfloor 0.3PN \rfloor$. In our method, the setting of r is related to the number of initial individuals PS and the dimensions D of the given problems.
- In [44], the size of each sub-population is usually different. In our method, each sub-population size is PN, which is in favor of the population evolution.

C. SaGDLS

GD is an efficient and fast technique for optimization methods to find an optimal solution. However, its drawbacks are also obvious. For instance, information from the gradient quite easily leads the search toward the local optima, and the convergence performance of classic GD-based optimization algorithms largely depends on the incremental step size. In this study, a novel SaGDLS is proposed to alleviate the above two issues by considering the coexistence of multiple optimal solutions. The pseudo-code of SaGDLS is present in **Algorithm 2**.

Given a decision vector $\mathbf{x}$ that is near to one of the optimal solutions, the objective function value and the negative of the gradient at $\mathbf{x}$ are $f(\mathbf{x})$ and $-\nabla f(\mathbf{x})$, respectively. Suppose the increment step size $\Delta\boldsymbol{\eta}$ of $\mathbf{x}$ to its nearest optimal solution is $\Delta\boldsymbol{\eta} = \{\eta_1, \dots, \eta_D\}$. Especially, the estimated objective function value of optimal solutions is $f_{\min}$. Therefore, the relationship among $\mathbf{x}$, $\Delta\boldsymbol{\eta}$, $f(\mathbf{x})$, $f_{\min}$, and $\nabla f(\mathbf{x})$ is as follows:

$$f_{\min} - f(\mathbf{x}) = \Delta\boldsymbol{\eta} \times \nabla f(\mathbf{x}) \tag{8}$$

and then it has

$$\Delta\boldsymbol{\eta} = (f_{\min} - f(\mathbf{x}))\nabla f(\mathbf{x})^{-1} \tag{9}$$

where $\nabla f(\mathbf{x})^{-1}$ is the pseudo-inverse of $\nabla f(\mathbf{x})$. With respect to (9), the value of $f(\mathbf{x})$ is known and $\nabla f(\mathbf{x})^{-1}$ can be obtained by numerical computations. Therefore, once the $f_{\min}$ has been estimated, the $\Delta\boldsymbol{\eta} = \{\eta_1, \dots, \eta_D\}$ can be calculated by the right-hand term of (9). Finally, a new vector can be achieved as follows:

$$\mathbf{x}_{new} = \mathbf{x} + \Delta\boldsymbol{\eta} \tag{10}$$

It is worth noting that, in (9), the right-hand term is used to estimate $\Delta\boldsymbol{\eta}$ on the left-hand side, therefore the $\nabla f(\mathbf{x})^{-1}$ in (9) is calculated by the forward finite difference independently.

To estimate $f_{\min}$, we utilize the coexistence of multiple optimal solutions with identical objective values. Suppose that N sub-populations have already been divided from ARD and

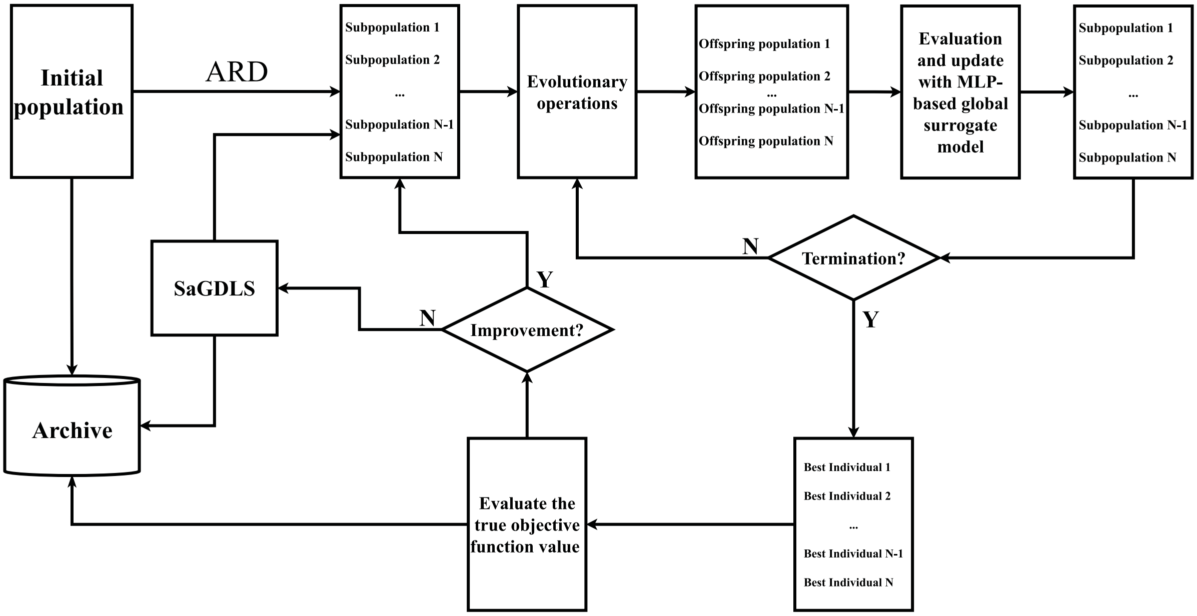

Fig. 3. Framework of DSADE.

the best objective function value f_b^t found comes from the ith sub-population at the current generation t. Then, there are two different situations to estimate $f_{\min}$ as follows:

- For the ith sub-population, $f_{\min}$ is estimated by $f_b^t + 0.0001$, where 0.0001 is the most often used accuracy requirement [52], [53], [54], [55] for multiple optimal solutions.
- For the other sub-populations, the best objective function value f_b^t found so far is directly used as $f_{\min}$.

In this scenario, if one sub-population has already achieved the best objective function value f_b^t, then it is expected that the other sub-populations should also achieve this value, otherwise, premature convergence may occur. Furthermore, in our proposed SaGDLS, $\Delta\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_D\}$ becomes variable step size where each variable of η is adaptively estimated.

Moveover, to make full use of the gradient information, an adaptive scheme has been designed. As shown in **Algorithm 2**, if a candidate solution has been successfully refined via the proposed SaGDLS, the same gradient information will be reused until no improvement can be obtained. In this way, the current gradient information has been fully dug to have the maximum achievement.

D. Complete Algorithm DSADE

In this section, the overall DSADE approach is described in detail.

At each generation, DSADE maintains the following information:

1) N sub-populations: $\{SP_i | i = 1, \ldots, N\}$;
2) PN individuals for $i$the sub-population: $\{\mathbf{x}_j^i | j = 1, \ldots, PN, i = 1, \ldots, N\}$;
3) The objective function value $f(\mathbf{x}_j^i), j = 1, \ldots, PN, i = 1, \ldots, N$.

The pseudo-code and framework of DSADE are outlined in **Algorithm 3** and Fig. 3, respectively . We first briefly describe the overall evolutionary progress of DSADE. Then, details are given to elaborate on the major components of DSADE which are marked with $*$ in **Algorithm 3**.

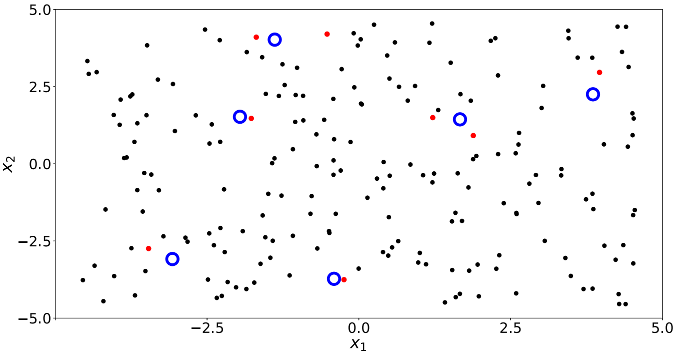

Fig. 4. Illustration of ARD. The eight seeds (red points) are identified from the 200 randomly initialized individuals (black points) via ARD. The blue circles represent the six global optima.

After the initialization phase by **Algorithm 1**, PS random individuals are automatically divided into N sub-populations, and each sub-population has PN individuals. Besides, if the decision vector has been evaluated by exact FE, such a vector and its objective function value will be stored in the training sets XS and YS, respectively. Afterward, an MLP regression model can be built based on the XS and YS. As a cheap surrogate $\hat{f}(\mathbf{x})$ for the exact FE, the MLP regression model is used to evaluate new intermediate individuals. In DSADE, the DE/current-to-rand/1/bin [56] is performed to produce offspring. Once the copied sub-population has evolved in sg generations, the most potential individual is evaluated by exact FE to update its corresponding predecessor (target vector) in the original sub-population. If the update is unsuccessful, the SaGDLS presented in **Algorithm 2** is activated to refine the best individual, otherwise, the SaGDLS is not performed. After several iterations, some different sub-populations may converge

Algorithm 3: DSADE.

```
Input:
  • $PS$: the number of initial individuals;
  • $D$: the number of variables;
  • $PN$: the sub-population size;
  • $sg$: the generation number for surrogated-based
  evolution.
Initialization:
  Execute ARD (Algorithm 1);
  Set $XS = \{\mathbf{x}_i | i = 1, \ldots, PS\}$;
  Set $YS = \{f(\mathbf{x}_i) | i = 1, \ldots, PS\}$.
Evolutionary Progress:
  while termination is not satisfied do
    * Use $XS$ and $YS$ to build a three-layer MLP
    regression model;
    for $n = 1$ to $N$ do
      * Copy the sub-population $SP_n$ as $SPC_n$;
      for $t_s = 1$ to $sg$ do
        For each $SPC_n$, produce its offspring population
        $OP_n$;
        Evaluate $OP_n$ with the MLP regression model;
        Update $SPC_n$ with $OP_n$;
      * Find the most potential individual $\mathbf{x}_p$ from $SPC_i$;
      Evaluate $\mathbf{x}_p$ by the exact FE;
      Archive $\mathbf{x}_p$ and $f(\mathbf{x}_p)$ into $XS$ and $YS$,
      respectively;
      if $\mathbf{x}_p$ is better than its predecessor $\mathbf{x}_i$ in $SP_n$ then
        Replace $\mathbf{x}_i$ with $\mathbf{x}_p$;
      else
        Find the best individual $\mathbf{x}_b$ from $SP_n$;
        Execute SaGDLS (Algorithm 2) to refine $\mathbf{x}_b$ as $\mathbf{x}'_b$;
        Use $\mathbf{x}'_b$ to update the worst individual in $SP_n$ if
        applicable;
    * Discard the redundant and inferior sub-populations,
    if any;
Output:
  The best individual $\mathbf{x}_b$ of sub-population
  $SP_n, n = 1, \ldots, N$.
```

into the same optimal solution, and some sub-populations may fall into the local optima. To avoid wasting exact FEs, such sub-populations will be detected and discarded.

In DSADE, four major components are specifically designed for EMMOPs.

- In DSADE, the fitness landscape of EMMOPs has multiple peaks (optimal solutions). We use a three-layer MLP neural network to enable the model capacity for learning and detecting as many peaks as possible. However, increasing the MLP model capacity helps reduce training error, but it also increases the risk of over-fitting, especially for the coexistence of multiple optimal solutions. To alleviate such an issue, we use all the exactly evaluated decision vectors to build the MLP regression model. Since most decision vectors are randomly distributed on the decision space, these data are balanced, varied, and unbiased to MLP model training.
- In DSADE, ARD is only executed once in the initialization phase. Afterward, each sub-population evolves independently. It is worth noting that, different from using the surrogate $\hat{f}(\mathbf{x})$ to prescreen offspring, we optimize the $\hat{f}(\mathbf{x})$ to obtain a desired candidate solution for exact FEs. To this end, we copy a given sub-population SP_n first, and then use the copied population SPC_n as the initial population to optimize the current MLP-based $\hat{f}(\mathbf{x})$. Fig. 3 shows the evolutionary progress to optimize $\hat{f}(\mathbf{x})$.
- After sg generations in **Algorithm 3**, the most potential individual x_p is chosen to have an exact FE. In this study, we first compare each $\mathbf{x}_j$ in SPC_n with its corresponding parent $\mathbf{x}'_j$ in SP_n, and the individual that has the largest potential improvement $\max_{j=1,\ldots,PN} (f(\mathbf{x})_j - \hat{f}(\mathbf{x})'_j)$ is determined as $\mathbf{x}_p$. In this way, we do not choose the individual with minimal $\hat{f}(\mathbf{x})$, which aims to balance the population diversity and convergence.
- During the evolutionary progress, some redundant sub-populations may converge toward the same optimal solution, and some inferior sub-populations may converge toward local optimal solutions. To avoid wasting exact FEs, we use the distance threshold δ and stagnation tolerance Q to detect the redundant and inferior sub-populations, respectively. Given two sub-populations, if the distance of their best individuals is less than δ, then we randomly discard one of these two sub-populations. If a certain sub-population has not been updated more than Q generations, such a sub-population is also discarded. Here, we borrowed the setting $\delta = 0.001$ from [57], and set the maximum value $\frac{MaxFEs-PS}{2 \cdot D \cdot N}$ to Q, where $MaxFEs$ is the maximum number of exact FEs.

E. Computational Time Complexity

With respect to the computational complexity of DSADE, apart from the exact FEs and ARD in the initialization phase, the major components in each generation are considered as follows:

1) The time complexity of using back-propagation to train a three-layer MLP neural network is $O(N_{itr} \cdot T_s \cdot (N_1 \cdot N_2 + N_2 \cdot N_3))$, where N_{itr} is the number of training iterations, T_s is the size of the training set, and N_1, N_2, and N_3 are the numbers of neurons of the three layers, respectively;
2) The time complexity of optimizing the MLP-based $\hat{f}(\mathbf{x})$ via DE is $O(N \cdot sg \cdot PN \cdot D)$;
3) The worst-case time complexity of SaGDLS is $O(D^3)$;
4) The worst-case time complexity of discarding the redundant and inferior sub-populations is $O(N \cdot D^2)$;

Therefore, the computational time complexity in each generation of DSADE is $O(N_{itr} \cdot T_s \cdot (N_1 \cdot N_2 + N_2 \cdot N_3) + N \cdot sg \cdot PN \cdot D + D^3 + N \cdot D^2)$.

IV. Experimental Study

A. Experimental Settings and Performance Metrics

As an emerging research problem in expensive optimization, so far, there has not been a well-designed test suite for EMMOPs to evaluate the performance of algorithms on seeking multiple optimal solutions. Therefore, we borrowed the 20 benchmark test functions from IEEE CEC'2020 multimodal competition [51]. These 20 test functions can be classified into three groups. The first group includes 10 widely used benchmark functions (F_1 to F_{10}) in the multimodal optimization community, functions (F_{11} to F_{15}) in the second group are low-dimensional composition test functions with many local optima, and the third group consists of five high-dimensional composition test functions (F_{16} to F_{20}). Different from the original settings of $MaxFEs$ for each cheap test function in [51], the $MaxFEs$ for SAEAs is set to a limited computational budget in this study. For example, the $MaxFEs$ is $4.0E + 03$ for F_{16} to F_{20}, which is only 1% of the original setting. Details can be found in Table S.I (supplementary file).

To evaluate the performance of DSADE, two evaluation criteria, namely peak ratio (PR) and success rate (SR) [51], are employed. PR represents the average ratio of all global optima found within an acceptable accuracy level ϵ over the number of trials. SR indicates the rate of successful trials over all trials. The way to calculate PR and SR is stated as follows:

$$\text{PR} = \frac{\sum_j^{N_T} N_{ago}^j}{NoG \cdot N_T} \qquad \text{SR} = \frac{N_{TS}}{N_T} \tag{11}$$

where N_T is the number of total trials and N_{ago}^j is the number of acceptable global optima in the jth trial, NoG is the number of known global optima of the test function, N_{TS} is the number of total successful trials. If all of the multiple global optima are found in a single trial, such a trial is a successful one.

B. Compared Algorithms and Parameter Settings

To measure the efficiency, DSADE is compared with three multimodal optimization approaches, namely, NCDE [53], MOMMOP [57] and EMO-MMO [58], and four expensive multimodal optimization approaches D/REM [2], GLEMOP [2], DSCPSO-EMM [38], and MaMPSO [40]. These seven competing approaches are briefly summarized in Table S.II (supplementary file).

With respect to DSADE, the number of initial individuals PS is set to $0.2 \cdot MaxFEs$. The sub-population size PN is set to $4 \cdot D$ so that four individuals at least can have to employ DE/current-to-rand/1/bin. The number of neutrons for each hidden layer is set to $200 \cdot D$. The designed MLP neural network is implemented by PyTorch v1.10.0. Particularly, the AdamW optimizer is adopted to minimize the loss function. The parameters of AdamW are set as default, which have achieved satisfactory performance in most experiments. With respect to the seven compared approaches, their parameter settings are as suggested in the respective references. With respect to PR and SR values, each algorithm is independently run in 25 trials at five different accuracy levels (i.e., ε = 1E-01, 1E-02, 1E-03, 1E-04, and 1E-05). To have fair comparisons, the $MaxFEs$ is the same for all approaches.

C. Comparison With State-of-the-Art Algorithms

TABLE I
WILCOXON'S RANK-SUM TEST RESULTS OF DSADE AND THE FIVE COMPARED ALGORITHMS

Algorithm	$R+$	$R-$
DSADE vs. NCDE	210.0	0.0
DSADE vs. MOMMOP	210.0	0.0
DSADE vs. EMO-MMO	210.0	0.0
DSADE vs. D/REM	137.0	73.0
DSADE vs. GLEMOP	202.5	7.5
DSADE vs. DSCPSO-EMM	208.5	1.5
DSADE vs. MaMPSO	208.5	1.5

In this section, the performance of DSADE is compared with seven algorithms. Table S.III (supplementary file) presents the PR and SR values obtained by DSADE at five accuracy levels. Since ε = 1.0E-04 is the most often used accuracy level in discussions [52], [53], [54], [55], PR values of the other seven approaches at such an accuracy level are compared with DSADE in Tables S.IV and V (supplementary file). The best PR value is highlighted in **boldface** for each test function. Moreover, Wilcoxon's rank-sum test [59] at a significant level $\alpha = 0.05$ is used here to test whether there is a statistical improvement between DSADE and the corresponding competitor. Symbols $+$ and $-$ indicate that DSADE is significantly better and significantly worse than the competitor, respectively.

1) Comparison With Multimodal Optimization Approaches: Table S.IV shows that DSADE achieves the best PR results on 14 out of 20 test functions. Particularly, the PR and SR values achieved by DSADE on F_1, F_2, F_3, F_4, and F_{10} are both 1.000, indicating DSADE can consistently find all the multiple global solutions for these five test functions in 25 consecutive trials. On the first 10 widely used test functions, DSADE achieves greater PR values than the other compared approaches. On the last ten composition test functions, the performance of the five competitive approaches and DSADE varies. While the performance of DSADE shows a slight decrease on the last 10 test functions. It is noteworthy that DSADE is capable of identifying at least one or two optimal solutions for F_{19} and F_{20}. Table S.IV also shows that the state-of-the-art algorithms NCDE, MOMMOP, and EMO-MMO struggle to locate multiple optimal solutions on most test functions simultaneously due to the limited computational budget. Their inferior performance indicates the difficulty of EMMOPs. In comparison, although DSADE cannot achieve high PR values on most test functions, it has more best PR values at the accuracy level ε = 1.0E-04 compared with the three multimodal optimization approaches, implying the feasibility of DSADE to solve EMMOPs.

Specifically, DSADE outperforms all the 20 test functions to NCDE, MOMMOP, and EMO-MMO. In Table I, Wilcoxon's test result confirms the performance of DSADE is superior to the three original multimodal optimization approaches as all $R+$ scores of DSADE are higher than the $R-$ scores of the corresponding competitor. It is clear that the application of surrogate assistance in DSADE can significantly reduce the requirement of exact FEs to find more global optimal solutions.

2) Comparison With Expensive Multimodal Optimization Approaches: To further verify its performance on EMMOPs, DSADE is compared with four other SAEAs. Given the truth

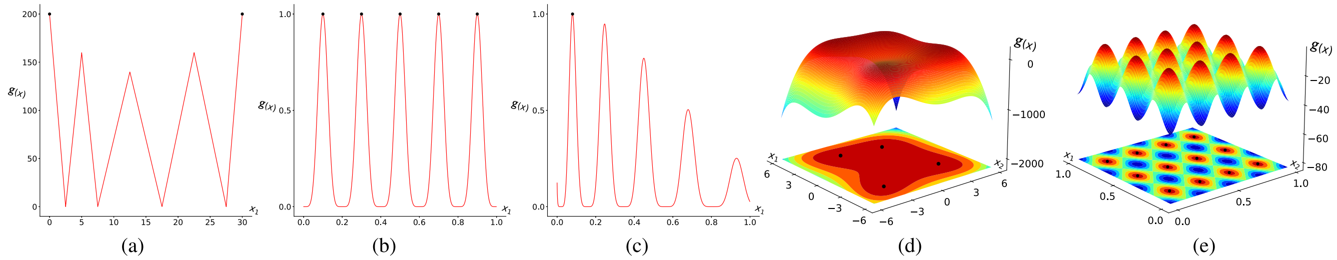

Fig. 5. Distribution of multiple optimal solutions on five EMMOPs. (a) F_1 (b) F_2 (c) F_3 (d) F_4 (e) F_{10}. The black points are acceptable optimal solutions.

that little attention has been paid to multiple optimal solutions of EMMOPs, we only find four SAEAs, i.e., D/REM, GLEMOP, DSCPSO-EMM, and MaMPSO, that focus on EMMOPs at present. Different from DSADE, D/REM, GLEMOP, and DSCPSO-EMM adopt the popular RBF network to approximate the multimodal fitness landscape, while a hybrid surrogate model including the RBF network, GP, and polynomial regression, is used in MaMPSO. It is worth noting that unlike DSADE, D/REM, and GLEMOP that only focus on the multiple global optima, DSCPSO-EMM and MaMPSO consider both the local and global optima, and thus their full performance is not available on these 20 test functions. When compared with these four SAEAs, the performance of DSADE is still competitive. Specifically, DSADE outperforms D/REM, GLEMOP, DSCPSO-EMM, and MaMPSO on 9, 17, 18, and 18 test functions, respectively. In contrast, D/REM, GLEMOP, DSCPSO-EMM, and MaMPSO perform better than DSADE on five, one, zero, and zero test function, respectively. Meanwhile, according to Table I, DSADE achieves more $R+$ scores than $R-$ scores over these four SAEAs on the 20 test functions.

Moreover, Fig. 5 depicts the fitness landscape and the obtained candidate solutions by DSADE on five test functions F_1, F_2, F_3, F_4, and F_{10}. Since the fitness landscape of these test functions can be plotted in a two- or three-dimensional diagram, an insight into the distribution of multiple optimal solutions can be visibly observed. Fig. 5 shows that even when the fitness landscape becomes more complex, such as the number of global optima becoming larger, DSADE can still locate all the optimal solutions on these test functions at high accuracy levels.

Overall, the above comparative analyses and statistical tests validate the feasibility and solid performance of our proposed DSADE to locate multiple optimal solutions on the 20 EMMOPs.

D. *Effectiveness of ARD*

In [44], the original decomposition technique was proposed to partition the decision space and detect promising areas for multimodal optimization problems. As an improved version, ARD can automate the two parameters f_{ref} and γ_{ref}, which are user parameter-free to partition the whole decision space and divide individuals into different promising areas. To validate our improvement, a computational trial, namely DSADE-OD, is designed. In this case, an original decomposition technique is used to replace ARD while the other components remain unchanged. The f_{ref} and γ_{ref} parameters are set to $\lfloor 0.3PN \rfloor$ as suggested in [44].

Table S.VI (supplementary file) lists the experimental PR results obtained by DSADE-OD. According to the results, DSADE-OD cannot match the performance of DSADE. On the test functions The PR values obtained by DSADE on most test functions are better than that of DSADE-OD. The reason is that the limited computational budget for EMMOPs cannot support the f_{ref} and γ_{ref} to detect the promising areas including both the global and local optima. For EMMOPs, the global optima are always preferred over the local optima. Meanwhile, the number of global optima is problem-dependent. A fixed value of f_{ref} and γ_{ref} is unsuitable for all test functions to detect the promising attraction basins in which the multiple global optima lie, thus decreasing the performance of the algorithm in solving EMMOPs.

Based on the above comparison, we can conclude that ARD is more suit the property of EMMOPs.

E. *Effectiveness of MLP-Based Surrogate Model*

Despite its distinctive advantages for EMMOPs, the MLP-based surrogate model has not received as much attention as the RBF network-based and GP-based surrogate models which are considered mainstream in SAEAs. Basically, MLP allows more hidden layers and each layer can have variable-length neurons to fit a complex nonlinear problem. Meanwhile, the training progress of MLP is highly adaptable with the help of the back-propagation technique. Once the structure of MLP has been determined, it does not require any other input parameters. To investigate the effectiveness of our MLP-based surrogate model, we designed three computational trials using different surrogate models. The RBF network, GP, and SVM have been used to replace the MLP-based surrogate models in DSADE, and accordingly, the corresponding algorithms are denoted as DSADE-RBF, DSADE-GP, and DSADE-SVM, respectively.

The empirical results obtained by DSADE, DSADE-RBF, DSADE-GP, and DSADE-SVM are presented in Table S.VII (supplementary file). We can see that: 1) the DSADE-RBF can find more than one optimal solution on most test functions. However, its performance is significantly inferior to that of DSADE; 2) DSADE-GP and DSADE-SVM can locate some optimal solutions among the first ten basic test functions, but their performance degrades seriously on the last ten composition functions. The main reason is that the surrogate of DSADE is designed as a global estimator to predict the multiple attraction basins and evolve the population for a certain number of generations. Therefore, the characteristic of generalization is highly desirable for a surrogate model in DSADE. To this end, all the evaluated data are used to train the surrogate, and therefore more information needs to be learnt. Compared with the other three surrogate models, MLP benefits from having the more hidden layers, variable-length neurons, and the back-propagation training technique, which enables it to learn information more effectively from a large number of input data.

The above discussions verify that the MLP-based surrogate model is more suitable for DSADE to seek multiple optimal solutions at a limited computational budget.

F. Effectiveness of Gradient-Based Local Search

Different from evolutionary strategy-related convergence enhancements, DSADE proposes a novel SaGDLS to achieve a rapid population convergence toward different optimal solutions. To investigate the effectiveness of SaGDLS, we design a variant of DSADE which discards the gradient descent, and therefore only DE/current-to-rand/1/bin is deployed to produce new individuals. This new variant is denoted as DSADE-WG, and its obtained experimental results are presented in Table S.VIII (supplementary file).

It can be observed that compared with DSADE, the performance of DSADE-WG drastically degrades on 16 out of 20 test functions. Moreover, DSADE-WG cannot find any optimal solution on $\mathrm{F}_1, \mathrm{F}_6, \mathrm{F}_8, \mathrm{F}_{16}, \mathrm{F}_{17}, \mathrm{F}_{18}, \mathrm{F}_{19}$, and F_{20} in 25 consecutive trials. Based on the above comparison, two points can be verified. First, SaGDLS largely enhances population convergence. Since DSADE wins all 20 test functions over DSADE-WG, the premature convergence of classic GD-based optimization algorithms would be alleviated by the adaptive strategy of SaGDLS. We attribute this to the ARD. Since the decision space has been divided into different small promising areas, it is easy to search each small area that contains a less number of optimal solutions. Second, the SaGDLS does not need a large amount of exact FEs as the classic GD-based optimization algorithms to obtain a highly accurate solution. Without SaGDLS, DSADE-WG uses all exact FEs to evaluate individuals. However, only one or two accurate optimal solutions can be located on 16 out of 20 functions. In contrast, although some exact FEs are used to calculate gradient, the convergence to multiple optimal solutions is not impaired. We attribute this to our proposed estimation of $f_{\min}$. Since each small area has been initially explored by each sub-population with DE/current-to-rand/1/bin, individuals have already approached to the near-optimal solutions. SaGDLS just refines rather than explores the best individual, and thus the estimation of $f_{\min}$ can be more efficient to exploit the given solution with a few exact FEs.

Considering the above comparisons, it can be concluded that the SaGDLS is quite efficient to assist DSADE to seek multiple optimal solutions for EMMOPs.

V. Application of DSADE to Real-World Problems

The application feasibility of DSADE is tested on a real-world problem called the building energy conservation design (ECD) [40], [60] which aims to minimize energy consumption by optimizing the structural parameter of a given building. To obtain energy consumption, the time-consuming software EnergyPlus is required to simulate the structure of the building. Meanwhile, different solutions may produce identical objective function values when attempting to minimize energy consumption. Herein, such a problem can be considered an EMMOP.

A. Experimental Settings

The problem setting is borrowed from [40], which takes the ECD of a single-room building in Beijing. Specifically, the input for software simulation consists of 12 decision variables which are most relevant to building energy performance. For each individual of DSADE, their genes represent a set of values for these 12 decision variables. Considering an exact FE, EnergyPlus should be run to simulate the building structure, and then calculate the energy consumption according to the input individual's genes. Afterward, the output result is the objective function value. Further details such as initial values, lower and upper bounds of variables can be found in [40].

We compare DSADE with the original multimodal optimization approach MOMMOP [57] and two SAEAs SA-MPSO [23] and DSP-SAEA [24]. A brief description of these two SAEAs can be found in the Introduction Section. Their major parameter settings are presented in Table S.II, and the other parameter settings are as suggested in the respective references. To have a clear observation, the $MaxFEs$ is set to a comparatively large value, i.e., 4.0E+03. Each algorithm is independently run 25 trials. Once all the global optima have been found, the corresponding trial will be stopped and the executing time will be recorded.

B. Result

Table S.IX (supplementary file) lists the PR, SR, and the average executing time obtained by DSADE, DSP-SAEA, SA-MPSO and MOMMOP, respectively. Compared with the other two SAEAs, DSADE needs more time to train the MLP-based surrogate and evolve the population. Nevertheless, DSADE can find all the global optimal solutions in 25 consecutive trials, which can satisfy the decision makers for their different demands. On the other hand, compared with the MOMMOP, although the DSADE achieves the same performance with MOMMOP on the PR and SR metrics, DSADE can solve the ECD problem using less exact FEs and executing time, which is more suitable for the real-world EMMOPs.

Based on the above comparisons, DSADE is highly competent to locate multiple global optimal solutions under a limited computational budget.

VI. Conclusion

This article proposes a new approach, DSADE, for seeking multiple optimal solutions for EMMOPs under a limited exact

FE budget. DSADE utilizes the MLP neural network to simulate the multimodal fitness landscape for the generalization ability of a surrogate model. As a multilayered deep learning neural network, MLP is more designable and suitable for complex fitness landscape fitting. Considering the discrete distribution of multiple optimal solutions, DSADE first detects the promising areas by the improved ARD, and then a new SaGDLS is designed for high-precision candidate solutions. Compared with the classic GD-based optimization algorithms, the SaGDLS is more efficient to refine individuals with a few exact FEs due to the adaptive estimation of the best objective function value. The combination of ARD and SaGDLS may offer a new way to seek multiple optimal solutions of EMMOPs for SAEAs.

We measure the performance of DSADE on 20 benchmark test functions. In the comparison with five state-of-the-art algorithms, DSADE shows its overwhelming efficiency to seek multiple optimal solutions simultaneously, and shows significant improvement to the compared algorithms on most test functions. The experimental study demonstrates the effectiveness of the proposed SaGDLS in exploiting high-precision candidate solutions.

Future work will focus on the improvements of the MLP-based surrogate model and evolutionary strategies for SAEAs to seek multiple optimal solutions for EMMOPs.

REFERENCES

[1] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2006.

[2] W. Gao, Z. Wei, M. Gong, and G. G. Yen, "Solving expensive multimodal optimization problem by a decomposition differential evolution algorithm," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2236–2246, Apr. 2023.

[3] Z.-H. Zhan et al., "Matrix-based evolutionary computation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 2, pp. 315–328, Apr. 2022.

[4] Z. Zhou, Y. S. Ong, M. H. Lim, and B. S. Lee, "Memetic algorithm using multi-surrogates for computationally expensive optimization problems," *Soft Comput.*, vol. 11, no. 10, pp. 957–971, 2007.

[5] Y. Wang, J. Lin, J. Liu, G. Sun, and T. Pang, "Surrogate-assisted differential evolution with region division for expensive optimization problems with discontinuous responses," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 780–792, Aug. 2022.

[6] S.-H. Wu, Z.-H. Zhan, and J. Zhang, "SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 478–491, Jun. 2021.

[7] Z. Li et al., "A novel deep learning framework based mask-guided attention mechanism for distant metastasis prediction of lung cancer," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 2, pp. 330–341, Apr. 2023.

[8] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[9] M. O. Akinsolu, B. Liu, V. Grout, P. I. Lazaridis, M. E. Mognaschi, and P. D. Barba, "A parallel surrogate model assisted evolutionary algorithm for electromagnetic design optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 93–105, Apr. 2019.

[10] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search methods," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 807–822, Dec. 2015.

[11] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.

[12] C. Hu, S. Zeng, and C. Li, "An uncertainty measure for prediction of non-Gaussian process surrogates," *Evol. Comput.*, vol. 31, no. 1, pp. 53–71, 2023.

[13] G. Bloch, F. Lauer, G. Colin, and Y. Chamaillard, "Support vector regression from simulation data and few experimental samples," *Inf. Sci.*, vol. 178, no. 20, pp. 3813–3827, 2008.

[14] J.-Y. Ji and M. L. Wong, "Surrogate-assisted parameter re-initialization for differential evolution," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2022, pp. 1592–1599.

[15] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 326–347, Jun. 2014.

[16] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, Aug. 2017.

[17] A. Dushatskiy, T. Alderliesten, and P. A. N. Bosman, "A novel approach to designing surrogate-assisted genetic algorithms by combining efficient learning of Walsh coefficients and dependencies," *ACM Trans. Evol. Learn. Optim.*, vol. 1, no. 2, pp. 1–23, Jul. 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3453141

[18] Y. Liu, Z. Yang, D. Xu, H. Qiu, and L. Gao, "A surrogate-assisted differential evolution for expensive constrained optimization problems involving mixed-integer variables," *Inf. Sci.*, vol. 622, pp. 282–302, 2023.

[19] F.-F. Wei et al., "A classifier-assisted level-based learning swarm optimizer for expensive optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 219–233, Apr. 2021.

[20] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005.

[21] Z. Yang, H. Qiu, L. Gao, X. Cai, C. Jiang, and L. Chen, "Surrogate-assisted classification-collaboration differential evolution for expensive constrained optimization problems," *Inf. Sci.*, vol. 508, pp. 50–63, 2020.

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

[23] Y. Liu, J. Liu, and Y. Jin, "Surrogate-assisted multipopulation particle swarm optimizer for high-dimensional expensive optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 7, pp. 4671–4684, Jul. 2022.

[24] Y. Liu, J. Liu, and S. Tan, "Decision space partition based surrogate-assisted evolutionary algorithm for expensive optimization," *Expert Syst. Appl.*, vol. 214, 2023, Art. no. 119075.

[25] J. Kůdela and R. Matoušek, "Combining Lipschitz and RBF surrogate models for high-dimensional computationally expensive problems," *Inf. Sci.*, vol. 619, pp. 457–477, 2023.

[26] X. Cai, L. Gao, and X. Li, "Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 365–379, Apr. 2020.

[27] K. Fu, X. Cai, B. Yuan, Y. Yang, and X. Yao, "An efficient surrogate assisted particle swarm optimization for antenna synthesis," *IEEE Trans. Antennas Propag.*, vol. 70, no. 7, pp. 4977–4984, Jul. 2022.

[28] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized multitasking for evolutionary optimization of expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 44–58, Feb. 2019.

[29] H. Zhen, W. Gong, and L. Wang, "Evolutionary sampling agent for expensive problems," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 716–727, Jun. 2023.

[30] X. Wu, Q. Lin, J. Li, K. C. Tan, and V. C. M. Leung, "An ensemble surrogate-based coevolutionary algorithm for solving large-scale expensive optimization problems," *IEEE Trans. Cybern.*, early access, Sep. 16, 2022, doi: 10.1109/TCYB.2022.3200517.

[31] H. Gu, H. Wang, and Y. Jin, "Surrogate-assisted differential evolution with adaptive multi-subspace search for large-scale expensive optimization," *IEEE Trans. Evol. Comput.*, early access, Dec. 06, 2022, doi: 10.1109/TEVC.2022.3226837.

[32] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, Apr. 2015.

[33] Y. Liu, J. Liu, S. Tan, Y. Yang, and F. Li, "A bagging-based surrogate-assisted evolutionary algorithm for expensive multi-objective optimization," *Neural Comput. Appl.*, vol. 34, no. 14, pp. 12097–12118, 2022.

[34] S. Qin, C. Sun, Q. Liu, and Y. Jin, "A performance indicator based infill criterion for expensive multi-/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 1085–1099, Aug. 2023.

[35] F.-F. Wei, W.-N. Chen, Q. Li, S.-W. Jeon, and J. Zhang, "Distributed and expensive evolutionary constrained optimization with on-demand evaluation," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 671–685, Jun. 2023.

[36] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, "A surrogate-assisted two-stage differential evolution for expensive constrained optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 715–730, Jun. 2023.

[37] Z. Yang, H. Qiu, L. Gao, D. Xu, and Y. Liu, "A general framework of surrogate-assisted evolutionary algorithms for solving computationally expensive constrained optimization problems," *Inf. Sci.*, vol. 619, pp. 491–508, 2023.

[38] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate-assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 794–808, Aug. 2021.

[39] C. Hu, S. Zeng, Y. Jiang, J. Sun, Y. Sun, and S. Gao, "A robust technique without additional computational cost in evolutionary antenna optimization," *IEEE Trans. Antennas Propag.*, vol. 67, no. 4, pp. 2252–2259, Apr. 2019.

[40] X. Ji, Y. Zhang, D. Gong, X. Sun, and Y. Guo, "Multisurrogate-assisted multitasking particle swarm optimization for expensive multimodal problems," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2516–2530, Apr. 2023.

[41] Q. Yang, W.-N. Chen, Y. Li, C. L. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 636–650, Mar. 2017.

[42] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[43] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[44] F. Gu, Y.-M. Cheung, and J. Luo, "An evolutionary algorithm based on decomposition for multimodal optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, 2015, pp. 1091–1097.

[45] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)–A review of applications in the atmospheric sciences," *Atmospheric Environ.*, vol. 32, no. 14/15, pp. 2627–2636, 1998.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[47] Z. Zhang, "Improved adam optimizer for deep neural networks," in *Proc. IEEE/ACM 26th Int. Symp. Qual. Serv.*, 2018, pp. 1–2.

[48] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Mini-batch gradient descent: Faster convergence under data sparsity," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 2880–2887.

[49] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 2121–2159, 2011. [Online]. Available: https://www.jmlr.org/papers/v12/

[50] S. L. Campbell and C. D. Meyer, *Generalized Inverses of Linear Transformations*. Philadelphia, PA, USA: SIAM, 2009.

[51] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization," Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne, VIC, Australia, Tech. Rep., 2013. Accessed: Apr. 2019. [Online]. Available: https://titan.csit.rmit.edu.au/~e46507/cec13-niching/competition/cec2013-niching-benchmark-tech-report.pdf

[52] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 246–263, Apr. 2015.

[53] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.

[54] W. Gao, G. G. Yen, and S. Liu, "A cluster-based differential evolution with self-adaptive strategy for multimodal optimization," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1314–1327, Aug. 2014.

[55] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.

[56] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution–An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, 2016.

[57] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.

[58] R. Cheng, M. Li, K. Li, and X. Yao, "Evolutionary multiobjective optimization-based multimodal optimization: Fitness landscape approximation and peak detection," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 692–706, Oct. 2018.

[59] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

[60] Z. Yong, Y. Li-Juan, Z. Qian, and S. Xiao-Yan, "Multi-objective optimization of building energy performance using a particle swarm optimizer with less control parameters," *J. Building Eng.*, vol. 32, 2020, Art. no. 101505.

Jing-Yu Ji received the M.Sc. degree in computer science from Sun Yat-Sen University, Guangzhou, China, in 2017 and the Ph.D. degree from Lingnan University, Hong Kong, in 2023. He is currently a Research Assistant Professor with the Department of Computing and Decision Sciences, Lingnan University. From 2017 to 2020, he was a Research Engineer with the Bonus Lab, National Institute for Research in Digital Science and Technology, France. His research interests include evolutionary computation, smart grid, and high-performance computing.

Zusheng Tan received the M.Sc. degree in information systems from the Department of Computer Science, Baptist University, Hong Kong, in 2019. He is currently working toward the Ph.D. degree with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong. His research includes information systems, machine learning, deep learning, and federated learning.

Sanyou Zeng received the M.Sc. degree in mathematics from Hunan University, Changsha, China, in 1995, and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2002. Since 2004, he has been a Professor with the China University of Geosciences, Wuhan, China. His research interests include evolutionary computation with machine learning for solving problems with constraints, multiobjective, dynamic environments, and expensive costs, especially antenna design problem.

Eric W. K. See-To received the B.Eng. degree from The Chinese University of Hong Kong, Hong Kong, in 1995, the M.B.A. degree from The University of Hong Kong, Hong Kong, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2005.

He is currently an Associate Professor and a Program Coordinator for M.Sc. in artificial intelligence and business analytics with Lingnan University, Hong Kong. His research interests include information systems and (big) data science applications, more specifically business data analytics, and AI education. His scholarly activities use data science techniques for the extraction of managerial insights in the areas of healthcare, finance, and electronic business, leveraging on the now available online big data streams such as social media. He had extensive consulting experiences for international firms in the banking and finance industry.

Man-Leung Wong (Member, IEEE) received the B.Sc., M.Phil., and Ph.D. degrees in computer science from The Chinese University of Hong Kong, Hong Kong, in 1988, 1990, and 1995, respectively. He is currently the Department Head and a Professor with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong. He is the Director of the LEO Dr. David P. Chan Institute of Data Science, Lingnan University. His research interests include artificial intelligence, data science, data mining, evolutionary computation, machine learning, and parallel algorithms on graphics processing units. His articles on these topics have been published in *Evolutionary Computation*, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Management Science*, *Journal of Management Information Systems*, *Information Sciences*, *Knowledge-Based Systems*, *Decision Support Systems*, *Expert Systems with Applications*, *Journal of the American Society for Information Science and Technology*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, IEEE INTELLIGENT SYSTEMS, IEEE ENGINEERING IN MEDICINE AND BIOLOGY MAGAZINE, *Fuzzy Sets and Systems*, and *International Journal of Approximate Reasoning*.

Supplementary Material for "A Surrogate-Assisted Evolutionary Algorithm for Seeking Multiple Solutions of Expensive Multimodal Optimization Problems"

TABLE S.I
The Basic Properties of 20 EMMOPs

Function	D	Number of global optima	Number of local optima	$MaxFEs$
F_1: Five-Uneven-Peak Trap	1	2	3	5.0E+02
F_2: Equal Maxima	1	5	0	5.0E+02
F_3: Uneven Decreasing Maxima	1	1	4	5.0E+02
F_4: Himmelblau	2	4	0	5.0E+02
F_5: Six-Hump Camel Back	2	2	2	5.0E+02
F_6: Shubert	2	18	many	2.0E+03
F_7: Vincent	2	36	0	2.0E+03
F_8: Shubert	3	81	many	2.0E+03
F_9: Vincent	3	216	0	2.0E+03
F_{10}: Modified Rastrigin	2	12	0	2.0E+03
F_{11}: Composition Function	2	6	many	2.0E+03
F_{12}: Composition Function	2	8	many	2.0E+03
F_{13}: Composition Function	2	6	many	2.0E+03
F_{14}: Composition Function	3	6	many	2.0E+03
F_{15}: Composition Function	3	8	many	2.0E+03
F_{16}: Composition Function	5	6	many	4.0E+03
F_{17}: Composition Function	5	8	many	4.0E+03
F_{18}: Composition Function	10	6	many	4.0E+03
F_{19}: Composition Function	10	8	many	4.0E+03
F_{20}: Composition Function	20	8	many	4.0E+03

TABLE S.II
DETAILS OF COMPARED STATE-OF-THE-ART ALGORITHMS

Alg.	Evolutionary Strategy	Major Parameter Settings
NCDE [1]	multiobjective optimization-based approach	$PS = 100$, $F = 0.5$, and $CR = 0.9$.
MOMMOP [2]	neighborhood crowding strategy-based DE approach	$PS = 100$, $F = 0.6$, and $CR = 0.8$.
EMO-MMO [3]	density indicator-based approach with fitness landscape approximation and peak detection	$PS = 500$, $m = 20$, and $\varphi = 0$.
D/REM [4]	RBF surrogate-assisted expensive optimization approach	If $MaxFEs \leq 300$, $PS = 50$; otherwise $PS = 100$.
GLEMOP [4]	global and local combined surrogate-assisted expensive optimization approach	If $MaxFEs \leq 300$, $PS = 50$; otherwise $PS = 100$.
DSCPSO-EMM [5]	dual-layer surrogate assisted PSO	$PS = 100$, RBFN-BP $= \{8, D, 8D\}$ and RBFN-BP $= \{D+1, \lceil\sqrt{D}+4\rceil, D+2\}$.
MaMPSO [6]	multi-surrogate-assisted multitasking PSO	$PS = 100$, $PN = 100$, $\varepsilon_{ini} = 0.5$, $c_{max} = 3$, and $DB_{ini} = 5D$.
DSP-SAEA [7]	decision space partition based surrogate-assisted evolutionary algorithm	$PS = 5D$, $PN = 50$, $n = 5$, and $\theta = 7$.
SA-MPSO [8]	surrogate assisted multi-population particle swarm optimizer	$PS = 100$, $\lambda = 0.5$, $fi = 200$, and $Si = 50$.

TABLE S.III
PR and SR Values Obtained by DSADE at Five Accuracy Levels

Accuracy	1.0E-01		1.0E-02		1.0E-03		1.0E-04		1.0E-05	
Level	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_4	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_6	0.256	0.000	0.256	0.000	0.256	0.000	0.256	0.000	0.256	0.000
F_7	0.433	0.000	0.419	0.000	0.400	0.000	0.394	0.000	0.378	0.000
F_8	0.019	0.000	0.016	0.000	0.016	0.000	0.015	0.000	0.014	0.000
F_9	0.077	0.000	0.067	0.000	0.059	0.000	0.053	0.000	0.050	0.000
F_{10}	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_{11}	0.700	0.000	0.667	0.000	0.667	0.000	0.667	0.000	0.667	0.000
F_{12}	0.388	0.000	0.388	0.000	0.388	0.000	0.388	0.000	0.388	0.000
F_{13}	0.667	0.000	0.667	0.000	0.667	0.000	0.667	0.000	0.667	0.000
F_{14}	0.767	0.300	0.600	0.000	0.600	0.000	0.600	0.000	0.600	0.000
F_{15}	0.375	0.000	0.375	0.000	0.375	0.000	0.375	0.000	0.375	0.000
F_{16}	0.517	0.400	0.250	0.000	0.250	0.000	0.250	0.000	0.250	0.000
F_{17}	0.350	0.000	0.200	0.000	0.200	0.000	0.200	0.000	0.200	0.000
F_{18}	0.267	0.100	0.117	0.000	0.117	0.000	0.117	0.000	0.117	0.000
F_{19}	0.037	0.000	0.037	0.000	0.037	0.000	0.037	0.000	0.037	0.000
F_{20}	0.113	0.000	0.075	0.000	0.075	0.000	0.062	0.000	0.062	0.000
Average	0.5483	0.3400	0.5042	0.3000	0.5029	0.3000	0.5016	0.3000	0.5006	0.3000

TABLE S.IV
PR Values Obtained by DSADE and NCDE, MOMMOP, and EMO-MMO at ε = 1E-04

Ins. / PR	DSADE	NCDE	MOMMOP	EMO-MMO
F_1	**1.000**	0.533	0.967	0.000
F_2	**1.000**	0.107	0.173	0.320
F_3	**1.000**	0.133	0.133	0.133
F_4	**1.000**	0.000	0.000	0.000
F_5	**1.000**	0.000	0.000	0.000
F_6	**0.256**	0.000	0.000	0.059
F_7	**0.394**	0.000	0.000	0.017
F_8	**0.015**	0.000	0.000	0.010
F_9	**0.053**	0.000	0.000	0.004
F_{10}	**1.000**	0.006	0.011	0.006
F_{11}	0.667	0.000	0.000	0.000
F_{12}	**0.388**	0.000	0.000	0.017
F_{13}	**0.667**	0.000	0.000	0.011
F_{14}	0.600	0.000	0.000	0.011
F_{15}	**0.375**	0.000	0.000	0.017
F_{16}	0.250	0.000	0.000	0.000
F_{17}	0.200	0.000	0.000	0.000
F_{18}	0.117	0.000	0.000	0.000
F_{19}	**0.037**	0.000	0.000	0.000
F_{20}	**0.062**	0.000	0.000	0.000

TABLE S.V

PR Values Obtained by DSADE and D/REM, GLEMOP, DSCPSO-EMM and MaMPSO at ε = 1E-04

Ins. / PR	DSADE	D/REM	GLEMOP	DSCPSO-EMM	MaMPSO
F_1	**1.000**	**1.000**	0.000	0.500	0.500
F_2	**1.000**	**1.000**	0.968	0.192	0.200
F_3	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
F_4	**1.000**	0.970	0.940	0.200	0.230
F_5	**1.000**	0.940	**1.000**	**1.000**	**1.000**
F_6	0.256	0.198	**0.273**	0.047	0.031
F_7	**0.394**	0.311	0.222	0.022	0.009
F_8	**0.015**	0.010	0.000	0.002	0.000
F_9	**0.053**	0.049	0.028	0.004	0.004
F_{10}	**1.000**	**1.000**	0.893	0.180	0.173
F_{11}	0.667	**0.733**	0.600	0.167	0.167
F_{12}	**0.388**	0.325	0.250	0.150	0.125
F_{13}	**0.667**	**0.667**	0.540	0.167	0.167
F_{14}	0.600	**0.667**	0.193	0.167	0.173
F_{15}	**0.375**	**0.375**	0.055	0.125	0.009
F_{16}	0.250	**0.667**	0.033	0.167	0.053
F_{17}	0.200	**0.325**	0.000	0.125	0.125
F_{18}	0.117	**0.167**	0.000	0.020	0.010
F_{19}	**0.037**	0.010	0.000	0.000	0.000
F_{20}	**0.062**	0.000	0.000	0.000	0.000

TABLE S.VI
EFFECTIVENESS INVESTIGATION ON ARD

Ins. / PR	DSADE	DSADE-OD	Ins. / PR	DSADE	DSADE-OD
F$_{1}$	**1.000**	**1.000**	F$_{11}$	**0.667**	0.600
F$_{2}$	**1.000**	**1.000**	F$_{12}$	**0.388**	0.000
F$_{3}$	**1.000**	**1.000**	F$_{13}$	**0.667**	0.000
F$_{4}$	**1.000**	0.960	F$_{14}$	**0.600**	0.153
F$_{5}$	**1.000**	0.940	F$_{15}$	**0.375**	0.155
F$_{6}$	**0.256**	0.117	F$_{16}$	**0.250**	0.000
F$_{7}$	**0.394**	0.314	F$_{17}$	**0.200**	0.000
F$_{8}$	**0.015**	0.008	F$_{18}$	**0.117**	0.000
F$_{9}$	**0.053**	0.002	F$_{19}$	**0.037**	0.000
F$_{10}$	**1.000**	**1.000**	F$_{20}$	**0.062**	0.000

TABLE S.VII
Effectiveness Investigation on MLP-based Surrogate Model

Ins. / PR	DSADE	DSADE-RBF	DSADE-GP	DSADE-SVM	Ins. / PR	DSADE	DSADE-RBF	DSADE-GP	DSADE-SVM
F_1	**1.000**	**1.000**	**1.000**	**1.000**	F_{11}	**0.667**	**0.667**	0.247	0.087
F_2	**1.000**	**1.000**	0.984	0.992	F_{12}	**0.388**	0.270	0.000	0.000
F_3	**1.000**	**1.000**	**1.000**	**1.000**	F_{13}	**0.667**	**0.667**	0.093	0.000
F_4	**1.000**	**1.000**	**1.000**	**1.000**	F_{14}	**0.600**	0.553	0.040	0.053
F_5	**1.000**	**1.000**	**1.000**	**1.000**	F_{15}	**0.375**	0.265	0.095	0.000
F_6	**0.256**	0.180	0.117	0.038	F_{16}	**0.250**	0.186	0.000	0.000
F_7	**0.394**	0.236	0.119	0.000	F_{17}	**0.200**	0.000	0.000	0.000
F_8	**0.015**	0.002	0.000	0.000	F_{18}	**0.117**	0.000	0.000	0.000
F_9	**0.053**	0.038	0.014	0.000	F_{19}	**0.037**	0.000	0.000	0.000
F_{10}	**1.000**	0.992	0.543	0.380	F_{20}	**0.062**	0.000	0.000	0.000

TABLE S.VIII
Effectiveness Investigation on SaGDLS

Ins. / PR	DSADE	DSADE-WG		DSADE	DSADE-WG	Ins. / PR	DSADE	DSADE-WG		DSADE	DSADE-WG
F_1	**1.000**	0.000	F_2	**1.000**	0.980	F_3	**1.000**	**1.000**	F_4	**1.000**	0.150
F_5	**1.000**	**1.000**	F_6	**0.256**	0.000	F_7	**0.394**	0.313	F_8	**0.015**	0.000
F_9	**0.053**	0.038	F_{10}	**1.000**	0.992	F_{11}	**0.667**	0.250	F_{12}	**0.388**	0.025
F_{13}	**0.667**	0.100	F_{14}	**0.600**	0.017	F_{15}	**0.375**	0.013	F_{16}	**0.250**	0.000
F_{17}	**0.200**	0.000	F_{18}	**0.117**	0.000	F_{19}	**0.037**	0.000	F_{20}	**0.062**	0.000

TABLE S.IX
RESULTS OF DSADE AND THE COMPARED ALGORITHMS ON THE ECD PROBLEM

Algorithm	PR	SR	Time (s)
DSADE	1.00	1.00	4633
SA-MPSO	0.50	0.00	2032
DSP-SAEA	0.54	0.08	2203
MOMMOP	1.00	1.00	52871

References

[1] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 601–614, 2012.

[2] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 830–843, 2015.

[3] R. Cheng, M. Li, K. Li, and X. Yao, "Evolutionary multiobjective optimization-based multimodal optimization: Fitness landscape approximation and peak detection," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 692–706, 2018.

[4] W. Gao, Z. Wei, M. Gong, and G. G. Yen, "Solving expensive multimodal optimization problem by a decomposition differential evolution algorithm," *IEEE Transactions on Cybernetics*, pp. 1–11, 2021.

[5] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate-assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 794–808, 2021.

[6] X. Ji, Y. Zhang, D. Gong, X. Sun, and Y. Guo, "Multisurrogate-assisted multitasking particle swarm optimization for expensive multimodal problems," *IEEE Transactions on Cybernetics*, pp. 1–15, 2021.

[7] Y. Liu, J. Liu, and S. Tan, "Decision space partition based surrogate-assisted evolutionary algorithm for expensive optimization," *Expert Systems with Applications*, vol. 214, p. 119075, 2023.

[8] Y. Liu, J. Liu, and Y. Jin, "Surrogate-assisted multipopulation particle swarm optimizer for high-dimensional expensive optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4671–4684, 2022.