



**VICTORIA UNIVERSITY**  
MELBOURNE AUSTRALIA

*Data-Driven Evolutionary Algorithm With  
Perturbation-Based Ensemble Surrogates*

This is the Published version of the following publication

Li, J, Zhang, Z, Wang, Hua and Zhang, Jun (2020) Data-Driven Evolutionary Algorithm With Perturbation-Based Ensemble Surrogates. IEEE Transactions on Cybernetics. ISSN 2168-2267

The publisher's official version can be found at  
<https://ieeexplore.ieee.org/document/9163270>  
Note that access to this version may require subscription.

Downloaded from VU Research Repository <https://vuir.vu.edu.au/42263/>

# Data-Driven Evolutionary Algorithm With Perturbation-Based Ensemble Surrogates

Jian-Yu Li, *Student Member, IEEE*, Zhi-Hui Zhan<sup>1</sup>, *Senior Member, IEEE*, Hua Wang<sup>2</sup>, *Member, IEEE*,  
and Jun Zhang<sup>3</sup>, *Fellow, IEEE*

**Abstract**—Data-driven evolutionary algorithms (DDEAs) aim to utilize data and surrogates to drive optimization, which is useful and efficient when the objective function of the optimization problem is expensive or difficult to access. However, the performance of DDEAs relies on their surrogate quality and often deteriorates if the amount of available data decreases. To solve these problems, this article proposes a new DDEA framework with perturbation-based ensemble surrogates (DDEA-PES), which contain two efficient mechanisms. The first is a diverse surrogate generation method that can generate diverse surrogates through performing data perturbations on the available data. The second is a selective ensemble method that selects some of the prebuilt surrogates to form a final ensemble surrogate model. By combining these two mechanisms, the proposed DDEA-PES framework has three advantages, including larger data quantity, better data utilization, and higher surrogate accuracy. To validate the effectiveness of the proposed framework, this article provides both theoretical and experimental analyses. For the experimental comparisons, a specific DDEA-PES algorithm is developed as an instance by adopting a genetic algorithm as the optimizer and radial basis function neural networks as the base models. The experimental results on widely used benchmarks and an aerodynamic airfoil design real-world optimization problem show that the proposed DDEA-PES algorithm outperforms some state-of-the-art DDEAs. Moreover, when compared with traditional nondata-driven methods, the proposed DDEA-PES algorithm only requires about 2% computational budgets to produce competitive results.

**Index Terms**—Data-driven evolutionary algorithm (DDEA), ensemble surrogates, genetic algorithm (GA).

Manuscript received November 26, 2019; revised May 19, 2020; accepted June 25, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102; in part by the Outstanding Youth Science Foundation under Grant 61822602; in part by the National Natural Science Foundations of China under Grant 61772207 and Grant 61873097; in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002; in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003; and in part by the Guangdong-Hong Kong Joint Innovation Platform under Grant 2018B050502006. This article was recommended by Associate Editor H. Takagi. (*Corresponding authors: Zhi-Hui Zhan; Jun Zhang.*)

Jian-Yu Li and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, South China University of Technology, Guangzhou 510006, China (e-mail: zhanapollo@163.com).

Hua Wang is with the College of Engineering and Science, Institute for Sustainable Industries and Liveable Cities, Victoria University, Melbourne, VIC 8001, Australia.

Jun Zhang is with Victoria University, Melbourne, VIC 8001, Australia (e-mail: csjun@scut.edu.cn).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCYB.2020.3008280

## I. INTRODUCTION

IN RECENT years, data-driven evolutionary algorithms (DDEAs) have received increasing attention in solving many real-world optimization problems, such as trauma system optimization [1], air ventilation system design [2], blast furnace optimization [3], and many others [4]. This is mainly due to two reasons. First, evolutionary algorithms (EAs) are efficient tools for tackling optimization problems with different properties and challenges, such as large scale [5]–[7], dynamic [8], multimodal [9]–[11], multiobjective [12]–[14], and many objective [15]. Second, there is an increasing number of real-world optimizations requiring distributed approaches [16], [17] and data-driven approaches [18], because their objective functions (and/or constraints functions) are always expensive, computationally intensive, or time consuming to perform. That is, evaluating the fitness (i.e., quality) of candidate solutions can be unaffordable in such real-world application problems [19]. For example, one evaluation of a high-fidelity crashworthiness analysis in the automotive industry can take several days and, therefore, finishing  $10^4$  times of evaluations for a crashworthiness design can take more than 100 years [20], which is unrealistic for engineering productions. Instead of using expensive fitness evaluations (FEs) and/or constraints evaluations, data-driven methods can provide cheaper and more efficient ways to carry out the evolutionary optimizations. Specifically, based on some evaluated data (e.g., candidate solutions evaluated by real FEs), data-driven methods can build surrogate models to approximate or replace the real FEs to drive the evolutions, which can reduce the needs on expensive FEs in the optimization procedures. Therefore, by combining EAs and data-driven methods, DDEAs can be more potential and efficient than traditional methods including traditional EAs because DDEAs can drive the optimizations through data and surrogates instead of performing expensive FEs [21], [22].

Generally speaking, existing research for improving DDEAs can be roughly classified into two categories. The first category mainly aims to improve data quality and data quantity, because data with higher quality and of larger amounts are useful for building more accurate surrogates [4]. Therefore, many data processing and data generation methods have been proposed, such as local smooth methods [3], data mining techniques [18], and artificial data generation [23]. The second

category is to improve surrogate quality, for example, the accuracy and robustness of surrogates. To obtain better surrogates, the users can select appropriate methods to build suitable surrogates, such as polynomial fitting [24], Kriging model [25], neural networks [26], and many others [27], [28]. In addition, when given a set of prebuilt homogeneous or heterogeneous surrogates, better models can be generated through managing and combining prebuilt surrogates properly [29], [30]. However, some studies also show that ensemble surrogates that are more efficient than a single surrogate in theoretical (mathematical functions) problems may not always work better in real-world application problems because the nature of each optimization problem may favor different surrogates [31]–[34]. Therefore, further and more intelligent surrogate ensemble methods are needed to be researched and studied. Moreover, as DDEAs rely heavily on surrogate predictions to evolve candidate solutions, their optimization accuracy may greatly deteriorate if they cannot make full use of the limited data to generate accurate surrogates.

To solve the above problems, this article proposes a new DDEA framework with perturbation-based ensemble surrogates, called DDEA-PES. The proposed framework contains two efficient mechanisms. First, it employs a diverse surrogate generation (DSG) method to generate a set of diverse surrogates. This is achieved by first performing data perturbations on the given dataset to obtain diverse datasets and second training surrogates on each new dataset independently. Second, it adopts a selective ensemble (SE) method to select some of the prebuilt surrogates to form a final ensemble surrogate model. By combining the DSG and the SE, the proposed framework can have the following three advantages. First, if the given data are insufficient for building accurate surrogates, data perturbations can increase the data quantity for building better surrogates. Second, the algorithm can make full use of the given data by using DSG to generate a large number of diverse datasets from a given dataset and to obtain a set of diverse surrogates. Third, the SE can obtain a final surrogate model with higher effectiveness and efficiency by selecting and combining the better surrogates from prebuilt surrogates. To validate the proposed framework, this article provides theoretical analyses to study the effectiveness of data perturbations. Furthermore, experimental comparisons are performed on benchmark functions and a real-world aerodynamic airfoil design optimization problem to investigate the DDEA-PES. To conduct the experimental comparisons, a specific algorithm is developed based on the proposed framework, which employs genetic algorithm (GA) as the optimizer and radial basis function neural networks (RBFNNs) as the base models, respectively. In addition, state-of-the-art algorithms and traditional methods are adopted as contenders in the experimental comparisons.

The remainder of this article is organized as follows. Section II provides background knowledge and related work, while Section III introduces the DDEA-PES and gives theoretical analyses. Section VI is for the experiments and comparisons. Finally, Section V gives the conclusion and future work.

## II. BACKGROUND AND RELATED WORK

### A. Data-Driven Evolutionary Algorithm

As the complexity and scale of optimization problems increase rapidly, FEs are increasingly expensive and difficult to access. In such cases, the performance of EAs often deteriorates due to the lack of enough FEs for evolving individuals, which poses great challenges to EAs in real-world applications [35]–[37]. Therefore, many EAs have been incorporated with data-driven methods such as surrogate models to improve their performance [38].

Usually, DDEAs mainly aim to drive the evolutionary search based on candidate solutions that have been evaluated by real FEs [39]. Such processes can be achieved by using surrogates to approximate and replace the real FEs as much as possible [18], [21]. In other words, by using evaluated data (i.e., the evaluated solutions) to build suitable surrogates, DDEAs can utilize these surrogates for driving the evolution search. As the data-driven mechanism can reduce the needs of conducting expensive FEs, DDEAs can easily outperform many traditional EAs when solving computationally intensive, time consuming, and expensive optimizations [22].

According to whether new candidate solutions can be evaluated by exact FEs, DDEAs can be divided into two categories [4]: 1) online DDEAs and 2) offline DDEAs. In online DDEAs, some candidate solutions will be still evaluated by real FEs during the optimization procedure. These newly evaluated solutions can be utilized as the new training data to refine and improve the existing surrogate models, which can further increase the surrogate accuracy and search efficiency [39]. Therefore, online DDEAs can be suitable for the problems that few FEs are still available from physical experiments or expensive calculations [40]. Differently, offline DDEAs build surrogate models only based on the historical evaluated data and no new solutions are evaluated by real FEs during the optimization procedure [19]. Although online and offline DDEAs are different in their data collection, they have similar ways of processing collected data and building surrogates. Therefore, methods proposed for processing data and managing surrogates in offline algorithms can be used in online algorithms as well. Based on this, without loss of generality, this article mainly details the proposed framework for offline DDEAs.

### B. Related Work

This part reviews some relevant research and discusses the differences between them and our DDEA-PES. As briefly introduced in Section I, research for enhancing DDEAs can be generally classified into two categories as on data and as on surrogate [41].

The first category aims to improve the quality and quantity of evaluated data [23], [41]. As the evaluated data are crucial for obtaining accurate surrogates, both the quality and quantity of evaluated data can have very significant influences on the optimization accuracy of DDEAs [4]. Consequently, many research have been conducted to solve these problems. For poor-quality data, including data with imbalanced distribution [42], incomplete information [43], and noisy [44],

preprocessing methods can be helpful [45]. For example, on a blast furnace optimization application, Chugh *et al.* [3] smoothed the noisy data through a local regression and then built Kriging models as surrogates based on processed data. As for big data applications [46], data mining and some related methods have been adopted to reduce data redundancy for building surrogates and the long calculation time for accessing big data. For instance, Wang *et al.* [18] used data mining techniques to recognize useful patterns in bid data, which could save about 90% of running time when optimizing a trauma system. In general, the above methods aim to improve data quality rather than the data quantity. Therefore, they may not work well if the amount of available data is insufficient in obtaining high-quality surrogates. Differently, the proposed DDEA-PES can make better use of the limited data through data perturbations, which can alleviate the data shortage problems.

As data shortage is often the largest challenge for approximating fitness functions, some research attempt to solve this problem by generating additional data [4], [12]. For instance, Guo *et al.* [23] generated artificial data through a low-order polynomial model. Although this method has obtained promising results, the drawback of this method is that the reliability of the generated data may depend heavily on the low-order model. Differently, Wang *et al.* [19] proposed an SE method, which generated diverse datasets through a bootstrap method. In this way, a number of different surrogates can be trained on these datasets, respectively, and then combined for predictions. The difference between this method and the DDEA-PES is that the bootstrap method obtains datasets by randomly resampling the evaluated data while the DDEA-PES generates datasets by data perturbations. Besides, transfer learning techniques are also likely to be effective for alleviating the data shortage. For instance, Ding *et al.* [47] transferred knowledge from computationally cheap problems to expensive problems, which can improve surrogate accuracy. However, such knowledge transfers require the shared characteristics or features between the source and target problems. In other words, transfer learning methods are problem dependent. In contrast, the proposed DDEA-PES does not require such assumptions and therefore can be suitable for a wider range of problems.

The second category aims to obtain better surrogates based on given data. So far, many methods have been studied for choosing more suitable methods and models to build surrogates. These models can be polynomial regression [24], Kriging models [25], traditional interpolation methods [48], and many others [49]. Also, machine-learning techniques are popular for building surrogates, which include artificial neural networks [26] and RBFNNs [27], [28]. Moreover, Sun *et al.* [48] proposed a novel fitness approximation strategy based on PSO, which could estimate fitness according to the positional relationship between particles in a PSO.

However, the above research have shown that each model has its own advantages and no surrogate model can be the best for all problems [4]. Therefore, many methods of model management are proposed to combine the advantages of different surrogates. For example, Wang *et al.* [29] made a better

balance between global and local searches by combining global and local surrogates. Similarly, Sun *et al.* [30] designed a two-layer surrogate-assisted PSO (TLSAPSO) that adopted local surrogates to locate global optimum and employed global surrogates to smooth out local optimum. Also, the surrogate-assisted cooperative swarm optimization algorithm (SA-COSO) was proposed with an estimation method and RBFNNs for solving high-dimensional problems [28]. Furthermore, based on a set of surrogates, committee-based active learning for the surrogate-assisted PSO algorithm (CAL-SAPSO) employed the committee-based decision for predictions [39]. As online DDEAs are able to evaluate new data through real FEs and employ them to update surrogates, many methods have been studied for selecting the most suitable individuals for FEs. Based on the mechanism for selecting individuals, related methods can be classified into generation- and individual-based strategies [4]. The generation-based strategies are to evaluate all solutions based on generations, according to the adaptive or predefined settings of the frequency for conducting evaluations [26]. Differently, only some individuals will be evaluated in individual-based strategies. In these strategies, the selections of individuals are often based on two factors: 1) the promising and 2) uncertain individuals [4]. The promising individuals, namely, the individuals with better-predicted fitness, may provide more useful information to capture the exact optimum position [26], while uncertain individuals, individuals with uncertainty predictions, can provide information to increase surrogate accuracy on uncertain areas [50]. However, how to measure uncertainty is not an easy problem. Hence, some methods are frequently used in many model management strategies to provide uncertainty measurements of predictions, such as the Gaussian process-assisted EA designed for the medium-scale expensive problem (GPEME) [40]. Also, some research measure the uncertainty based on the variance of surrogate outputs [39], [50]. In addition, a branch of strategies called infill criteria is researched to consider both the prediction fitness and uncertainty together to combine their advantages, including expected lower confidence bound (LCB) [40], probability of improvement (PoI) [51], and expected improvement (ExI) [52]. Furthermore, based on this, multiobjective infill criteria have also shown effectiveness when minimizing fitness and uncertainty together [21]. In summary, the above methods mainly consider how to manage and update surrogate models based on given data by employing techniques in data analysis and knowledge discovery. Different from these existing methods, the DDEA-PES provides a novel and efficient way for model building and management, which can adaptively select the proper surrogate ensemble according to the problem at hand.

### III. PROPOSED DDEA-PES ALGORITHM

#### A. Overall Framework

The overall framework of DDEA-PES is shown in Fig. 1. As mentioned before that the model management methods for offline algorithms can also be used in online algorithms [19], for simplicity, Fig. 1 is the version of offline DDEA-PES,

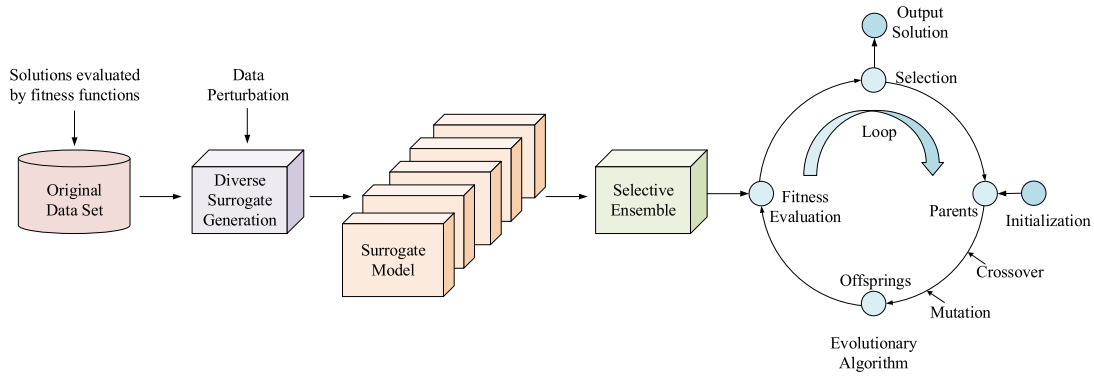


Fig. 1. Diagram of the overall DDEA-PES framework.

which denotes all data evaluated by real fitness functions as original data.

In general, the DDEA-PES can be divided into two parts: 1) the employed EA and 2) the employed surrogate model. The employed EA in DDEA-PES is the same as traditional EAs, which includes initialization, variation, FE, and selection. Hence, DDEA-PES can use different EAs as the optimizers, such as PSO and GA.

The surrogate model employed in DDEA-PES is the perturbation-based ensemble surrogates (PES) proposed in this article. Given the original data, PES first performs the DSG based on data perturbations to generate a set of surrogate models. Then, it employs the SE to select some of the existing surrogates to form the final ensemble model. This ensemble model will be used to replace the real FEs in the selection part of the employed EA. Driven by the ensemble model, the employed EA iteratively evolves its individuals and finally outputs the solutions when the stop criteria are met. The data perturbation, DSG, and SE will be detailed in the following contents one by one.

### B. Data Perturbation

The data perturbation aims to generate diverse datasets by perturbing the original data in order to build different surrogates for ensemble selections. To better describe the data perturbations, we use a similar method as in [41] and the notations used in the following contents are introduced herein. First, this article denotes the data evaluated by real FEs and the data generated by data perturbations as “original data” and “generated data,” respectively. Second, we denote the training dataset ( $TS$ ) containing all the original data with the corresponding fitness as  $TS = \{x_i, F(x_i)\}_{i=1}^N$ , where  $N$  is the total number of the original data and  $F(x)$  is the fitness value of  $x$ . Third, a subset of  $TS$  is denoted as  $S$  that contains the selected data for data perturbations. Based on these notations, the generated dataset  $K$  generated by data perturbations can be represented as follows:

$$K = \{(x_{\text{gen}}, F(x_{\text{gen}})) | x_{\text{gen}} = x_s + \Delta x, |\Delta x| \leq l, x_s \in S\} \quad (1)$$

$$l = \sqrt{\frac{\sum_{i=1}^D (UB_i - LB_i)^2}{D}} \times 10^{-6} \quad (2)$$

where  $\Delta x$  is a random vector,  $l$  is the maximum length of  $\Delta x$ ,  $D$  is the problem dimension, and  $UB_i$  and  $LB_i$  are the upper bound and lower bound of the  $i$ th dimension, respectively. Then, the diverse training set ( $DTS$ ) can be represented as the union of  $K$  and  $TS$

$$DTS = TS \cup K. \quad (3)$$

Notice that if we have sufficient small  $l$  in (1), the fitness of  $x_{\text{gen}}$  and  $x_s$  can be nearly the same for continuous fitness functions. Therefore, we set the fitness value of  $x_{\text{gen}}$  the same as  $x_s$ , that is,  $F(x_{\text{gen}}) = F(x_s)$ , where  $F(x_{\text{gen}})$  and  $F(x_s)$  will be referred as the exact fitness and the approximated fitness of  $x_{\text{gen}}$ , respectively, in the rest of this article. In this way, we can obtain the additional data (i.e.,  $x_{\text{gen}}$ ) without consuming FEs.

Now, we analyze the effect of such data perturbations. Considering a surrogate model  $M$  and denoting its prediction on data  $x$  as  $M(x)$ , its prediction error can be defined as

$$Err(M, F, x) = L(F(x), M(x)) \quad (4)$$

where  $L$  is a loss function. For simplicity, we consider the absolute loss function in this part, that is,  $L(a, b) = |a - b|$  with  $a$  and  $b$  as real numbers. Then, given  $x_{\text{gen}}$ , which is generated by perturbing  $x_s$ , and its approximate fitness  $F(x_s)$ , we have

$$\begin{aligned} Err_{\text{appr}}(M, F, x_{\text{gen}}) &= L(F(x_s), M(x_{\text{gen}})) \\ &= |F(x_s) - M(x_{\text{gen}})| \\ &= |F(x_s) - F(x_{\text{gen}}) + F(x_{\text{gen}}) - M(x_{\text{gen}})| \\ &\leq |F(x_s) - F(x_{\text{gen}})| \\ &\quad + |F(x_{\text{gen}}) - M(x_{\text{gen}})| \\ &= |F(x_s) - F(x_{\text{gen}})| \\ &\quad + Err_{\text{exact}}(M, F, x_{\text{gen}}) \end{aligned} \quad (5)$$

where  $Err_{\text{appr}}$  and  $Err_{\text{exact}}$  represent the prediction error when the fitness of data is approximated and is exact, respectively. For clarity, the following contents simply denote  $x_{\text{gen}}$  as  $x$  when it belongs to  $K$ . Then, given a dataset  $K$  which is generated through (1) with the data  $x_s$  in  $S$ , the expected error produced by  $M$  on  $K$  can be defined as

$$E_{\text{appr}}(M, F, K) = \sum_{x \in K} p(x) \cdot Err_{\text{gen}}(M, F, x) \quad (6)$$

where  $E_{\text{appr}}(M, F, K)$  represents the expected prediction error of  $M$  on  $K$  when the fitness of all data  $x$  in  $K$  is approximated,

and  $p(x)$  is the distribution of data  $x$ . Combining (5) and (6), we can have

$$\begin{aligned}
E_{\text{appr}}(M, F, K) &= \sum_{x \in K} p(x) \cdot \text{Err}_{\text{gen}}(M, F, x) \\
&\leq \sum_{x \in K} p(x) \\
&\quad \times (\text{Err}_{\text{exact}}(M, F, x) + |F(x_s) - F(x)|) \\
&= \sum_{x \in K} p(x) \cdot \text{Err}_{\text{exact}}(M, F, x) \\
&\quad + \sum_{x \in K} p(x) \cdot |F(x_s) - F(x)| \\
&= E_{\text{exact}}(M, F, K) + \sum_{x \in K} p(x) \cdot |F(x_s) - F(x)|
\end{aligned} \tag{7}$$

where  $E_{\text{exact}}(M, F, K)$  represents the expected prediction error of  $M$  on  $K$  when the fitness of all data  $x$  in  $K$  is exact. As the data in  $TS$  rarely will be the same with others, we can simply assume that the value of  $p(x)$  for all data is the same on a set  $K$ . That is, data are uniformly distributed on  $K$ . Then, (6) can be rewritten as

$$E_{\text{appr}}(M, F, K) \leq E_{\text{exact}}(M, F, K) + \frac{1}{|K|} \sum_{x \in K} |F(x_s) - F(x)| \tag{8}$$

where  $|K|$  is the size of dataset  $K$ .

Furthermore, according to (1), as the data in  $S$  are finite, we can always find a constant  $Q$  that can satisfy (9) for any data  $x_s$  in  $S$  and for the corresponding  $x = x_s + \Delta x$  in  $K$

$$|F(x_s) - F(x)| \leq Q \cdot |x - x_s| = Q \cdot |\Delta x| \leq Q \cdot l. \tag{9}$$

Combining (8) and (9), we can have

$$\begin{aligned}
E_{\text{appr}}(M, F, K) &\leq E_{\text{exact}}(M, F, K) + \frac{\sum_{x \in K} Ql}{|K|} \\
&= E_{\text{exact}}(M, F, K) + Ql.
\end{aligned} \tag{10}$$

The inequation given in (10) can provide two implications. First, when approximating continuous objective functions [which can have a small  $Q$  to satisfy (9)], data perturbations can help obtain high accuracy surrogates within less FEs because the training error of surrogates based on data with approximated fitness can be similar to those based on exact fitness. For an extreme example, when the approximated functions are constant functions ( $Q$  can be zero), the surrogates trained on data with approximated fitness will be the same as those on exact fitness. Second, by properly controlling  $l$ , the training error obtained by surrogates based on data with approximated fitness can be similar to those based on exact fitness. This provides a cost-effective way of building surrogates. Therefore, the surrogate trained on datasets after perturbations can be similar to those on datasets with real fitness, where the former requires less FEs than the latter.

### C. Diverse Surrogate Generation

Based on the data perturbations, diverse surrogates can be generated independently, as shown in Fig. 2. Algorithm 1 is

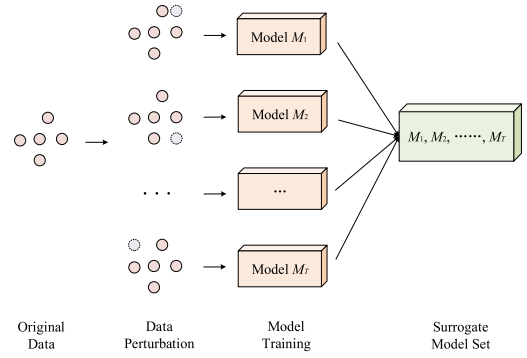


Fig. 2. Diagram of DSG.

---

#### Algorithm 1: DSG

---

**Input:**  $TS$ -the original training data set;  
 $T$ -the number of surrogates to be obtained.  
**Output:**  $SMS$ -the surrogate model set containing  $T$  surrogates.

```

1 Begin
2   Build a surrogate model  $M_1$  on  $TS$ .
3   // Compute  $diff_i$  of each data (for guiding later data selection)
4   For each  $x_i$  in  $TS$  Do
5     Use  $M_1$  to predict the fitness of  $x_i$ , stored as  $Y_{pre,i}$ ;
6     Calculate the difference  $diff_i = Y_{pre,i} - F(x_i)$ ; //  $F(x_i)$  is fitness of  $x_i$ 
7   End For
8   Sort the data in  $TS$  according to their  $diff$  with descending order;
9   Set  $S$  as the first 50% data of the sorted  $TS$ ;
10  Initialize  $SMS$  as  $\{M_1\}$ ;
11  For  $j = 2$  to  $T$  Do
12    Set  $K$  as an empty set;
13    For each  $x_i$  in  $S$  Do
14      Generate  $x_{new}$  through  $x_i$  by using Eq. (1);
15       $F(x_{new}) = F(x_i)$ ;
16       $K = K \cup (x_{new}, F(x_{new}))$ ;
17    End For
18     $DTS_j = TS \cup K$ ; // refer to Eq. (3)
19    Build surrogate model  $M_j$  on  $DTS_j$ ;
20     $SMS = SMS \cup \{M_j\}$ ;
21  End For
22 End

```

---

the pseudocode of DSG. As minimization problems can be easily transformed into maximization problems, Algorithm 1 is for minimization problems without loss of generality. The inputs of DSG include a set of original data,  $TS$ , and the total surrogate number to be obtained,  $T$ , while the output is a surrogate model set ( $SMS$ ) with  $T$  diverse surrogates. In the implementation,  $T$  surrogates are all RBFNNs [19] so that the algorithms can simply receive their parameters (i.e., the number and weights of neurons) and rebuild the same RBFNNs if needed. Furthermore, many research have shown that in both theory and practice, the linear combination of highly nonlinear models can dramatically decrease the variance of generation error [53], and RBFNNs are also nonlinear models.

In DSG, there are two main procedures. In the first procedure, some data of  $TS$  will be selected as  $S$  in (1). To obtain  $S$ , DSG builds the first surrogate  $M_1$  on  $TS$  and then predicts the fitness of all data in  $TS$  as  $Y_{pre}$ . After this, it computes the prediction error,  $diff = Y_{pre} - F(x)$ , for all the data. The larger the  $diff$  is, the large the prediction error is. Thus, data with larger  $diff$  will be selected to construct  $S$  for data generation. Herein, DSG will add the first half of data with larger  $diff$

**Algorithm 2: SE**


---

**Input:**  $x_{best}$ -the best solution in the evaluated data set;  
 $F(x_{best})$ -the fitness value of the  $x_{best}$ ;  
 $T$ -the number of surrogate models to be selected;  
 $OSMS$ -the original surrogate model set containing more than  $T$  surrogates.  
**Output:**  $SMS$ -the surrogate model set containing  $T$  surrogates.

```

1 Begin
2   For each surrogate  $M_i$  in  $OSMS$  Do
3     Use  $M_i$  to predict the fitness of  $x_{best}$ , denoted as  $Y_i$ ;
4     Calculate prediction error as  $Err_i = (F(x_{best}) - Y_i)^2$ ;
5   End For
6   Sort the surrogates in  $OSMS$  in ascending order according to  $Err_i$ ;
7   Select the first  $T$  surrogates in  $OSMS$  to construct  $SMS$ ;
8 End

```

---

into  $S$ . The reason for using “a half” lies in that too much will make the algorithm time consuming while too few data may be insufficient, and hence “a half” makes the balance. The second procedure of DSG is to generate DTS through data perturbations and then build surrogates based on DTS. For building each surrogate, DSG first sets  $K$  as an empty set. Then, it generates data by performing perturbations on each data in  $S$  and adds the generated data into  $K$ . After this, the DSG obtains a new dataset  $DST$  as the union of  $K$  and  $TS$  and employs the  $DST$  to build a new surrogate model. Finally, the built surrogate model will be added to  $SMS$ . The above processes will be performed iteratively until  $T$  surrogates are generated. Last but not least, as the first surrogate built only on  $TS$  is also added into the  $SMS$ , the total number of new surrogates built on different  $DSTs$  is  $T - 1$ .

*D. Selective Ensemble*

In the literature, the ensembles of surrogate models or ensemble learning methods have shown effectiveness in improving DDEAs [19], [50]. Therefore, this work also considers further enhancing the approximation accuracy through the combination of different surrogates. In this article, a straightforward SE method is proposed. That is, from a set of surrogates, the algorithm just selects some of them for use. The pseudocode of SE is shown as Algorithm 2. The inputs of SE include the best solution in the current generation ( $x_{best}$ ), the number of surrogates to be selected ( $T$ ), and the original model set containing more than  $T$  surrogates, while its output is an SMS with  $T$  surrogates. The idea of SE is to select the surrogates with better accuracy. As DDEAs care about the surrogate accuracy on promising area rather than the unpromising area, the SE employs the best solution to measure the accuracy of different surrogates. Therefore, SE first computes the prediction value given by each surrogate in  $OSMS$  on the best solution and then selects the surrogates with smaller prediction error. Although the algorithm can perform SE and reselect the surrogates during optimizations, how to design the frequency for performing SE can be problem dependent. Therefore, the SE used in this article is performed before the optimizations and its selection of surrogates is fixed during the whole optimization procedure. That is, we use the best solution in historical data to preselect the surrogate before

TABLE I  
BENCHMARK PROBLEMS

Problem	Optimum	Characteristics	Dimension
Ellipsoid	0	unimodal	10, 30, 50, 100
Rosenbrock	0	multimodal	10, 30, 50, 100
Ackley	0	multimodal	10, 30, 50, 100
Griewank	0	multimodal	10, 30, 50, 100
Rastrigin	0	multimodal	10, 30, 50, 100

the evolutionary search starts. When evaluating a new candidate solution, the average of prediction results given by all the selected surrogates will be calculated as the predicted fitness of that candidate solution.

## IV. EXPERIMENTAL STUDIES

*A. Algorithm Settings*

To test the proposed framework, we develop a specific algorithm for the experiments and companions, which adopt the GA with simulated binary crossover (GA-SBX) [54] and RBFNNs as the employed EA and the base models in the DDEA-PES framework, respectively. For configurations, all the RBFNNs are configured with  $D$  neurons, where  $D$  is the problem dimension. In addition, the maximum generation for performing GA-SBX is set as 500 and the number of surrogates built by DSG and the surrogates selected by SE is set as 200 and 100, respectively.

For comparisons, state-of-the-art algorithms are adopted as competitors: CAL-SAPSO [39], SA-COSO [28], GPEME [40], DDEA-SE [19], and MGP-SLPSO [21]. These algorithms have different characteristics. First, CAL-SAPSO that employs surrogates to make committee-based decisions shares some similar characteristics with DDEA-PES in the model combination. Second, GPEME is an efficient online DDEAs using Kriging models. Third, different from CAL-SAPSO and GPEME that are designed for small- and medium-scale optimizations, SA-COSO and MGP-SLPSO are online DDEAs for high-dimensional problems [28]. Fourth, DDEA-SE is a powerful offline DDEA, which can help to show the effectiveness of our DDEA-PES. All configurations of these DDEAs are set according to their corresponding references. In addition, the GA-SBX used in DDEA-PES is configured as the same with that used in DDEA-SE [19], which can provide more fair comparisons. Specifically, the crossover parameter, mutation parameter, and population size of GA-SBX are 1.0 and  $1/D$ , and 100, respectively, with  $D$  as the dimension of corresponding problems.

*B. Experimental Setup*

The experiments first employ commonly used benchmark functions [19] to test the proposed algorithm, where the benchmarks are presented in Table I. Although these benchmark problems seem to be simple, they cover a wide range of problem characteristics (e.g., multimodal and nonseparable) for observing the features of different optimization algorithms. Furthermore, these problems can be extremely difficult to optimize when the number of available FEs is limited. As

TABLE II  
COMPARISONS ON OPTIMIZATION RESULTS AMONG DDEA-PES AND TRADITIONAL METHODS

Problem	Dim	DDEA-PES (11D FEs)	Random sample (11D FEs)	GA-SBX (11D FEs)	GA-SBX (110D FEs)	GA-SBX (550D FEs)
Ellipsoid	10	1.34E+00±6.20E-01	1.46E+02±3.99E+01(+)	1.17E+02±2.58E+01(+)	4.72E+01±1.34E+01(+)	<b>1.15E+00±3.99E-01(-)</b>
	30	<b>5.76E+00±2.27E+00</b>	2.09E+03±1.91E+02(+)	2.06E+03±2.45E+02(+)	3.58E+02±8.54E+01(+)	8.55E+00±2.15E+00(+)
	50	1.43E+01±4.57E+00	6.34E+03±5.20E+02(+)	6.09E+03±3.94E+02(+)	6.56E+02±1.45E+02(+)	<b>1.31E+01±2.52E+00(-)</b>
	100	4.83E+02±4.94E+02	3.03E+04±1.65E+03(+)	2.67E+04±1.32E+03(+)	9.86E+02±2.18E+02(+)	<b>2.28E+01±6.27E+00(-)</b>
Rosenbrock	10	2.94E+01±7.52E+00	6.63E+02±3.16E+02(+)	5.14E+02±1.33E+02(+)	1.51E+02±5.35E+01(+)	<b>1.93E+01±5.45E+00(-)</b>
	30	7.01E+01±8.15E+00	4.83E+03±9.70E+02(+)	5.54E+03±7.71E+02(+)	6.77E+02±1.69E+02(+)	<b>4.82E+01±6.89E+00(-)</b>
	50	1.09E+02±1.06E+01	1.07E+04±1.34E+03(+)	9.77E+03±1.38E+03(+)	6.42E+02±2.07E+02(+)	<b>6.13E+01±4.08E+00(-)</b>
	100	4.98E+02±8.78E+01	2.79E+04±1.58E+03(+)	2.40E+04±1.59E+03(+)	4.51E+02±7.75E+01(-)	<b>1.08E+02±2.96E+00(-)</b>
Ackley	10	<b>6.40E+00±1.44E+00</b>	1.91E+01±1.12E+00(+)	1.88E+01±6.81E-01(+)	1.67E+01±1.08E+00(+)	8.68E+00±1.26E+00(+)
	30	<b>4.82E+00±3.63E-01</b>	2.04E+01±1.77E-01(+)	2.05E+01±1.95E-01(+)	1.66E+01±7.90E-01(+)	6.36E+00±7.70E-01(+)
	50	<b>4.81E+00±4.20E-01</b>	2.06E+01±1.13E-01(+)	2.06E+01±1.33E-01(+)	1.47E+01±9.76E-01(+)	4.83E+00±3.48E-01(≈)
	100	5.18E+00±6.22E-01	2.08E+01±4.63E-02(+)	2.07E+01±5.64E-02(+)	1.09E+01±5.92E-01(+)	<b>4.36E+00±3.32E-01(-)</b>
Griewank	10	<b>1.31E+00±1.36E-01</b>	1.13E+02±1.61E+01(+)	9.68E+01±2.48E+01(+)	2.90E+01±8.59E+00(+)	1.74E+00±3.00E-01(+)
	30	<b>1.25E+00±9.28E-02</b>	5.06E+02±3.16E+01(+)	5.12E+02±5.75E+01(+)	9.52E+01±2.28E+01(+)	3.52E+00±8.87E-01(+)
	50	<b>1.33E+00±9.22E-02</b>	9.52E+02±4.80E+01(+)	8.91E+02±7.20E+01(+)	8.12E+01±1.57E+01(+)	3.05E+00±5.30E-01(+)
	100	<b>3.72E+00±1.93E+00</b>	2.19E+03±6.76E+01(+)	1.91E+03±7.85E+01(+)	7.35E+01±1.52E+01(+)	3.73E+00±2.32E-01(≈)
Rastrigin	10	5.99E+01±1.76E+01	1.04E+02±1.29E+01(+)	9.35E+01±1.02E+01(+)	7.45E+01±1.01E+01(+)	<b>4.73E+01±7.72E+00(-)</b>
	30	<b>1.48E+02±2.99E+01</b>	3.93E+02±1.74E+01(+)	3.95E+02±1.69E+01(+)	2.80E+02±1.98E+01(+)	2.19E+02±1.15E+01(+)
	50	<b>2.35E+02±4.84E+01</b>	7.05E+02±2.38E+01(+)	6.67E+02±2.39E+01(+)	4.62E+02±2.81E+01(+)	3.83E+02±1.98E+01(+)
	100	<b>8.24E+02±1.25E+02</b>	1.51E+03±3.85E+01(+)	1.41E+03±4.62E+01(+)	9.16E+02±2.91E+01(+)	8.70E+02±2.46E+01(+)
+/-/≈	NA	20/0/0	20/0/0	20/0/0	9/2/9	

for the experiments,  $11 \times D$  is the maximum number of available FEs for each algorithm, where  $D$  is the corresponding problem dimension. Especially for offline DDEAs,  $11 \times D$  data are sampled by Latin hypercube sampling (LHS) [55] before their evolution and after this, no real FEs will be performed. In order to reduce the statistical errors, the average results over 25 independent runs are used for comparisons. For the purpose of clearer comparisons, both the average and standard deviation of the optimization error are presented. Besides, Wilcoxon's rank-sum tests with a significant level  $\alpha = 0.05$  are performed to make the comparisons statistically sound, the proposed algorithm performs significantly better than, similar to, and significantly worse than the DDEA-PES which are represented by the symbols “+,” “≈,” and “-,” respectively.

### C. Effectiveness of DDEA-PES

This part accesses the effectiveness of data-driven methods in DDEA-PES. In the experiments, the proposed methods are compared with GA-SBX (without surrogates) and random sample method ( $11 \times D$  data sampled by LHS). The GA-SBX and DDEA-PES are only different in that GA-SBX only uses real FEs for evolution while DDEA-PES only employs surrogates. In addition, the random sample method is actually the offline data utilized in DDEA-PES. This can serve as a baseline to compare the DDEA-PES. Besides, to observe the strength of DDEA-PES, the GA-SBX using  $110 \times D$  and  $550 \times D$  FEs is also employed for comparisons, where the DDEA-PES only consumes  $11 \times D$  FEs to obtain offline data.

The comparison results are provided in Table II, with the best results marked in bold. Table II suggests the effectiveness of DDEA-PES. Based on Wilcoxon's rank-sum tests, the DDEA-PES outperforms both the random sample method and the GA-SBX (with 11D Fes) on all the problems. Furthermore,

TABLE III  
COMPARISONS BETWEEN OFFLINE DATA-DRIVEN ALGORITHMS

Problem	Dim	DDEA-PES	DDEA-SE
Ellipsoid	10	1.34E+00±6.20E-01	<b>1.02E+00±4.90E-01(-)</b>
	30	5.76E+00±2.27E+00	<b>5.09E+00±1.30E+00(-)</b>
	50	<b>1.43E+01±4.57E+00</b>	1.51E+01±4.63E+00(+)
	100	4.83E+02±4.94E+02	<b>3.12E+02±6.13E+01(-)</b>
Rosenbrock	10	<b>2.94E+01±7.52E+00</b>	2.95E+01±5.04E+00(≈)
	30	7.01E+01±8.15E+00	<b>5.67E+01±5.34E+00(-)</b>
	50	1.09E+02±1.06E+01	<b>8.41E+01±4.05E+00(-)</b>
	100	4.98E+02±8.78E+01	<b>2.65E+02±2.48E+01(-)</b>
Ackley	10	<b>6.40E+00±1.44E+00</b>	<b>6.40E+00±1.14E+00(≈)</b>
	30	<b>4.82E+00±3.63E-01</b>	4.83E+00±5.10E-01(≈)
	50	<b>4.81E+00±4.20E-01</b>	4.82E+00±3.85E-01(≈)
	100	<b>5.18E+00±6.22E-01</b>	7.27E+00±7.09E-01(+)
Griewank	10	<b>1.31E+00±1.36E-01</b>	<b>1.31E+00±1.46E-01(-)</b>
	30	<b>1.25E+00±9.28E-02</b>	1.34E+00±7.46E-02(+)
	50	<b>1.33E+00±9.22E-02</b>	1.94E+00±2.45E-01(+)
	100	<b>3.72E+00±1.93E+00</b>	1.81E+01±2.12E+00(+)
Rastrigin	10	<b>5.99E+01±1.76E+01</b>	6.59E+01±1.89E+00(+)
	30	<b>1.48E+02±2.99E+01</b>	1.85E+02±1.61E+01(+)
	50	2.35E+02±4.84E+01	<b>1.87E+02±3.03E+01(-)</b>
	100	<b>8.24E+02±1.25E+02</b>	<b>8.11E+02±8.26E+01(≈)</b>
+/-/≈	NA	7/6/7	

the DDEA-PES can perform generally better than and similarly to the GA-SBX with 110D FEs and 550D FEs, respectively. That is, it can use about 10% or 2% budgets of FEs to generate better or similar results when compared with GA-SBX without surrogates. This has shown the effectiveness of DDEA-PES.

### D. Comparisons With Offline DDEAs

This part compares the DDEA-PES with DDEA-SE on all the benchmark problems, of which results are provided in Table III. It can be seen that although DDEA-SE is a state-of-the-art offline DDEA, the DDEA-PES can perform better



TABLE IV  
COMPARISONS BETWEEN THE DDEA-PES AND ONLINE DATA-DRIVEN ALGORITHMS ON LOW- AND MEDIUM-DIMENSIONAL PROBLEMS

Problem	$D$	Metric	DDEA-PES	CAL-SAPSO	GPEME
Ellipsoid	10	Mean	<b>1.34E+00</b>	1.42E+00 ( $\approx$ )	3.64E+01 (+)
		Std.	6.20E-01	8.30E-01	1.68E+01
	30	Mean	5.76E+00	<b>4.05E+00 (-)</b>	1.19E+03 (+)
		Std.	2.27E+00	1.11E+00	2.12E+02
Rosenbrock	10	Mean	2.94E+01	<b>1.74E+01 (-)</b>	1.80E+02(+)
		Std.	7.52E+00	3.90E+00	6.54E+01
	30	Mean	<b>7.01E+01</b>	5.18E+01 (+)	2.68E+03 (+)
		Std.	8.15E+00	1.01E+01	8.21E+02
Ackley	10	Mean	<b>6.40E+00</b>	2.01E+01 (+)	1.41E+01 (+)
		Std.	1.44E+00	2.40E-01	2.30E+00
	30	Mean	<b>4.82E+00</b>	1.67E+01 (+)	1.93E+01 (+)
		Std.	3.63E-01	2.70E-01	3.00E-01
Griewank	10	Mean	1.31E+00	<b>1.29E+00 (<math>\approx</math>)</b>	2.95E+01 (+)
		Std.	1.36E-01	1.40E-01	1.17E+01
	30	Mean	1.25E+00	<b>1.08E+00 (-)</b>	2.71E+02 (+)
		Std.	9.28E-02	3.87E-02	4.72E+01
Rastrigin	10	Mean	<b>5.99E+01</b>	8.87E+01 (+)	7.08E+01 (+)
		Std.	1.76E+01	2.15E+01	1.23E+01
	30	Mean	1.48E+02	<b>8.54E+01 (-)</b>	3.02E+02 (+)
		Std.	2.99E+01	1.76E+01	2.86E+01
		+/ $\approx$ /-	NA	4/1/5	10/0/0

than and similar to DDEA-SE on 7 and 4 of the 20 problems, respectively. Furthermore, the DDEA-SE builds 2000 surrogates for model management and the DDEA-PES only builds 200 surrogates, which suggests the efficiency of the DDEA-PES.

#### E. Comparisons With Online DDEAs

In this part, DDEA-PES is compared with state-of-the-art online DDEAs. Considering that online algorithms are proposed for different problems, that is, CAL-SAPSO and GPEME for low- and medium-dimensional problems while SA-COSO and MGP-SLPSO for high-dimensional problems, we divided the comparisons into two parts, problems with 10 and 30 dimensions and problems with 50 and 100 dimensions. Hence, the CAL-SAPSO and GPEME are adopted in 10- and 30-D problems while the SA-COSO and MGP-SLPSO are employed in 50- and 100-D problems.

Table IV presents the comparison results on low and medium problems. When compared with GPEME, the DDEA-PES shows its strengths by significantly outperforming GPEME on all the ten benchmark problems. When compared with CAL-SAPSO, the DDEA-PES can still obtain the best results (in bold) on five of the ten problems. As the DDEA-PES is implemented in an offline version, it can have better performance when implemented in an online version which can evaluate candidate solutions to update surrogate models during the optimization process. In short, the comparisons with CAL-SAPSO and GPEME have shown the potential of the DDEA-PES.

Table V presents the comparison results on high-dimensional problems. The results show that the DDEA-PES significantly outperforms SA-COSO on nine compared problems and is only significantly worse on one problem, which suggests the good performance of DDEA-PES on high-dimensional problems. When compared with MGP-SLPSO,

TABLE V  
COMPARISONS BETWEEN THE DDEA-PES AND ONLINE DATA-DRIVEN ALGORITHMS ON HIGH-DIMENSIONAL PROBLEMS

Problem	$D$	Metric	DDEA-PES	SA-COSO	MGP-SLPSO
Ellipsoid	50	Mean	1.34E+00	2.31E+02(+)	<b>8.42E-02(-)</b>
		Std.	6.20E-01	6.38E+01	4.70E-02
	100	Mean	5.76E+00	9.58E+02(+)	<b>8.32E-02(-)</b>
		Std.	2.27E+00	2.36E+02	3.51E-01
Rosenbrock	50	Mean	<b>2.94E+01</b>	6.12E+02(+)	1.67E+02(+)
		Std.	7.52E+00	2.09E+02	1.44E+01
	100	Mean	<b>7.01E+01</b>	2.08E+03(+)	5.23E+02(+)
		Std.	8.15E+00	4.48E+02	2.87E+01
Ackley	50	Mean	<b>6.40E+00</b>	1.32E+01(+)	1.25E+01(+)
		Std.	1.44E+00	8.60E-01	1.56E+00
	100	Mean	<b>4.82E+00</b>	1.59E+01(+)	1.45E+01(+)
		Std.	3.63E-01	6.00E-01	1.01E+00
Griewank	50	Mean	1.31E+00	2.69E+01(+)	<b>3.41E-01(-)</b>
		Std.	1.36E-01	4.68E+00	1.95E-01
	100	Mean	1.25E+00	7.42E+01(+)	<b>7.96E-02(-)</b>
		Std.	9.28E-02	1.65E+01	3.53E-02
Rastrigin	50	Mean	<b>5.99E+01</b>	4.09E+02(+)	3.67E+02(+)
		Std.	1.76E+01	3.01E+01	5.70E+01
	100	Mean	1.48E+02	8.22E+02(+)	<b>5.52E+02(-)</b>
		Std.	2.99E+01	6.90E+01	8.85E+01
		+/ $\approx$ /-	NA	9/0/1	5/0/5

the DDEA-PES performs significantly better on five and significantly worse on five problems, showing that DDEA-PES, in general, is competitive with MGP-SLPSO. Moreover, DDEA-PES obtains better results on Rosenbrock and Ackley functions while MGP-SLPSO on Ellipsoid and Griewank functions, suggesting that they are efficient for different problems. In a word, the above experiments have shown the strengths of DDEA-PES when compared with online data-driven algorithms.

#### F. Contribution Analysis of Different Components

This part further studies the contributions and influences of the proposed DSG and SE individually. For this, DDEA-PES is compared with its variants without DSG, SE, or both of them. According to their components, these variants are denoted as DDEA-PES without DSG (DDEA-PES-without-DSG), DDEA-PES without SE (DDEA-PES-without-SE), and DDEA-PES without both DSG and SE (DDEA-PES-without-DSG-SE). All these variants use the same optimizer as that used in DDEA-PES.

The comparison results are provided in Table VI. Based on Wilcoxon's rank-sum test, the original DDEA-PES significantly outperforms DDEA-PES-without-DSG, DDEA-PES-without-SE, and DDEA-PES-without-DSG-SE on 13, 16, and 20 problems of the 20 tested problems, respectively. Moreover, over the 20 problems, DDEA-PES can obtain the best results (as marked in bold) on 13 problems. These results show that both the DSG and SE have their contributions to the great performance of DDEA-PES and removing any of them will decrease the algorithm performance.

#### G. Effects of Different Area Sizes for Data Perturbation

To further study the effect of different area sizes for data perturbations, DDEA-PES is compared with its variants with different  $l$  for perturbations, where the original  $l$  is

TABLE VI  
COMPARISONS BETWEEN ALGORITHM VARIANTS WITH OR WITHOUT DSG AND SE

Problem	$D$	DDEA-PES	DDEA-PES-without-DSG	DDEA-PES-without-SE	DDEA-PES-without-DSG-SE
Ellipsoid	10	<b>1.34E+00±6.20E-01</b>	1.43E+00±6.89E-01(+)	<b>1.34E+00±7.30E-01(+)</b>	3.01E+00±1.67E+00(+)
	30	<b>5.76E+00±2.27E+00</b>	8.81E+00±3.83E+00(+)	7.35E+00±2.28E+00(+)	1.83E+01±6.93E+00(+)
	50	<b>1.43E+01±4.57E+00</b>	1.58E+01±2.31E+00(+)	1.91E+01±6.38E+00(+)	6.76E+01±3.74E+01(+)
	100	<b>4.83E+02±4.94E+02</b>	5.84E+02±6.09E+02(+)	4.94E+02±2.63E+02(+)	1.24E+03±9.66E+02(+)
Rosenbrock	10	<b>2.94E+01±7.52E+00</b>	3.71E+01±6.42E+00(+)	3.20E+01±7.23E+00(+)	4.64E+01±1.46E+01(+)
	30	7.01E+01±8.15E+00	7.05E+01±7.38E+00(≈)	<b>6.85E+01±8.07E+00(-)</b>	9.59E+01±2.17E+01(+)
	50	1.09E+02±1.06E+01	<b>9.70E+01±6.94E+00(≈)</b>	1.05E+02±1.63E+01(≈)	1.56E+02±3.77E+01(+)
	100	4.98E+02±8.78E+01	<b>3.99E+02±1.30E+02(-)</b>	5.45E+02±1.43E+02(+)	8.41E+02±7.14E+02(+)
Ackley	10	<b>6.40E+00±1.44E+00</b>	6.51E+00±1.92E+00(+)	6.45E+00±1.07E+00(≈)	7.71E+00±1.47E+00(+)
	30	<b>4.82E+00±3.63E-01</b>	5.13E+00±3.48E-01(+)	5.32E+00±5.31E-01(+)	7.39E+00±1.01E+00(+)
	50	4.81E+00±4.20E-01	<b>4.75E+00±3.33E-01(≈)</b>	4.78E+00±2.32E-01(≈)	6.95E+00±8.97E-01(+)
	100	5.18E+00±6.22E-01	<b>4.78E+00±2.08E-01(≈)</b>	5.01E+00±6.67E-01(+)	8.15E+00±9.82E-01(+)
Griewank	10	<b>1.31E+00±1.36E-01</b>	1.50E+00±1.17E-01(+)	1.36E+00±2.23E-01(+)	2.14E+00±7.71E-01(+)
	30	<b>1.25E+00±9.28E-02</b>	1.30E+00±9.27E-02(+)	1.41E+00±1.73E-01(+)	2.68E+00±5.63E-01(+)
	50	<b>1.33E+00±9.22E-02</b>	1.57E+00±3.50E-01(+)	1.57E+00±1.69E-01(+)	3.61E+00±1.45E+00(+)
	100	3.72E+00±1.93E+00	<b>2.76E+01±2.85E+01(+)</b>	1.68E+01±2.13E+01(+)	8.97E+01±1.00E+02(+)
Rastrigin	10	<b>5.99E+01±1.76E+01</b>	8.02E+01±2.12E+01(+)	6.98E+01±2.53E+01(+)	8.85E+01±2.11E+01(+)
	30	<b>1.48E+02±2.99E+01</b>	1.51E+02±4.49E+01(≈)	1.57E+02±3.61E+01(+)	2.28E+02±5.11E+01(+)
	50	2.35E+02±4.84E+01	2.25E+02±5.78E+01(-)	<b>2.24E+02±5.34E+01(-)</b>	3.98E+02±8.86E+01(+)
	100	<b>8.24E+02±1.25E+02</b>	9.61E+02±1.32E+01(+)	8.76E+02±1.90E+02(+)	1.04E+03±8.00E+01(+)
+/-	NA	13/5/2	16/2/2	20/0/0	

TABLE VII  
COMPARISON AMONG DDEA-PES VARIANTS WITH DIFFERENT MAXIMUM LENGTHS FOR DATA PERTURBATION

Problem	Dim	$l = l_0$	$l = 0$ (generate duplicate data)	$l = 10^{-2} \cdot l_0$	$l = 10^2 \cdot l_0$	$l = 10^4 \cdot l_0$
Ellipsoid	10	1.34E+00±6.20E-01	1.01E+00±2.87E-01(-)	1.07E+00±4.65E-01(-)	<b>9.06E-01±4.86E-01(-)</b>	1.39E+00±1.05E+00(≈)
	30	<b>5.76E+00±2.27E+00</b>	6.55E+00±1.86E+00(+)	7.46E+00±2.79E+00(+)	6.24E+00±8.62E-01(+)	7.44E+00±3.33E+00(+)
	50	<b>1.43E+01±4.57E+00</b>	<b>1.43E+01±1.80E+00(≈)</b>	1.51E+01±3.10E+00(+)	1.58E+01±4.28E+00(+)	1.45E+01±3.90E+00(≈)
	100	4.83E+02±4.94E+02	<b>1.49E+02±9.20E+01(-)</b>	3.53E+02±2.16E+02(+)	5.83E+02±5.07E+02(+)	4.00E+02±2.99E+02(-)
Rosenbrock	10	<b>2.94E+01±7.52E+00</b>	3.11E+01±8.47E+00(+)	3.18E+01±7.62E+00(+)	3.65E+01±5.09E+00(+)	3.33E+01±6.88E+00(+)
	30	7.01E+01±8.15E+00	7.54E+01±2.18E+00(+)	7.62E+01±5.49E+00(+)	<b>6.71E+01±6.68E+00(-)</b>	7.10E+01±7.56E+00(+)
	50	1.09E+02±1.06E+01	1.05E+02±1.07E+01(≈)	1.02E+02±8.58E+00(≈)	<b>9.73E+01±6.49E+00(-)</b>	1.07E+02±1.58E+01(≈)
	100	4.98E+02±8.78E+01	6.67E+02±1.88E+02(+)	5.63E+02±2.25E+02(+)	5.47E+02±1.88E+02(+)	<b>4.87E+02±1.59E+02(-)</b>
Ackley	10	6.40E+00±1.44E+00	6.87E+00±1.42E+00(+)	6.24E+00±1.15E+00(-)	<b>6.10E+00±3.92E-01(-)</b>	6.67E+00±1.21E+00(+)
	30	<b>4.82E+00±3.63E-01</b>	5.40E+00±2.41E-01(+)	5.12E+00±4.88E-01(+)	5.16E+00±2.14E-01(+)	5.38E+00±4.98E-01(+)
	50	<b>4.81E+00±4.20E-01</b>	5.05E+00±4.11E-01(+)	5.21E+00±4.11E-01(+)	4.96E+00±2.25E-01(-)	4.99E+00±2.56E-01(+)
	100	5.18E+00±6.22E-01	5.66E+00±5.64E-01(+)	5.44E+00±1.95E-01(+)	<b>4.26E+00±2.81E-01(-)</b>	5.57E+00±4.55E-01(+)
Griewank	10	1.31E+00±1.36E-01	1.23E+00±1.47E-01(-)	1.24E+00±1.33E-01(-)	1.32E+00±1.50E-01(≈)	<b>1.22E+00±1.51E-01(-)</b>
	30	<b>1.25E+00±9.28E-02</b>	1.37E+00±1.67E-01(+)	1.27E+00±9.32E-02(≈)	<b>1.25E+00±1.25E+00(≈)</b>	1.25E+00±1.23E-01(≈)
	50	<b>1.33E+00±9.22E-02</b>	1.40E+00±1.77E-01(+)	1.34E+00±4.15E-02(≈)	1.34E+00±8.89E-02(≈)	1.34E+00±9.79E-02(≈)
	100	3.72E+00±1.93E+00	7.75E+00±5.23E+00(+)	<b>2.75E+00±8.12E-01(-)</b>	3.14E+00±1.29E+00(-)	3.69E+00±1.51E+00(≈)
Rastrigin	10	<b>5.99E+01±1.76E+01</b>	6.41E+01±3.25E+01(+)	6.07E+01±1.31E+01(≈)	6.86E+01±1.83E+01(+)	6.88E+01±2.24E+01(+)
	30	1.48E+02±2.99E+01	1.65E+02±2.77E+01(+)	<b>1.44E+02±2.88E+01(≈)</b>	1.53E+02±3.61E+01(+)	1.48E+02±2.99E+01(≈)
	50	2.35E+02±4.84E+01	2.11E+02±3.93E+01(-)	2.12E+02±5.05E+01(-)	<b>1.82E+02±2.55E+01(-)</b>	2.26E+02±5.71E+01(≈)
	100	<b>8.24E+02±1.25E+02</b>	8.90E+02±1.47E+02(+)	9.81E+02±1.40E+02(+)	9.57E+02±1.95E+02(+)	9.71E+02±1.02E+02(+)
+/-	NA	14/2/4	9/5/6	9/3/8	9/4/7	

denoted as  $l_0$  [refer to (1)]. The comparison results provided in Table VII show that different  $l$  can be suitable for different problems and different dimensions. For example, DDEA-PES with  $l = 10^2 \cdot l_0$  outperforms  $l = 10^{-2} \cdot l_0$  on the 10-D Ellipsoid problem while it performs worse than  $l = 10^{-2} \cdot l_0$  on the 100-D Ellipsoid problem. Also,  $l = 10^2 \cdot l_0$  performs better and worse than  $l = 10^4 \cdot l_0$  on the 10-D Ellipsoid problem and 100-D Rosenbrock problem, respectively. Furthermore, as  $l = l_0$  can, in general, outperform other  $l$  values,  $l$  is recommended to be configured according to (2). In summary, the proposed data perturbation can improve the optimization accuracy of DDEAs.

#### H. Effects of Different Criteria for Constructing Dataset

This part considers the effects of different criteria for constructing the dataset  $S$  (refer to Algorithm 1). As the employed criterion of the original DDEA-PES is  $\text{diff} = Y_{\text{pre}} - F(x)$ , its variants with  $\text{diff} = \|Y_{\text{pre}} - F(x)\|^2$ ,  $\text{diff} = F(x)$ , and  $\text{diff} = -F(x)$  are employed for comparisons. The results provided in Table VIII show that the original  $\text{diff}$  outperforms others. Based on Wilcoxon's rank-sum tests on the 20 problems, the original  $\text{diff}$  can perform significantly better than  $\text{diff} = \|Y_{\text{pre}} - F(x)\|^2$ ,  $\text{diff} = F(x)$ , and  $\text{diff} = -F(x)$  on 17, 14, and 20 problems, respectively. Furthermore, the original  $\text{diff}$  obtains the best average results (marked in bold) on 11 of

TABLE VIII  
COMPARISON AMONG DDEA-PES VARIANTS WITH DIFFERENT CRITERIA FOR DATA PERTURBATION

Problem	Dim	Original	$diff = \ Y_{pre} - F(x)\ ^2$	$diff = F(x)$	$diff = -F(x)$
Ellipsoid	10	1.34E+00±6.20E-01	1.50E+00±5.71E-01(+)	<b>1.32E+00±4.59E-01(≈)</b>	2.18E+00±1.16E+00(+)
	30	<b>5.76E+00±2.27E+00</b>	8.18E+00±4.02E+00(+)	6.92E+00±3.97E+00(+)	7.48E+00±2.22E+00(+)
	50	<b>1.43E+01±4.57E+00</b>	2.40E+01±1.21E+01(+)	1.55E+01±3.26E+00(+)	1.59E+01±2.12E+00(+)
	100	<b>4.83E+02±4.94E+02</b>	1.82E+03±1.16E+03(+)	5.11E+02±5.44E+02(+)	5.01E+02±6.20E+02(+)
Rosenbrock	10	<b>2.94E+01±7.52E+00</b>	3.63E+01±7.87E+00(+)	3.49E+01±1.10E+01(+)	3.99E+01±5.97E+00(+)
	30	<b>7.01E+01±8.15E+00</b>	7.10E+01±9.58E+00(+)	7.56E+01±4.80E+00(+)	7.70E+01±8.85E+00(+)
	50	1.09E+02±1.06E+01	<b>1.08E+02±1.02E+01(≈)</b>	1.17E+01±5.56E+00(+)	1.15E+02±1.42E+01(+)
	100	4.98E+02±8.78E+01	4.76E+02±1.42E+02(-)	<b>2.39E+02±6.17E+01(-)</b>	7.10E+02±3.23E+02(+)
Ackley	10	<b>6.40E+00±1.44E+00</b>	6.82E+00±1.26E+00(+)	5.51E+00±1.71E+00(-)	6.54E+00±1.10E+00(+)
	30	<b>4.82E+00±3.63E-01</b>	5.58E+00±5.30E-01(+)	5.10E+00±5.13E-01(+)	5.56E+00±7.45E-01(+)
	50	4.81E+00±4.20E-01	5.12E+00±3.47E-01(+)	<b>4.30E+00±3.22E-01(-)</b>	5.03E+00±4.82E-01(+)
	100	5.18E+00±6.22E-01	5.51E+00±7.15E-01(+)	<b>3.90E+00±3.76E-01(-)</b>	5.94E+00±4.73E-01(+)
Griewank	10	<b>1.31E+00±1.36E-01</b>	1.36E+00±1.81E-01(≈)	1.53E+00±1.64E-01(+)	1.46E+00±3.07E-01(+)
	30	<b>1.25E+00±9.28E-02</b>	1.40E+00±1.76E-01(+)	1.46E+00±2.00E-01(+)	1.50E+00±7.74E-02(+)
	50	<b>1.33E+00±9.22E-02</b>	1.67E+00±3.07E-01(+)	1.64E+00±1.15E-01(+)	1.82E+00±3.45E-01(+)
	100	<b>3.72E+00±1.93E+00</b>	6.60E+01±8.73E+01(+)	7.08E+01±1.10E+02(+)	1.40E+01±1.35E+01(+)
Rastrigin	10	5.99E+01±1.76E+01	7.85E+01±2.29E+01(+)	7.10E+01±2.44E+01(+)	6.71E+01±1.07E+01(+)
	30	1.48E+02±2.99E+01	1.54E+02±2.77E+01(+)	1.45E+02±7.09E+01(≈)	1.69E+02±3.11E+01(+)
	50	2.35E+02±4.84E+01	2.81E+02±6.34E+01(+)	2.72E+02±2.79E+01(+)	2.90E+02±8.53E+01(+)
	100	8.24E+02±1.25E+02	9.57E+02±1.65E+02(+)	8.64E+02±1.21E+02(+)	1.03E+03±5.72E+01(+)
+/-	NA	17/2/1	14/2/4	20/0/0	

the 20 problems. This indicates that selecting which data to perform perturbations can be crucial to the surrogate accuracy, and in general, the selection criterion employed in this article is effective and efficient.

### I. Effects of Selected Surrogate Number

This part studies the effects of the selected surrogate number for building the final ensemble model. For this, we compared the performance of DDEA-PES with different numbers of selected surrogates, including 5, 10, 15, 20, 50, 100 (the original setting), and 150, on 10- and 100-D Ellipsoid, Ackley, and Rastrigin problems, respectively. The comparison results are provided in Fig. 3, which indicates that the influence of the selected surrogate number has a strong relationship with the problem dimension. For example, 150 surrogates have similar average results with ten surrogates on the 10-D Ellipsoid problem, while 150 surrogates can outperform ten surrogates on the 100-D Ellipsoid problem. Similarly, on the Ackley problem, selecting ten surrogates is better for the 10-D case while selecting 100 surrogates is recommended for the 100-D case. These results may be due to the reason that the problem complexity increases rapidly as the dimension increases. In such circumstances, the number of selected surrogates enough for problems of low dimensions may be insufficient for problems of high dimensions. Therefore, it is suggested to use more surrogates when approximating higher dimensional and more complex problems.

### J. Aerodynamic Airfoil Design Optimization

This part employs an aerodynamic airfoil design optimization problem to test the performance of DDEA-PES in real-world applications [56]. As predicting the performance of a candidate airfoil shape design requires time-consuming computational fluid dynamics simulations, it is ideal to

employ DDEAs to solve the aerodynamic airfoil design optimization problem.

The objective of the problem is to optimize the geometry of the airfoil to maximize the lift-over-drag ratio at predefined transonic flow situations. The airfoil geometry can be defined by the base airfoil case and ten additional controlling variables of the Hicks–Henne bump functions [57], [58]. Therefore, the problem can be simply formulated as

$$\max_{\theta_1, \theta_2, \dots, \theta_{10}} C_{L/D}(M, Re, AoA, Hicks(\theta_1, \theta_2, \dots, \theta_{10})) \quad (11)$$

$$\text{s.t. } -0.001 \leq \theta_i \leq 0.001, \quad i = 1, 7 \quad (12)$$

$$-0.006 \leq \theta_i \leq 0.006, \quad i = 2, 5 \quad (13)$$

$$-0.009 \leq \theta_i \leq 0.009, \quad i = 3, 4 \quad (14)$$

$$-0.002 \leq \theta_i \leq 0.002, \quad i = 6, 10 \quad (15)$$

$$-0.007 \leq \theta_i \leq 0.007, \quad i = 8, 9 \quad (16)$$

where  $C_{L/D}$  is the fitness to denote the lift-over-drag ratio,  $\theta_1, \theta_2, \dots, \theta_{10}$  are the ten decision variables,  $Hicks()$  represents the Hicks–Henne bump function set, and the predefined parameters  $M$ ,  $Re$ , and  $AoA$  are the Mach number, Reynolds number, and the Angle of Attack, respectively [57], [58].

In the experiments, the NACA 0012 airfoil [59] is adopted as the base case and the parameters  $M$ ,  $Re$ , and  $AoA$  are set as 0.5,  $5 \times 10^6$ , and  $4^\circ$ , respectively, in (11). The fitness value  $C_{L/D}$  of each airfoil design is obtained by the simulation in the software Xfoil [60]. For this 10- $D$  problem, the maximum number of evaluations is 110 (i.e.,  $11 \times D$ ) for all algorithms, where offline DDEAs will sample 110 data before the optimization while online DDEAs spend 50 evaluations (i.e.,  $5 \times D$ ) for initial sampling if their algorithms do not have specific settings. All algorithms are run 25 times and the average results are provided for comparisons. In addition, as SA-COSO and MGP-SLPSO are developed for high-dimensional problems, they are not adopted in this part. Instead, in this comparison, we additionally add three online

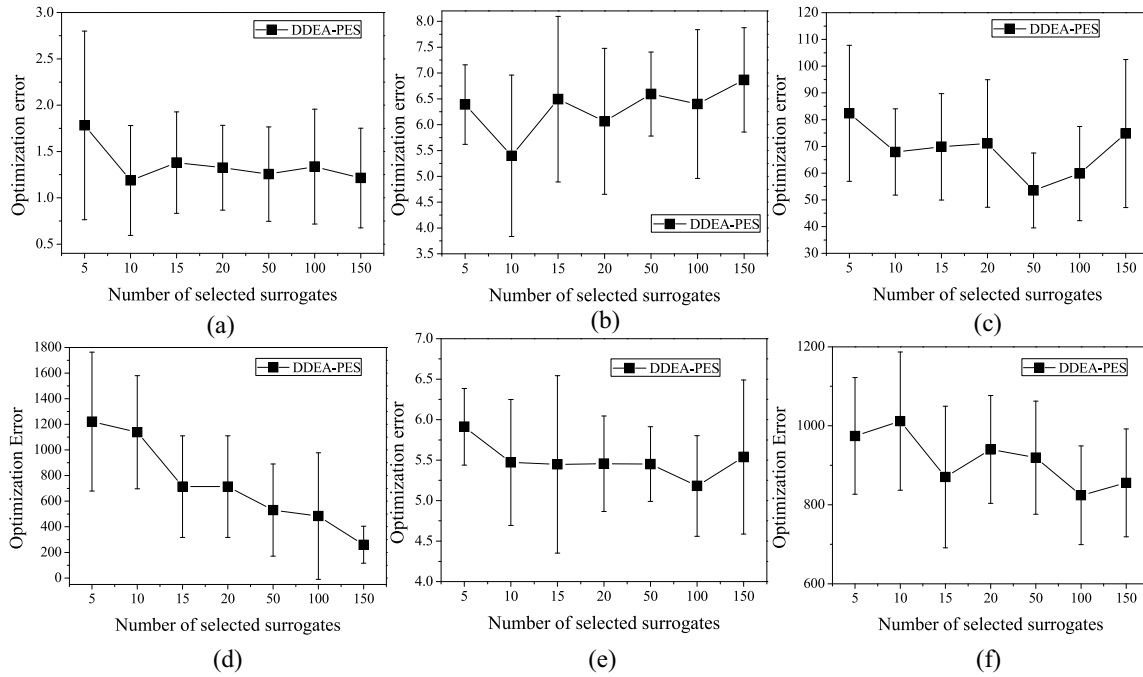


Fig. 3. Optimization error obtained by DDEA-PES variants with different numbers of selected surrogates. (a) Ellipsoid problem at 10 dimensions. (b) Ackley problem at 10 dimensions. (c) Rastrigin problem of 10 dimensions. (d) Ellipsoid problem at 100 dimensions. (e) Ackley problem at 100 dimensions. (f) Rastrigin problem of 100 dimensions.

TABLE IX  
RESULTS OF LIFT OVER DRAG RATIO ON THE AERODYNAMIC  
AIRFOIL DESIGN PROBLEM AFTER 25 RUNS

Algorithm	Best	Median	Mean	Std.
DDEA-PES	<b>105.03</b>	<b>100.01</b>	<b>99.68</b>	1.60
DDEA-SE	99.60	98.81	98.92	<b>1.31</b>
CAL-SAPSO	100.69	99.19	99.33	1.67
GPME	96.84	94.55	93.80	2.49
EAS-SM3	99.05	93.90	94.03	2.87
EAS-SM5	104.55	95.01	94.72	4.95
EAS-SM12	99.81	98.80	98.71	1.74
GA-SBX	93.35	76.81	77.17	9.12
Random sample	92.22	72.30	74.46	7.39

DDEAs that have been examined on real-world problems. They are EAS-SM3, EAS-SM5, and EAS-SM12, which are the best three algorithms among 12 algorithms on a real-world application optimization [32]. Both the EAS-SM3 and EAS-SM5 employ a single surrogate, respectively, based on a cubic RBFNN and a Kriging model with Gaussian correlation and first-order polynomial, while the EAS-SM12 adopts the ensemble surrogates with optimal weights.

The experimental results are provided in Table IX. Among the tested algorithms, the proposed DDEA-PES produced the best  $C_{L/D}$  as 105.03 and obtained the best performance in terms of the median and mean results, showing the effectiveness of DDEA-PES. Moreover, the geometry and pressure coefficient of the original NACA 0012 and the best-optimized airfoil obtained by DDEA-PES are plotted in Fig. 4 for visualizations. It can be seen that our optimized airfoil can have a smoother change of pressure coefficient (e.g., when

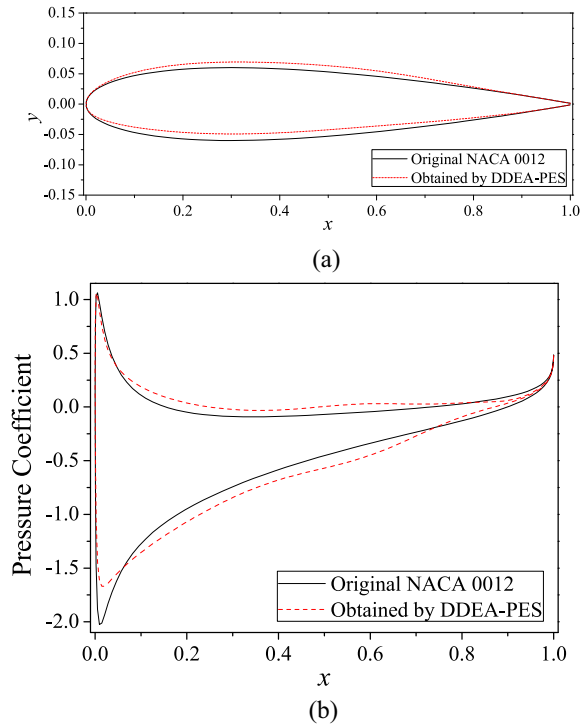


Fig. 4. Geometry and pressure coefficients of the original NACA 0012 and the best-optimized airfoils. (a) Geometry of the airfoil. (b) Pressure coefficient of the airfoil.

$x = 0.05$ ) than the original airfoil, which can lead to a considerable improvement of airfoil performance. Overall, the above results have verified the effectiveness of DDEA-PES in the real-world airfoil design application problem.

## V. CONCLUSION

In this article, a novel and efficient framework called DDEA-PES is proposed with two efficient mechanisms. The first is the DSG that can generate diverse surrogates based on a limited dataset, which can make better use of the limited data and improve surrogate qualities. The second is the SE that selects some of the prebuilt surrogates to form the final ensemble model in order to further improve the model's effectiveness and efficiency.

For future work, the proposed algorithm will be extended to more complicated real-world application problems. Moreover, other types of surrogates will be attempted as the base model to further improve the algorithm accuracy and efficiency. In addition, other methods such as semisupervised regression will be studied to better deal with the data shortage problems when building surrogates.

## REFERENCES

- [1] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 536–549, Feb. 2020.
- [2] T. Chugh, K. Sindhya, K. Miettinen, Y. Jin, T. Kratky, and P. Makkonen, "Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 1541–1548.
- [3] T. Chugh, N. Chakraborti, K. Sindhya, and Y. Jin, "A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem," *Mater. Manuf. Process.*, vol. 32, no. 10, pp. 1172–1178, 2017.
- [4] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 442–458, Jun. 2019.
- [5] Z.-J. Wang, Z.-H. Zhan, S. Kwong, H. Jin, and J. Zhang, "Adaptive granularity learning distributed particle swarm optimization for large-scale optimization," *IEEE Trans. Cybern.*, early access, Mar. 27, 2020, doi: [10.1109/TCYB.2020.2977956](https://doi.org/10.1109/TCYB.2020.2977956).
- [6] X. Zhang, K. J. Du, Z.-H. Zhan, S. Kwong, T. L. Gu, and J. Zhang, "Cooperative co-evolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties," *IEEE Trans. Cybern.*, early access, Sep. 20, 2019, doi: [10.1109/TCYB.2019.2937565](https://doi.org/10.1109/TCYB.2019.2937565).
- [7] Z.-J. Wang *et al.*, "Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2715–2729, Jun. 2020.
- [8] X.-F. Liu *et al.*, "Neural network-based information transfer for dynamic optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1557–1570, May 2020.
- [9] Z.-J. Wang *et al.*, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 114–128, Feb. 2020.
- [10] H. Zhao *et al.*, "Local binary pattern based adaptive differential evolution for multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3343–3357, Jul. 2020.
- [11] Z.-G. Chen, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, Aug. 2020.
- [12] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. Chung, and Y. H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.
- [13] J. Sun, S. Gao, H. Dai, J. Cheng, M. Zhou, and J. Wang, "Bi-objective elite differential evolution algorithm for multivalued logic networks," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 233–246, Jan. 2020.
- [14] S. Gao, S. Song, J. Cheng, Y. Todo, and M. Zhou, "Incorporation of solvent effect into multi-objective evolutionary algorithm for improved protein structure prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 4, pp. 1365–1378, Jul./Aug. 2018.
- [15] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, Aug. 2019.
- [16] Z.-H. Zhan *et al.*, "Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 704–716, Mar. 2017.
- [17] Z.-H. Zhan, Z.-J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Trans. Cybern.*, early access, Oct. 21, 2019, doi: [10.1109/TCYB.2019.2944873](https://doi.org/10.1109/TCYB.2019.2944873).
- [18] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 939–952, Apr. 2016.
- [19] H. Wang, Y. Jin, C. Sun, and J. Doherty, "Offline data-driven evolutionary optimization using selective surrogate ensembles," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 203–216, Apr. 2019.
- [20] L. Gu, "A comparison of polynomial based regression models in vehicle safety analysis," in *Proc. ASME Design Eng. Tech. Conf.*, vol. 33, 2001, pp. 3444–3448.
- [21] 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 Trans. Evol. Comput.*, vol. 23, no. 3, pp. 459–472, Jun. 2019.
- [22] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005.
- [23] D. Guo, T. Chai, J. Ding, and Y. Jin, "Small data driven evolutionary multi-objective optimization of fused magnesium furnaces," in *Proc. IEEE Symp. Series Comput. Intell.*, 2016, pp. 1–8.
- [24] Z. Zhou, Y. S. Ong, M. H. Nguyen, and D. Lim, "A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 2832–2839.
- [25] 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 Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, Feb. 2018.
- [26] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 481–494, Oct. 2002.
- [27] S. Z. Martínez and C. A. Coello Coello, "MOEA/D assisted by RBF networks for expensive multi-objective optimization problems," in *Proc. ACM Genet. Evol. Comput. Conf.*, 2013, pp. 1405–1412.
- [28] 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.
- [29] Y. Wang, D. Yin, S. Yang, and G. Sun, "Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1642–1656, May 2019.
- [30] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Comput.*, vol. 19, no. 6, pp. 1461–1475, 2015.
- [31] M. Babaei and I. Pan, "Performance comparison of several response surface surrogate models and ensemble methods for water injection optimization under uncertainty," *Comput. Geosci.*, vol. 91, pp. 19–32, Jun. 2016.
- [32] V. Christelis, G. Kopsiaftis, and A. Mantoglou, "Performance comparison of multiple and single surrogate models for pumping optimization of coastal aquifers," *Hydrol. Sci. J.*, vol. 64, no. 3, pp. 336–349, 2019.
- [33] V. Christelis, R. G. Regis, and A. Mantoglou, "Surrogate-based pumping optimization of coastal aquifers under limited computational budgets," *J. Hydroinform.*, vol. 20, no. 1, pp. 164–176, 2018.
- [34] V. Christelis and A. Mantoglou, "Physics-based and data-driven surrogate models for pumping optimization of coastal aquifers," *Eur. Water*, vol. 57, pp. 481–488, Nov. 2017.
- [35] Z. G. Chen *et al.*, "Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912–2926, Aug. 2019.
- [36] S.-Z. Zhou, Z.-H. Zhan, Z.-G. Chen, S. Kwong, and J. Zhang, "A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 8, 2020, doi: [10.1109/TITS.2020.2994779](https://doi.org/10.1109/TITS.2020.2994779).

- [37] D. Liang, Z.-H. Zhan, Y. Zhang, and J. Zhang, "An efficient ant colony system approach for new energy vehicle dispatch problem," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 17, 2019, doi: [10.1109/TITS.2019.2946711](https://doi.org/10.1109/TITS.2019.2946711).
- [38] R. Allmendinger, M. T. M. Emmerich, J. Hakanen, Y. Jin, and E. Rigoni, "Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case," *J. Multi Crit. Decis. Anal.*, vol. 24, nos. 1–2, pp. 5–24, 2017.
- [39] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.
- [40] 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.
- [41] J.-Y. Li, Z.-H. Zhan, C. Wang, H. Jin, and J. Zhang, "Boosting data-driven evolutionary algorithm with localized data generation," *IEEE Trans. Evol. Comput.*, early access, Mar. 17, 2020, doi: [10.1109/TEVC.2020.2979740](https://doi.org/10.1109/TEVC.2020.2979740).
- [42] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, May. 2015.
- [43] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng, "Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2437–2448, Nov. 2015.
- [44] H. Wang, Q. Zhang, L. Jiao, and X. Yao, "Regularity model for noisy multiobjective optimization," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 1997–2009, Sep. 2016.
- [45] P. Lim, C. K. Goh, and K. C. Tan, "Evolutionary cluster-based synthetic oversampling ensemble (ECO-ensemble) for imbalance learning," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2850–2861, Sep. 2017.
- [46] A. Karpatne *et al.*, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2318–2331, Oct. 2017.
- [47] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized multi-tasking for evolutionary optimization of expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 44–58, Feb. 2019.
- [48] C. Sun, J. Zeng, J. Pan, S. Xue, and Y. Jin, "A new fitness estimation strategy for particle swarm optimization," *Inf. Sci.*, vol. 221, pp. 355–370, Feb. 2013.
- [49] 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.
- [50] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, Mar. 2019.
- [51] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, Aug. 2006.
- [52] N. Namura, K. Shimoyama, and S. Obayashi, "Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 898–913, Dec. 2017.
- [53] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.
- [54] K. Deb and H. G. Beyer, "Self-adaptive genetic-algorithms with simulated binary crossover," *Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.
- [55] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
- [56] Y.-S. Ong, P. B. Nair, and K. Y. Lum, "Max-min surrogate-assisted evolutionary algorithm for robust design," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 392–404, Aug. 2006.
- [57] R. M. Hicks and P. A. Henne, "Wing design by numerical optimization," *J. Aircraft*, vol. 15, no. 7, pp. 407–412, 1978.
- [58] X. Fei, J. Branke, and N. Gülpınar, "New sampling strategies when searching for robust solutions," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 273–287, Apr. 2019.
- [59] I. H. Abbott and A. E. Von Doenhoff, *Theory of Wing Sections: Including a Summary of Airfoil Data*. New York, NY, USA: Courier, 2012.
- [60] (2013). *Xfoil*. [Online]. Available: <http://web.mit.edu/drela/Public/web/xfoil/>



**Jian-Yu Li** (Student Member, IEEE) received the B.S. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2018, where he is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering.

His research interests mainly include computational intelligence, data-driven optimization, machine learning, including deep learning and their applications in real-world problems, and in environments of distributed computing and big data.



**Zhi-Hui Zhan** (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently the Changjiang Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His current research interests include evolutionary computation algorithms, swarm intelligence algorithms, and their applications in real-world problems, and in environments of cloud

computing and big data.

Dr. Zhan was a recipient of the Outstanding Youth Science Foundation from National Natural Science Foundations of China in 2018 and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. His doctoral dissertation was awarded the IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation and the China Computer Federation Outstanding Ph.D. Dissertation. He is listed as one of the Most Cited Chinese Researchers in Computer Science. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the *Neurocomputing*, and the *International Journal of Swarm Intelligence Research*.



**Hua Wang** (Member, IEEE) received the Ph.D. degree from the University of Southern Queensland, Toowoomba, QLD, Australia, in 2004.

He was a Professor with the University of Southern Queensland. He is currently a full-time Professor with Victoria University, Melbourne, VIC, Australia. He has more than ten years teaching and working experience in applied informatics at both enterprise and university. As a Chief Investigator, three Australian Research Council Discovery grants have been awarded since 2006, and 200 peer-

reviewed scholar papers have been published. Six Ph.D. students have already graduated under his principal supervision. He has expertise in electronic commerce, business process modeling, and enterprise architecture.



**Jun Zhang** (Fellow, IEEE) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Visiting Scholar with Victoria University, Melbourne, VIC, Australia. His current research interests include computational intelligence, cloud computing, high-performance computing, data mining, wireless sensor networks, operations research, and power electronic circuits. He has published over 100 technical papers in the above areas.

Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011, and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.