Boosting Data-Driven Evolutionary Algorithm With Localized Data Generation

Jian-Yu Li, *Student Member, IEEE*, Zhi-Hui Zhan, *Senior Member, IEEE*, Chuan Wang, Hu Jin, *Senior Member, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—By efficiently building and exploiting surrogates, data-driven evolutionary algorithms (DDEAs) can be very helpful in solving expensive and computationally intensive problems. However, they still often suffer from two difficulties. First, many existing methods for building a single ad hoc surrogate are suitable for some special problems but may not work well on some other problems. Second, the optimization accuracy of DDEAs deteriorates if available data are not enough for building accurate surrogates, which is common in expensive optimization problems. To this end, this article proposes a novel DDEA with two efficient components. First, a boosting strategy (BS) is proposed for self-aware model managements, which can iteratively build and combine surrogates to obtain suitable surrogate models for different problems. Second, a localized data generation (LDG) method is proposed to generate synthetic data to alleviate data shortage and increase data quantity, which is achieved by approximating fitness through data positions. By integrating the BS and the LDG, the BDDEA-LDG algorithm is able to improve model accuracy and data quantity at the same time automatically according to the problems at hand. Besides, a tradeoff is empirically considered to strike a better balance between the effectiveness of surrogates and the time cost for building them. The experimental results show that the proposed BDDEA-LDG algorithm can generally outperform both traditional methods without surrogates and other state-of-the-art DDEA son widely used benchmarks and an arterial traffic signal timing real-world optimization problem. Furthermore, the proposed BDDEA-LDG algorithm can use only about 2% computational budgets of traditional methods for producing competitive results.**

***Index Terms*—Boosting strategy (BS), data-driven evolutionary algorithm (DDEA), expensive optimization problems (EOPs), localized data generation (LDG), surrogate.**

Manuscript received June 18, 2019; revised September 17, 2019, November 19, 2019, and January 14, 2020; accepted March 2, 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 Guangdong Natural Science Foundation Research Team under Grant 2018B030312003, and in part by the Ministry of Science and ICT through the National Research Foundation of Korea under Grant NRF-2019H1D3A2A01101977. *(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 51006, China (e-mail: zhanapollo@163.com).

Chuan Wang is with the College of Software, Henan Normal University, Xinxiang 453007, China.

Hu Jin and Jun Zhang are with the Department of Electrical Engineering, Hanyang University, Ansan 15588, South Korea (e-mail: junzhang@ieee.org).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

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/TEVC.2020.2979740

I. Introduction

As a branch of evolutionary algorithms (EAs), data-driven EAs (DDEAs) are effective and efficient in solving real-world expensive optimization problems (EOPs) [1], [2]. As traditional EAs rely heavily on fitness evaluations (FEs) to produce and select new populations, their performance often deteriorates when the number of available FEs is not enough [2]. This is usually the case in real-world applications, where the FEs may be too expensive or computational intensive to access [3], [4]. Different from traditional EAs, by using data (e.g., evaluated solutions) and surrogates to replace the FEs and to drive the evolution, DDEAs are able to obtain satisfactory solutions within a limited amount of available FEs [5]–[8]. Furthermore, in some very difficult application problems, such as blast furnace optimizations [8], [9], trauma system optimizations [10], and fused magnesium furnaces optimizations [11], no real FEs can be conducted any more during the evolutionary process due to their practical conditions, such as deadline constraints or insufficient budgets, making the traditional methods almost impossible for solving these EOPs. In such application scenarios, offline DDEAs are more useful and efficient, because they can build surrogates only on the basis of historical evaluated data to replace real FEs and drive the optimizations [8]. Based on the above, DDEAs are more efficient and useful than traditional EA methods in solving expensive and computationally intensive application problems. However, how to efficiently utilize the available data and surrogates is still the main challenge in DDEAs.

Generally speaking, to enhance DDEAs, one should consider both the surrogate model and the data, because they are both essential to the performance of DDEAs [8], [10]. That is, researchers have tried to improve DDEAs by obtaining better surrogate models and better data. For example, selecting suitable and appropriate models and methods for building surrogates can improve DDEAs, such as using polynomial fitting methods [14], mathematical estimations [16], and machine learning techniques [17]–[19]. Also, DDEAs can be enhanced by managing and combining a set of surrogates [15], [20]–[22]. Moreover, as the data quality can also affect the surrogate performance, data processing methods like local smooth [23] and data mining techniques [10] can be helpful in further improving DDEAs. Furthermore, in the cases that evaluated data are not enough for building an accurate surrogate, data generation can be an effective approach to increase data quantity [11], [24]–[25].

In this article, we focus on both the model management and the data quantity to propose a boosting DDEA (BDDEA) with localized data generation (LDG) method, name BDDEA-LDG algorithm. The proposal of the BDDEA-LDG algorithm is based on the following two motivations.

First, although many valuable surrogate guidelines and experience have been provided for enhancing DDEAs, they are empirically designed for some special problems but may not work well on other problems [26]–[28]. For example, a surrogate with simplified models for the automated design problem of dispatching rules [26] may not work well for the weld sequence optimization problems [27]. Therefore, the users may still need to test existing surrogate models one by one in order to find a suitable model for solving new problems. This motivates us to study whether the surrogates can obtain promising performance by self-improvement or self-adaptation. That is, a surrogate model may be able to boost its performance by accommodating itself to the optimization problems at hand. Following this and inspired by the idea of boosting in ensemble learning [29]–[31], we propose a boosting strategy (BS) for efficient and self-aware model management. The BS sequentially builds a set of different surrogates and incorporates them into a combination model to approximate the real FEs. During this process, each new surrogate is built with an emphasis on the approximation error made by earlier surrogates. In this way, the combination model can be iteratively improved by repeatedly incorporating newly built surrogates, because each newly built surrogate can help to correct the prediction mistakes made by existing (i.e., earlier) surrogates.

Second, the optimization accuracy of DDEAs will greatly deteriorate if there are not enough data for building accurate surrogates. Therefore, the LDG method is further proposed in this article to increase data quantity and alleviate data shortages. There are two advantages of the LDG: 1)it approximates the fitness of synthetic data through their positions, which is computational efficiency and easy to implement and 2) it can be employed to assist the BS to generate data in areas where existing surrogates have large prediction errors, so that the new surrogates built on the synthetic data will emphasize more on the prediction accuracy of corresponding areas. This can help achieve the goals of the BS, i.e., efficient and self-aware model management.

As a result, by combining the BS and the LDG, the proposed BDDEA-LDG can accommodate itself to different problems and, at the same time, its optimization accuracy can be less influenced by the shortage of data quantity. These advantages can make BDDEA-LDG suitable for solving various EOPs in different situations. Besides, a tradeoff is experimentally considered to strike a balance between the effectiveness of surrogate models and the time cost for building them. To validate the performance of BDDEA-LDG, the experiments and comparisons are conducted on widely used benchmarks with 10 to 100 dimensions and also on a real-world application problem of arterial traffic signal optimization. The comparison results show that the proposed BDDEA-LDG algorithm can generally outperform the state-of-the-art DDEAs when given the same FEs, especially, on the problems where the available data are not enough for building accurate surrogates. Furthermore, the experiments show that the proposed BDDEA-LDG algorithm only requires about 10% FEs for producing better results and 2% FEs for producing competitive results, when compared with traditional optimization methods without surrogates.

The remainder of this article is organized as follows. Section II briefly introduces the DDEAs and related work, while Section III details the proposed BDDEA-LDG algorithm. The experiments, including settings, comparisons, and analyses, are provided in Section VI. Finally, Section V draws the conclusion.

II. Background and Related Work

A. Data-Driven Evolutionary Algorithms

Generally speaking, the key issue of DDEAs is to utilize data to reduce the needed FEs and drive the evolution [8]. Such data utilizations are often achieved through surrogates [14]. That is, by building suitable surrogates based on evaluated data, the DDEAs are able to employ these surrogates to replace the real FEs and then reduce the needs for accessing real FEs. Therefore, DDEAs can have more advantages than traditional EAs when solving expensive and computationally intensive problems [10], [13].

As for the algorithm framework, a DDEA often has the surrogate model management (SMM) part and the evolutionary optimization procedure (EOP) part [15], [16]. The SMM will manage surrogate models for better approximations while the EOP will employ surrogates into the EAs to perform evolution [15]. Also, the SMM can adjust and update surrogates according to the feedback and data from the EOP [17]. Based on whether the EOP can obtain new data through real FEs, DDEAs can be implemented in two versions: 1) online DDEAs and 2) offline DDEAs [8]. In online DDEAs, the EOP can evaluate several data through real FEs. These newly evaluated data can be used by the SMM to further provide landscape information and to help construct more accurate surrogate models [18]. Therefore, online DDEAs are suitable for the situation that a few FEs are still available from physical experiments or expensive calculations during the evolution process [19]. By contrast, offline DDEAs are designed for the situation that the real FEs are too expensive to perform or too difficult to access [15]. In these cases, EOP cannot obtain any new data through real FEs. Instead, it can only use historical data to drive the evolution, which is different from the online DDEAs. As mentioned above, although there are differences between online and offline DDEAs, both of their main ideas are to reduce the needed FEs and drive the evolution through utilizing evaluated data.

B. Related Work

So far, many methods have been proposed to further enhance DDEAs [8]. This section briefly reviews related work and discusses the differences between them and the BDDEA-LDG. Generally speaking, as described in Section II-A, DDEAs can be classified into two categories: 1) offline DDEAs and 2) online DDEAs. As a number of DDEAs are proposed for solving multiobjective [32], [33] or many-objective problems [34], the following contents will also clarify their multi-/many-objective characteristics when surveying them among the offline and online DDEAs.

In offline DDEAs, algorithms need to build surrogate models only based on the given data to explore the search space, because no new data can be evaluated during the optimization process. For example, Wang *et al.* [15] proposed a DDEA using selective ensemble surrogates (DDEA-SE), which is a state-of-the-art algorithm with excellent efficiency for offline data-driven optimizations. The DDEA-SE builds a large number of surrogates based on data resampled from offline data and then adaptively selects some of the prebuilt surrogates for approximating FEs in different evolutionary stages, so that the prediction error can be reduced. Moreover, as no new data can be obtained during the optimization process, the quality of the given data can heavily affect the accuracy of DDEAs [8]. Therefore, many preprocessing methods have also been proposed for data with poor quality, such as imbalanced [35], [36], incomplete [37], and noisy data [38]. For instance, in a many-objective blast furnace problem, Chugh *et al.* [9] adopted a local regression method to reduce the noise in the offline data set and then built Kriging models to improve the reference vector guided EA. For big data and redundant data, data redundancy and long computation time can be reduced through data mining and related methods. For example, in a trauma system design problem, Wang *et al.* [10] proposed a novel multiobjective algorithm employing a clustering method to recognize the useful data patterns for building surrogates, where about 90% of running time was finally saved. Furthermore, for the situations that the size of the given data is not enough to build accurate surrogates, generating additional data can be a potential way to solve this problem [8]. For example, in a multiobjective fused magnesium furnace optimization problem, Guo *et al.* [11] used a low-order polynomial model to generate synthetic data and predict their fitness. Although the above algorithms and BDDEA-LDG are offline DDEAs, BDDEA-LDG integrates the BS and LDG to improve the surrogate models, and therefore is different from the above algorithms.

In online DDEAs, additional data can be evaluated during the optimization process. As a result, this provides more space for algorithm improvements when compared with offline DDEAs. As offline DDEAs can be considered as a special case of online DDEAs, the aforementioned methods proposed for offline DDEAs can also be employed in online DDEAs. Besides, as new data can be evaluated by real FEs to test the current surrogates, online DDEAs can adaptively select proper models and perform model managements. In model selection, different appropriate models and methods can be selected to build surrogates, which can include traditional interpolation methods [16] and machine learning techniques [17], such as polynomial regression model [20], Kriging model [23], [41], artificial neural networks [42]–[44], and radial basis function neural networks (RBFNNs) [45]–[47]. Furthermore, new approximation methods have also been studied. For example, Sun *et al.* [16] proposed a new fitness approximation strategy for particle swarm optimization (PSO), which estimated fitness based on the positional relationship between individuals. For model management, there are two major branches. One branch is to combine or integrate different surrogates, because different surrogate models have different advantages. For example, Wang *et al.* [21] integrated global and local surrogates to balance global exploration and local exploitation. Also, Sun *et al.* [47] proposed a surrogate-assisted cooperative swarm optimization (SA-COSO), which employs a surrogate-assisted PSO for local search and a surrogate-assisted social learning PSO for explorations. Another branch of model management strategy considers how to update surrogate models. In online DDEAs, better surrogate models can be obtained by selecting more crucial individuals to be evaluated. Generally speaking, these strategies will consider the way and the criteria for individual selections. According to the way of selecting individuals, there can be generation-based and individual-based strategies [2]. Generation-based strategies perform FEs according to the generation, where the frequency for performing real FEs can be adaptive [42] or predefined [56]. In contrast, individual-based strategies evaluate some individuals in a population at each generation [50]. As for the selection criteria, there are often two considerations: 1) the promising individuals and 2) the uncertain individuals [2]. The promising individuals have better-predicted fitness and may help figure out the exact optimum positions [42], [57], while evaluating the uncertain individuals can increase surrogate reliability [2], [50]. However, it is difficult to measure the prediction uncertainty. Therefore, some methods, like Kriging models [23], [51], are favored by many strategies because they are able to provide measurements of prediction uncertainty. However, Kriging models may not work well on high dimensional problems due to their expensive time cost. Therefore, some researches have tried to transform decision variables from lots of dimensions to fewer dimensions, such as the Gaussian process surrogate model assisted EA for medium-scale problem (GPEME) [19]. Except for the Kriging models, some researches employ the variance of surrogate outputs to measure the uncertainty [18], [58]. In addition, as evaluating promising and uncertain individuals have different advantages, many strategies called infill criteria are proposed and studied based on the combinations of them, such as expected lower confidence bound [19], probability of improvement [59], and expected improvement [52], [60]. Moreover, Tian *et al.* [13] proposed a multiobjective infill criterion driven GP-assisted social learning PSO (MGP-SLPSO), where the multiobjective infill criteria are shown to be efficient when optimizing fitness and minimizing uncertainty together in solving high dimensional problems.

III. Proposed Algorithm

A. Localized Data Generation

The main idea of LDG is to generate data within the neighborhood of evaluated data, so as to increase the data quantity and indirectly improve the quality of surrogates. To avoid confusion in the following contents, the data evaluated by FEs and the data generated by LDG are denoted as "original data" and "synthetic data," respectively.

The original data can be presented as input–output pairs to form a training data set $TD = \{(x_i, F(x_i))|i = 1, 2, \ldots, N\}$, where N is the number of original data x (i.e., the data evaluated by real FEs). The task of LDG is to generate new synthetic training data based on the data in TD. Also, we denote S as a subset of TD that contains the selected data for generating new data, and the generated synthetic data set K generated by LDG is represented as

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

$$l = \sqrt{\frac{\sum_{j=1}^{D} \left(U_j - L_j\right)^2}{D}} \cdot 10^{-6} \quad (2)$$

where $F(\cdot)$ is the true fitness function, l controls the neighborhood size of the original data, Δx is a random vector, and D is the dimension, while U_j and L_j represent the upper bound and lower bound of jth dimension, respectively. To avoid ambiguity, we further define an augmented training data (ATD) set as the union of TD and K

$$ATD = TD \bigcup K. \quad (3)$$

Note that if l in (1) is small enough, the fitness of x_{new} and x_s can be very similar when the fitness function is a continuous function. Based on this, we denote that the true fitness value of x_{new} is the same as x_s, namely, $F(x_{\text{new}}) = F(x_s)$. In this way, we can obtain the fitness value of the additional data x_{new} without consuming any FEs. Although the data generation may bring noises (especially, when the landscape of the objective function is very sharp where two close individuals may have significantly different fitness values), we can properly configure the parameter l so that the LDG is performed in a safe region to avoid producing noises. The value of l is set according to (2), which will differ from problems to problems according to their boundaries (i.e., U_j and L_j) and is scaled by a small value 10^{-6} to further narrow down the size of the safe region. To investigate its effectiveness and sensitiveness, related experiments and analyses on benchmark functions with different characteristics, such as multimodal and nonseparable, are provided in Section IV-H.

As minimization problems can be converted to maximization problems, Algorithm 1 simply presents the pseudo code of LDG for minimum optimizations. The inputs of LDG are the original data set TD, a surrogate model set (SMS) containing NS surrogates, and the value of NS, while its output is the synthetic data set K. In the implementation of this article, all the NS surrogates are RBFNNs, which are efficient and easy-to-implement [16], [17]. In this way, the LDG is able to simply store the network parameters (i.e., weights and number of neurons) and rebuild the same surrogates when needed. The LDG mainly has four steps. The first is to re-evaluate all the data by employing NS surrogates to obtain the average prediction of fitness, denoted as Y_{pre}. The second is to compute the difference, $\text{diff} = Y_{\text{pre}} - F(x)$, for the data in TD. The third is to sort all the original data according to their *diff* in descending order and set the first 50% of them as S. The fourth is to generate K with Saccording to (1). In the third step, the selection criterion of the large *diff* is based on the following consideration. First, for a historical data x that has been evaluated, its real fitness $F(x)$ is known. Then, a large *diff* means that the error between the prediction Y_{pre} and the real fitness $F(x)$ is large. Therefore, LDG should be performed on this data. Actually, the value of l for performing LDG is suggested to be small enough so that the synthetic data generated by LDG can have similar fitness with data x. Also, it should be noted that the $\text{diff} = Y_{\text{pre}} - F(x)$ is designed for minimization problems here, and if for maximization problems, $\text{diff} = F(x) - Y_{\text{pre}}$ is suggested. To validate this selection criterion, related experiments and analyses are performed and provided in Section IV-H of this article. In addition, the reason for using 50% data is that more data may obtain more accurate surrogates while too much data can make learning step time-consuming, and, therefore, a half makes the balance. Further, the experiments on using different sizes of data are provided in Section IV-H.

Algorithm 1 LDG

Input: *TD*-the original training data set,
SMS-the surrogate model set,
T-the the number of surrogates in the *SMS*.
Output: *K*-the synthetic data set.
Begin
//Compute *diff* of each data (for guiding data selections)
For each x_i in *TD* **Do**
Use the T surrogates in *SMS* to predict the fitness of x_i;
Calculate the average of the above T predicted fitness as $Y_{pre,i}$;
Calculate the difference $diff_i = Y_{pre,i} - F(x_i)$;
End For
Sort the data in *TD* according to their *diff* with descending order;
Set *S* as the first 50% samples of the sorted *TD*;
Set *K* as empty set;
For each x_i in *S* **Do**
Generate x_{new} through x_i and Eq. (1);
Set $F(x_{new}) = F(x_i)$;
$K = K \cup (x_{new}, F(x_{new}))$;
End For
End

B. Model Management With Boosting Strategy

BS sequentially builds surrogates and iteratively updates the SMS, as shown in Fig. 1. In Fig. 1, the LDG and related model training are sequentially performed until enough surrogates are obtained. Every time a surrogate is built, it will be stored in the SMS and, therefore, the SMS will be iteratively changed. The pseudo code of the whole process is provided in Algorithm 2. To better describe how Algorithm 2 works, an example is given here. First, the initial surrogate M_1 will be built on the ATD_1, where the ATD_1 is initialized as the same as the original data set TD. Second, M_1 is used to select data for the next LDG. The result produced by LDG, namely, the synthetic data set K_1 (refer to Algorithm 1), will be added into the ATD_1, resulting in a larger data set, ATD_2. Third, the second surrogate M_2 will be trained on the basis of the ATD_2. Fourth, the M_1 and M_2 are employed together to select data in

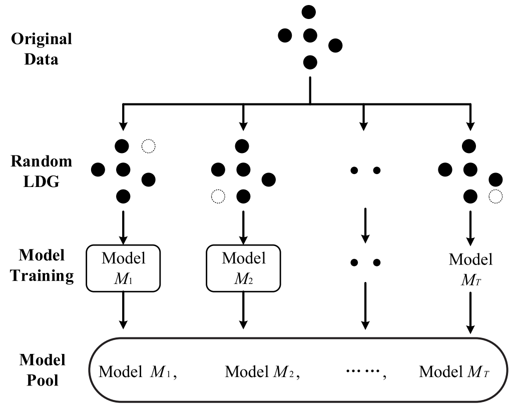


Fig. 1. Diagram of BS.

Algorithm 2 BS

Input: *TD*-the original training data set,
T-the number of surrogate models to be obtained.
Output: *SMS*-the surrogate model set containing *T* surrogates.
Begin
1: Initialize augmented data set ATD_1 as TD;
2: Build the first surrogate model, M_1, based on ATD_1;
3: Set $SMS = \{M_1\}$;
4: **For** $j = 2$ to T **Do**
5: $K_j =$ **LDG**(TD, SMS, $j-1$); //refer to **Algorithm 1**.
6: $ATD_j = ATD_{j-1} \cup K_j$;
7: Build new surrogate model, M_j, based on ATD_j;
8: $SMS = SMS \cup \{M_j\}$;
9: **End For**
End

LDG for obtaining K_2 and the third data set ATD_3. Then the third surrogate M_3 is built based on the ATD_3. The above process will be performed repeatedly until T different surrogates are obtained, where T is the total number of surrogates defined by users.

To better illustrate the relationship between newly built surrogates and existing surrogates, we provide a mathematical analysis. Given data x and T existing surrogates $M_1, \ldots, M_T$, we denote their prediction as $M_1(x), \ldots, M_T(x)$, respectively. Then, the combination model obtained in BS, denoted as $M_{\mathrm{BS},T}$, satisfies

$$M_{\mathrm{BS},T}(x) = \frac{1}{T}\sum_{i=1}^{T} M_i(x). \quad (4)$$

Its generation error can be defined on the distribution $p(x)$ and real fitness $F(x)$ of data x as

$$E\big(M_{\mathrm{BS},T}\big) = \int L\big(F(x), M_{\mathrm{BS},T}(x)\big)p(x)dx \quad (5)$$

where L is the loss function. Although there are many different loss functions, we use the quadratic loss function here for simplicity, namely, $L(a, b) = (a-b)^2$ where a and b are real numbers. Like other researches in machine learning, our main task is to obtain a model M that has a small $E(M)$ [29], [62].

Now, we consider a newly built surrogate M_{T+1}. With (4) and (5), the empirical risk of $M_{\mathrm{BS},T+1}$ satisfies (6). This equation shows that the aim of the newly built surrogate M_{T+1} is not only to approximate the real fitness but also to eliminate the prediction error made by existing surrogates $M_1, \ldots, M_T$. More specifically, $F(x)$ in (6) is the real fitness on x and $F(x) - M_i(x)$ is actually the prediction error made by model M_i. According to the last line in (6), to obtain a model $M_{\mathrm{BS},T+1}$ with smaller E is to find an M_{T+1} that its prediction $M_{T+1}(x)$ on data x is more similar to the sum of two value, the prediction error made by existing surrogates and the real fitness of x. In other words, the total generation error can be reduced if newly built surrogates can approximate the sum of real fitness from FEs and prediction error from existing surrogates. Therefore, it is suggested that the new surrogates should be built with considerations on the prediction error made by existing surrogates. This is consistent with the ideas of the BS. In addition, the above analysis can be further extended to other cases because that (6) holds as long as the $L(a, b)$ is a function of the difference between value a and b, like absolute loss function $L(a, b) = |a-b|$

$$\begin{aligned}
E\big(M_{\mathrm{BS},T+1}\big) &= \int L\big(F(x), M_{\mathrm{BS},T+1}(x)\big)p(x)dx \\
&= \int \left(F(x) - \frac{1}{T+1}\sum_{j=1}^{T+1} M_i(x)\right)^2 p(x)dx \\
&= \int \left(\frac{\sum_{i=1}^{T}(F(x) - M_i(x)) + F(x_i) - M_{T+1}(x)}{T+1}\right)^2 \\
&\quad \times p(x)dx \\
&= \int L\left(\frac{\sum_{i=1}^{T}(F(x) - M_i(x)) + F(x_i)}{T+1}, \frac{M_{T+1}(x)}{T+1}\right) \\
&\quad \times p(x)dx. \qquad (6)
\end{aligned}$$

C. Whole Proposed Algorithm

The diagram of the complete BDDEA-LDG is shown in Fig. 2. Without loss of generality, Fig. 2 presents the version of offline BDDEA-LDG and denotes all evaluated data as offline data, because methods for offline DDEAs can also be employed in online DDEAs [15].

Like other DDEAs, BDDEA-LDG can be mainly described in two parts, the EOP, and the SMM part, as shown in

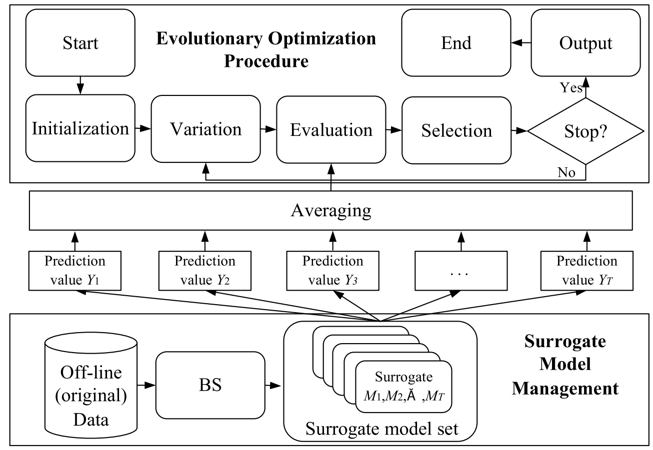


Fig. 2. Diagram of complete BDDEA-LDG.

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

Fig. 2. Its EOP is similar to traditional EAs, which includes initialization, variation (i.e., crossover and mutation), FE, and selection. Consequently, different kinds of EAs can be adopted as the optimizer in the BDDEA-LDG, such as particle swarm optimization [63], differential evolution [64], ant colony system [65], and genetic algorithm (GA) [66].

The SMM of the BDDEA-LDG focuses on building surrogate models. Based on the original data, the BS sequentially builds a set of surrogates with the help of LDG, where the surrogates will be stored in an SMS. When performing the FE of an individual, the algorithm will use all the surrogates in the model set to predict the fitness of this individual. The average of these predicted values will then be calculated as the final prediction result, which will be employed in the selection procedure in the EOP. In this way, the EOP can employ these prediction results to drive the evolution. When the stop criteria are met, the EOP will output the best individual based on the predictions as the final solution, and then the algorithm finishes.

IV. Experimental Studies

A. Experimental Setup

In the experiments, five commonly used benchmark problems [15] are adopted to test the proposed algorithm, as presented in Table I. To show the effectiveness of BDDEA-LDG, not only random sample and traditional EA methods but also some state-of-the-art DDEA algorithms are employed for comparisons. The employed DDEAs are: DDEA-SE [15], CAL-SAPSO [18], GPEME [19], MGP-SLPSO [13], and SA-COSO [47]. Besides their promising performance, there are other reasons for choosing these algorithms. First, CAL-SAPSO can help observe the features of the combination model in BDDEA-LDG because CAL-SAPSO also employs ensemble surrogates to make committee-based decisions. Second, GPEME is a representative algorithm that uses Kriging models for online data-driven optimizations, which can reflect the potential of BDDEA-LDG for being extended to online optimization. Third, because CAL-SAPSO and GPEME are proposed for small and medium scale problems, MGP-SLPSO and SA-COSO can be used for the comparisons on high-dimensional problems [13], [47]. Fourth, DDEA-SE is a powerful offline DDEA, which is ideal for comparing of offline data-driven optimizations.

In the experiments, all the compared algorithms are configured according to their original papers. As for BDDEA-LDG, the underlying optimization algorithm is the same as that used in DDEA-SE [15], which is a variant of GA using a simulated binary crossover (SBX), polynomial mutation, and tournament selection [66]. Also, its parameters are configured the same as those in DDEA-SE for fair comparisons [15]. That is, the population size is 100, the crossover and mutation probabilities are 1.0 and 1/D, respectively, where D is the problem dimension.

As for the surrogates, all the base models used in BDDEA-LDG are RBFNNs. There are two main reasons for using RBFNNs. First, RBFNN is a fast, computationally efficient, and easy-to-implement method for approximation tasks [15], [18], [67]. Second, RBFNNs have been widely used as surrogates in [12], [15], and [17], which are the compared algorithms in this article, and, therefore, using RBFNNs in BDDEA-LDG can help achieve fair comparisons. The settings of all RBFNNs in BDDEA-LDG are configured the same as those in DDEA-SE [15], so that their comparisons can be fair. Specifically, in the BDDEA-LDG, the employed activation function of each RBFNNis the Gaussian radial basis function and the number of neurons in its hidden layer equals to the problem dimension, D. In the state-of-the-art offline algorithm, DDEA-SE [15], the settings of its RBFNNs are set according to its original paper, which are the same with those in BDDEA-LDG. As for the state-of-the-art online DDEAs, the settings of surrogates are also set according to their original papers. In CAL-SAPSO, the RBFNNsare based on MATLAB toolbox [18], which uses two neurons in the hidden layer and employs the Gaussian radial basis function as the activation function. In SA-COSO [47], the RBFNNs keep learning from the data until the number of the hidden neurons reach 8, where all the activation function of hidden neurons are Gaussian radial basis functions.

To conduct fair numerical experiments, we also make the following experiment settings.

First, the maximum number of available FEs for all the algorithms is 11-D. Especially, for offline data-driven algorithms, 11-D data are sampled by Latin hypercube sampling (LHS) [68] before the optimizations and no more FEs will be allowed during these arch procedures. As for the online algorithms, their parameters are configured the same as those in their original papers. According to the literature, CAL-SAPSO and GPEME begin with 5-D exact FEs and terminates when 11-D FEs are exhausted, where the 5-D FEs are needed to obtain offline data to initialize their databases before the optimizations [15], [18], [19]. Differently, SA-COSO and MGP-SLPSO start with 0 FEs and terminates if 11-D FEs are exhausted, because they do not need to construct databases using offline data in advance [13], [47]. Instead, these two algorithms obtain their initial databases by evaluating their evolving populations during the optimization processes.

Second, to reduce statistical errors, all algorithms are tested 25 times independently on each problem and the average results are used. In addition, Wilcoxon's rank-sum test with a significant level $\alpha = 0.05$ is adopted as the hypothesis testing to compare algorithms. Based on the Wilcoxon's rank-sum test, the symbols "+," "≈," and "−" are, respectively, employed to show that the proposed algorithm performs significantly better than, similar to, and significantly worse than the algorithm compared. As the Wilcoxon's rank-sum test can be only used for pairwise comparisons, the Friedman test with the Bergmann–Hommel *post-hoc* test (significance level

TABLE II
AVERAGE OF TIME COST (UNIT: SECOND) OVER 25 INDEPENDENT RUNS OF DIFFERENT OFFLINE ALGORITHMS ON ELLIPSOID AND RASTRIGIN PROBLEMS

Problem	D	BDDEA-LDG (T=100, gen=100)	BDDEA-LDG (T=50, gen=500)	DDEA-SE (T=2000)
Ellipsoid	10	2.56E+01	**2.08E+01**	2.13E+01
	30	1.02E+02	8.75E+01	**7.68E+01**
	50	2.27E+02	**1.61E+02**	2.62E+02
	100	1.20E+03	**6.19E+02**	1.38E+03
Rastrigin	10	2.51E+01	2.05E+01	**1.77E+01**
	30	9.63E+01	7.59E+01	**7.24E+01**
	50	2.24E+02	**1.61E+02**	3.43E+02
	100	1.22E+03	**6.32E+02**	1.36E+03
Average Ranking		2.38	1.69	1.94
Adjusted p-value		0.5074	0.6171	NA

TABLE III
COMPARISONS ON OPTIMIZATION RESULTS BETWEEN VARIANTS OF THE PROPOSED ALGORITHM WITH DIFFERENT SETTINGS

Problem	D	Metric	T=50 and gen=500	T=50 and gen=100	T=100 and gen=500
Ellipsoid	10	Mean	**1.01E+00**	1.17E+00(+)	1.12E+00(+)
		Std.	3.99E-01	4.38E-01	3.80E-01
	30	Mean	6.66E+00	7.11E+00(+)	**3.40E+00(-)**
		Std.	2.09E+00	2.18E+00	7.46E+00
	50	Mean	1.31E+01	1.71E+01(+)	**1.28E+01(≈)**
		Std.	3.19E+00	3.48E+00	3.67E+00
	100	Mean	5.55E+01	2.82E+02(+)	**4.79E+01(-)**
		Std.	1.12E+01	6.31E+01	8.14E+01
Rastrigin	10	Mean	**6.51E+01**	8.42E+01(+)	6.79E+01(+)
		Std.	2.96E+01	2.67E+01	2.23E+01
	30	Mean	**1.46E+02**	1.59E+02(+)	1.52E+02(≈)
		Std.	4.34E+01	2.76E+01	2.76E+01
	50	Mean	1.90E+02	2.18E+02(+)	**1.89E+02(≈)**
		Std.	3.18E+01	4.04E+01	4.24E+01
	100	Mean	4.05E+02	7.45E+02(+)	**3.56E+02(-)**
		Std.	1.44E+02	6.72E+01	7.71E+01
+/≈/-			NA	8/0/0	2/3/3
Average Ranking			1.63	3	1.38
Adjusted p-value			NA	**0.0035**	0.6171

= 0.05) is further employed to carry out multiple comparisons of different algorithms.

B. Tradeoff Between Optimization Procedure and Model Management

Before the comparisons with other algorithms, we consider the tradeoff between the EOP and SMM for the better performance of the BDDEA-LDG, because properly allocating the computational budgets and resources between EOP and SMM can be crucial to the algorithm performance [2]. For example, an accurate surrogate model may be of little use if the optimizer is configured with short runtime and fails to converge before terminations, while a poorly trained surrogate cannot help locate the true optimum no matter how long the optimizer searches for.

To begin with, the time cost of BDDEA-LDGwith 100 surrogates and 100 generations are tested on the benchmark problems. For convenience, we denote the surrogate number as T and the generation number as *gen*. The results of the time cost (in seconds) on representative unimodal and multimodal problems, namely, Ellipsoid and Rastrigin, are shown in Table II. In addition, the DDEA-SE is regarded as the comparison baseline and the control method in Friedman test with the Bergmann–Hommel *post-hoc* test (significance level = 0.05), because it is also an offline DDEA with efficient managements of ensemble surrogates [15].

As shown in Table II, the BDDEA-LDG ($T = 100$, $gen = 100$) seems to allocate too many budgets on its SMM so that it needs longer running time than DDEA-SE on 10- and 30-D problems, even though it consumes shorter time on 50- and 100-D problems. This may be due to the fact that, in BDDEA-LDG, the data set for building surrogates is iteratively enlarged. Consequently, the corresponding training time increases dramatically as the data set enlarges. In fact, according to the LDG (Algorithm 1) and the BS (Algorithm 2), BDDEA-LDG will add $0.5|TD|$ new data into the data set before building a new surrogate, where $|TD|$ means the number of data in TD. That is, the first surrogate is built on $|TD|$ data, the second surrogate on $1.5|\ TD|$ data, and the ith surrogate on $(1 + 0.5i) \cdot |TD|$ data, which are increasingly time consuming. Therefore, it is better to decrease T for better efficiency.

As decreasing T may affect optimization accuracy while increasing *gen* may improve accuracy [1], we cut the T from 100 to 50 and at the same time increase the *gen* from 100 to 500, so as to tradeoff the budget for SMM against the EOP without losing too much optimization accuracy. This time, the execution time of the BDDEA-LDG is more satisfactory and it only consumes a longer time on three problems. Furthermore, according to the Friedman test, BDDEA-LDG ($T = 50, gen = 500$) obtains better ranking than DDEA-SE and its p-value (0.6171) indicates that they have similar performance in terms of the time cost.

To further investigate how this tradeoff will influence the optimization accuracy, Table III compares the optimization results obtained by different settings. The results show that the change of the optimization results due to the tradeoff seems to be acceptable, when considering the overall performance. First, if *gen* is 500, the results obtained by 50 surrogates have similar overall accuracy with those obtained by 100 surrogates. More specifically, the algorithm with 50 surrogates performs better than, similar to, and worse than the 100 surrogates on 2, 3, and 3 of the eight problems, respectively. As building 50 surrogates requires a much shorter time than 100 surrogates, it is reasonable to cut the T to 50. Second, when T is 50, the results after 500 generations outperform 100 generations on all the 8 problems, especially, on high dimensional problems. Although the algorithm with 500 generations requires higher time cost than the variant with 100 generations does, its improvements on optimization accuracy deserve. Concluding from the above, the BDDEA-LDG is recommended to be configured with 50 surrogates and 500 generations and, therefore, the following experiments also employ these configurations.

C. Comparisons With Traditional Methods

In this part, the BDDEA-LDG is compared with traditional methods, including GA with SBX (denoted as GA-SBX) and a random sample method. The configurations of GA-SBX are

TABLE IV
COMPARISONS ON OPTIMIZATION RESULTS BETWEEN THE PROPOSED ALGORITHM AND TRADITIONAL METHODS

Problems	D	BDDEA-LDG (11D offline data)	Random sample (11D offline data)	GA-SBX (11D online data)	GA-SBX (110D online data)	GA-SBX (550D online data)
Ellipsoid	10	**1.01E+00±3.99E-01**	1.46E+02±3.99E+01(+)	1.17E+02±2.58E+01(+)	4.72E+01±1.34E+01(+)	1.15E+00±3.99E-01(+)
	30	**6.66E+00±2.09E+00**	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	**1.31E+01±3.19E+00**	6.34E+03±5.20E+02(+)	6.09E+03±3.94E+02(+)	6.56E+02±1.45E+02(+)	**1.31E+01±2.52E+00(≈)**
	100	5.55E+01±1.12E+01	3.03E+04±1.65E+03(+)	2.67E+04±1.32E+03(+)	9.86E+02±2.18E+02(+)	**2.28E+01±6.27E+00(-)**
Rosenbrock	10	3.52E+01±8.58E+00	6.63E+02±3.16E+02(+)	5.14E+02±1.33E+02(+)	1.51E+02±5.35E+01(+)	**1.93E+01±5.45E+00(-)**
	30	5.00E+01±7.35E+00	4.83E+03±9.70E+02(+)	5.54E+03±7.71E+02(+)	6.77E+02±1.69E+02(+)	**4.82E+01±6.89E+00(-)**
	50	9.81E+01±8.99E+00	1.07E+04±1.34E+03(+)	9.77E+03±1.38E+03(+)	6.42E+02±2.07E+02(+)	**6.13E+01±4.08E+00(-)**
	100	1.93E+02±2.26E+01	2.79E+04±1.58E+03(+)	2.40E+04±1.59E+03(+)	4.51E+02±7.75E+01(+)	**1.08E+02±2.96E+00(-)**
Ackley	10	**6.39E+00±8.38E-01**	1.91E+01±1.12E+00(+)	1.88E+01±6.81E- 1(+)	1.67E+01±1.08E+00(+)	8.68E+00±1.26E+00(+)
	30	**5.57E+00±6.33E-01**	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	**4.81E+00±3.76E-01**	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	4.71E+00±3.08E-01	2.08E+01±4.63E-02(+)	2.07E+01±5.64E-02(+)	1.09E+01±5.92E-01(+)	**4.36E+00±3.32E-01(-)**
Griewank	10	**1.29E+00±1.34E-01**	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	**1.37E+00±1.00E-01**	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	**1.42E+00±8.23E-02**	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	**1.80E+00±2.34E-01**	2.19E+03±6.76E+01(+)	1.91E+03±7.85E+01(+)	7.35E+01±1.52E+01(+)	2.93E+00±5.32E-01(+)
Rastrigin	10	6.51E+01±2.96E+01	1.04E+02±1.29E+01(+)	9.35E+01±1.02E+01(+)	7.45E+01±1.01E+01(+)	**4.73E+01±7.72E+00(-)**
	30	**1.46E+02±4.34E+01**	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	**1.90E+02±3.18E+01**	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	**4.05E+02±1.44E+02**	1.51E+03±3.85E+01(+)	1.41E+03±4.62E+01(+)	9.16E+02±2.91E+01(+)	8.10E+02±2.27E+01(+)
+/≈/-		NA	20/0/0	20/0/0	20/0/0	11/2/7
Average Ranking		1.38	4.78	4.23	3.00	1.63
Adjusted p-value		NA	**0.0000**	**0.0000**	**0.0046**	0.6171

the same as that used in BDDEA-LDG, and the difference between them is that GA-SBX only employs real FEs for the evolution while the evolution of BDDEA-LDG is driven by data and surrogates. In addition, the random sample method is actually the offline data sampled by LHS, which is an ideal baseline to observe the effectiveness of BDDEA-LDG. Also, GA-SBX with 110-D and 550-D FEs are also employed for companions, which can help figure out the strengths of BDDEA-LDG.

The comparison results provided in Table IV are analyzed by Friedman test with the Bergmann–Hommel *post-hoc* test (significance level = 0.05), where the control method is BDDEA-LDG. The results indicate the effectiveness of BDDEA-LDG. First, Table IV shows that the BDDEA-LDG outperforms GA-SBX when given the same budgets (11-D FEs) on all the problems, reflecting the advantages of using surrogates. Furthermore, BDDEA-LDG can still outperform the GA-SBX with 110-D FEs and have competitive performance when compared with the GA-SBX with 550-D FEs. That is, BDDEA-LDG is able to use 10% FEs budgets to generate better results and 2% budgets to generate competitive results when compared with GA-SBX. Second, the BDDEA-LDG also produces better results than the random sample method on all the benchmark problems. This illustrates that the performance of the BDDEA-LDG is not by chance, but by its appropriate data generation and model management.

D. Comparisons With Offline Data-Driven Evolutionary Algorithms

This part compares the offline DDEAs on all the benchmark problems and provides the results in Table V. Although DDEA-SE is a state-of-the-art offline DDEA, the BDDEA-LDG can obtain better overall performance than the DDEA-SE. According to the Wilcoxon's rank-sum tests, the BDDEA-LDG performs significantly better than, similar to, and significantly worse than DDEA-SE on 8, 4, and 4 problems, respectively. Moreover, BDDEA-LDG produces the best optimization results (as marked in bold) on 12 of the 20 problems. According to the Friedman test with the Bergmann–Hommel *post-hoc* test (significance level = 0.05 and with the BDDEA-LDG as the control method), BDDEA-LDG has a smaller ranking value than DDEA-SE (p-value = 0.1175).

TABLE V
COMPARISONS BETWEEN OFFLINE DDEAS

Problem	D	BDDEA-LDG	DDEA-SE
Ellipsoid	10	**1.01E+00±3.99E-01**	1.02E+00±4.90E-01 (≈)
	30	6.66E+00±2.09E+00	**5.09E+00±1.30E+00 (-)**
	50	**1.31E+01±3.19E+00**	1.51E+01±4.63E+00 (+)
	100	**5.55E+01±1.12E+01**	3.12E+02±6.13E+01 (+)
Rosenbrock	10	3.52E+01±8.58E+00	**2.95E+01±5.04E+00 (-)**
	30	**5.00E+01±7.35E+00**	5.67E+01±5.34E+00 (+)
	50	9.81E+01±8.99E+00	**8.41E+01±4.05E+00 (-)**
	100	**1.93E+02±2.26E+01**	2.65E+02±2.48E+01 (+)
Ackley	10	**6.39E+00±8.38E-01**	6.40E+00±1.14E+00 (≈)
	30	5.57E+00±6.33E-01	**4.83E+00±5.10E-01 (-)**
	50	**4.81E+00±3.76E-01**	4.82E+00±3.85E-01 (≈)
	100	**4.71E+00±3.08E-01**	7.27E+00±7.09E-01 (+)
Griewank	10	**1.29E+00±1.34E-01**	1.31E+00±1.46E-01 (≈)
	30	1.37E+00±1.00E-01	**1.34E+00±7.46E-02 (≈)**
	50	**1.42E+00±8.23E-02**	1.94E+00±2.45E-01 (+)
	100	**1.80E+00±2.34E-01**	1.81E+01±2.12E+00(≈)
Rastrigin	10	**6.51E+01±2.96E+01**	6.59E+01±1.89E+00(≈)
	30	**1.46E+02±4.34E+01**	1.85E+02±1.61E+01(+)
	50	1.90E+02±3.18E+01	**1.87E+02±3.03E+01(≈)**
	100	**4.05E+02±1.44E+02**	8.11E+02±8.26E+01(+)
+/≈/-		NA	8/8/4
Average Ranking		1.33	1.68
Adjusted p-value		NA	0.1175

TABLE VI
COMPARISONS BETWEEN THE PROPOSED ALGORITHM AND ONLINE DDEAS ON LOW- AND MEDIUM-DIMENSIONAL PROBLEMS

Problem	*D*	Metric	BDDEA-LDG	CAL-SAPSO	GPEME
Ellipsoid	10	Mean	1.01E+00	**9.70E-01 (≈)**	3.64E+01 (+)
		Std.	3.99E-01	8.30E-01	1.68E+01
	30	Mean	6.66E+00	**4.05E+00 (-)**	1.19E+03 (+)
		Std.	2.09E+00	1.11E+00	2.12E+02
Rosen--brock	10	Mean	3.52E+01	**1.74E+01 (-)**	1.80E+02(+)
		Std.	8.58E+00	3.90E+00	6.54E+01
	30	Mean	**5.00E+01**	5.18E+01 (+)	2.68E+03 (+)
		Std.	7.35E+00	1.01E+01	8.21E+02
Ackley	10	Mean	**6.39E+00**	2.01E+01 (+)	1.41E+01 (+)
		Std.	8.38E-01	2.40E-01	2.30E+00
	30	Mean	**5.57E+00**	1.67E+01 (+)	1.93E+01 (+)
		Std.	6.33E-01	2.70E-01	3.00E-01
Griewank	10	Mean	**1.29E+00**	**1.29E+00 (≈)**	2.95E+01 (+)
		Std.	1.34E-01	1.40E-01	1.17E+01
	30	Mean	1.37E+00	**1.08E+00 (-)**	2.71E+02 (+)
		Std.	1.00E-01	3.87E-02	4.72E+01
Rastrigin	10	Mean	**6.51E+01**	8.87E+01 (+)	7.08E+01 (+)
		Std.	2.96E+01	2.15E+01	1.23E+01
	30	Mean	1.46E+02	**8.54E+01 (-)**	3.02E+02 (+)
		Std.	4.34E+01	1.76E+01	2.86E+01
+/≈/-			NA	4/2/4	10/0/0
Average Ranking			1.5	1.68	2.82
Adjusted *p*-value			NA	0.6698	**0.0060**

The above results show the effectiveness of BDDEA-LDG. Its outstanding performance is likely brought by the LDG and the BS, which can improve the surrogate performance according to the features and characteristics of current problems. In addition, the BDDEA-LDG is more likely to yield promising results on high dimensional problems. On all the benchmarks with 100 decision variables, BDDEA-LDG outperforms the DDEA-SE significantly. It seems that 11-D data are not enough for locating the optimum in high dimensional problems. Therefore, employing LDG to generate data may provide more useful information and then enhance solution accuracy. In general, BDDEA-LDG can be considered as competitive in solving offline data-driven optimization problems.

E. Comparisons With Online Data-Driven Evolutionary Algorithms

This part compares BDDEA-LDG with state-of-the-art online DDEAs. As CAL-SAPSO and GPEME are proposed for low and medium dimensional problems while SA-COSO and MGP-SLPSO are for high dimensional problems [18], [19], [47], the comparisons are divided into two parts, problems within 30 dimensions and problems exceeding 30 dimensions. Also, the CAL-SAPSO and GPEME are only compared in 10- and 30-D problems and the SA-COSO and MSP-SLPSO in 30-, 50-, and 100-D problems, as the literature does [15].

Table VI provides the comparison results on 10- and 30-D problems, showing that the BDDEA-LDG can outperform GPEME and obtain competitive results when compared with CAL-SAPSO. Although CAL-SAPSO obtains the best results in six problems while the BDDEA-LDG only in 5, BDDEA-LDG performs better than and similar to CAL-SAPSO on 4 and 2 problems, respectively. Moreover, the Friedman test with the Bergmann–Hommel *post-hoc* test (significance level = 0.05) shows that BDDEA-LDG has a smaller average ranking than CAL-SAPSO and the *p*-value indicate that the BDDEA-LDG performs similar to CAL-SAPSO. These comparisons support the effectiveness of BDDEA-LDG on low and medium problems.

Table S.I in the supplementary material provides the comparison results on 30-, 50-, and 100-D problems. In Table S.I in the supplementary material, BDDEA-LDG is shown to be efficient on medium and high dimensional problems, significantly outperforming SA-COSO and MGP-SLPSO on 15 and 9 test problems, respectively. According to the Friedman test with the Bergmann–Hommel *post-hoc* test (significance level = 0.05 and BDDEA-LDG as the control method), BDDEA-LDG shows significant improvements over SA-COSO and obtains best ranking among the three algorithms. Furthermore, the experiments show that BDDEA-LDG and MGP-SLPSO are suitable for different kinds of problems. For example, MGP-SLPSO outperforms BDDEA-LDG on Ellipsoid and Griewank problems at all the tested dimensions while BDDEA-LDG outperforms MGP-SLPSO on Rosenbrock, Ackley, and Rastrigin problems at all the tested dimensions. Nevertheless, BDDEA-LDG significantly outperforms MGP-SLPSO on nine problems while it is only significantly beaten by MGP-SLPSO on six problems, showing that in general BBDDEA-LDG has better performance than MGP-SLPSO on these problems.

F. Contribution Analysis of Different Components in the Proposed Algorithm

This part further studies the contributions and influences of BS and LDG individually.

First, the experiments are conducted to compare different surrogate strategies. That is, the same optimizer, GA-SBX, are configured with different surrogate models to develop four variants of BDDEA-LDG: 1) the original BDDEA-LDG; 2) the variant without BS; 3) the variant without LDG; and 4) the variant without both BS and LDG. The above four algorithms are simply denoted as BDDEA-LDG, DDEA-LDG-w/o-BS, BDDEA-w/o-LDG, and DDEA-w/o-BS-LDG, respectively. DDEA-LDG-w/o-BS adopts the single RBFNN built on data after LDG, BDDEA-w/o-LDG employs the simple ensemble of 50 RBFNNs built on original offline data, and DDEA-w/o-BS-LDG uses a single RBFNN built on original offline data. Table VII provides the optimization results and average ranking values while Table S.II in the supplementary material provides the *p*-value obtained by Friedman test with the Bergmann–Hommel *post-hoc* test (significance level = 0.05). According to average ranking, BDDEA-LDG is the best among all the four algorithms, followed by DDEA-LDG-w/o-LDG and BDDEA-w/o-BS, while the DDEA-w/o-BS-LDG is the worst. The ranking results show that using BS or LDG is better than not using them, and the combination of BS and LDG can obtain better results than using one of them. Furthermore, according to the Wilcoxon's rank-sum test, BDDEA-LDG significantly outperforms DDEA-LDG-w/o-BS and BDDEA-w/o-LDG on 20 and

TABLE VII
COMPARISONS BETWEEN ALGORITHM VARIANTS WITH OR WITHOUT BS AND LDG

Problem	D	BDDEA-LDG	DDEA-LDG-w/o-BS	BDDEA-w/o-LDG	DDEA-w/o-BS-LDG
Ellipsoid	10	**1.01E+00±3.99E-01**	3.47E+00±2.26E+00(+)	1.34E+00±7.30E-01(+)	3.01E+00±1.67E+00(+)
	30	**6.66E+00±2.09E+00**	2.41E+01±1.38E+01(+)	7.35E+00±2.28E+00(+)	1.83E+01±6.93E+00(+)
	50	**1.31E+01±3.19E+00**	5.00E+01±2.20E+01(+)	1.91E+01±6.38E+00(+)	6.76E+01±3.74E+01(+)
	100	**5.55E+01±1.12E+01**	2.46E+02±1.89E+02(+)	4.79E+02±2.63E+02(+)	1.24E+03±9.66E+02(+)
Rosenbrock	10	3.52E+01±8.58E+00	5.36E+01±3.62E+01(+)	**3.14E+01±7.23E+00(-)**	4.64E+01±1.46E+01(+)
	30	**5.00E+01±7.35E+00**	9.90E+01±2.28E+01(+)	6.84E+01±8.07E+00(+)	9.59E+01±2.17E+01(+)
	50	**9.81E+01±8.99E+00**	1.27E+02±1.69E+01(+)	1.07E+02±1.63E+01(+)	1.56E+02±3.77E+01(+)
	100	**1.93E+02±2.26E+01**	3.12E+02±1.65E+02(+)	4.12E+02±1.43E+02(+)	8.41E+02±7.14E+02(+)
Ackley	10	**6.39E+00±8.38E-01**	9.12E+00±2.27E+00(+)	6.45E+00±1.07E+00(+)	7.71E+00±1.47E+00(+)
	30	5.57E+00±6.33E-01	7.26E+00±1.40E+00(+)	**5.32E+00±5.31E-01(-)**	7.39E+00±1.01E+00(+)
	50	4.81E+00±3.76E-01	6.59E+00±8.70E-01(+)	**4.77E+00±2.32E-01(-)**	6.95E+00±8.97E-01(+)
	100	**4.71E+00±3.08E-01**	7.49E+00±1.15E+00(+)	4.99E+00±6.67E-01(+)	8.15E+00±9.82E-01(+)
Griewank	10	**1.29E+00±1.34E-01**	2.36E+00±1.31E+00(+)	1.36E+00±2.23E-01(+)	2.14E+00±7.71E-01(+)
	30	**1.37E+00±1.00E-01**	3.08E+00±9.33E-01(+)	1.41E+00±1.73E-01(+)	2.68E+00±5.63E-01(+)
	50	**1.42E+00±8.23E-02**	3.45E+00±1.45E+00(+)	1.57E+00±1.69E-01(+)	3.61E+00±1.45E+00(+)
	100	**1.80E+00±2.34E-01**	1.10E+01±6.79E+00(+)	1.68E+01±2.13E+01(+)	8.97E+01±1.00E+02(+)
Rastrigin	10	**6.51E+01±2.96E+01**	9.94E+01±2.60E+01(+)	6.98E+01±2.53E+01(+)	8.85E+01±2.11E+01(+)
	30	**1.46E+02±4.34E+01**	2.27E+02±4.55E+01(+)	1.57E+02±3.61E+01(+)	2.28E+02±5.11E+01(+)
	50	**1.90E+02±3.18E+01**	3.82E+02±5.66E+01(+)	2.23E+02±5.34E+01(+)	3.98E+02±8.86E+01(+)
	100	**4.05E+02±1.44E+02**	8.06E+02±1.62E+02(+)	8.76E+02±1.90E+02(+)	1.04E+03±8.00E+01(+)
+/≈/-		NA	20/0/0	17/0/3	20/0/0
Average Ranking		1.15	3.20	2.05	3.60

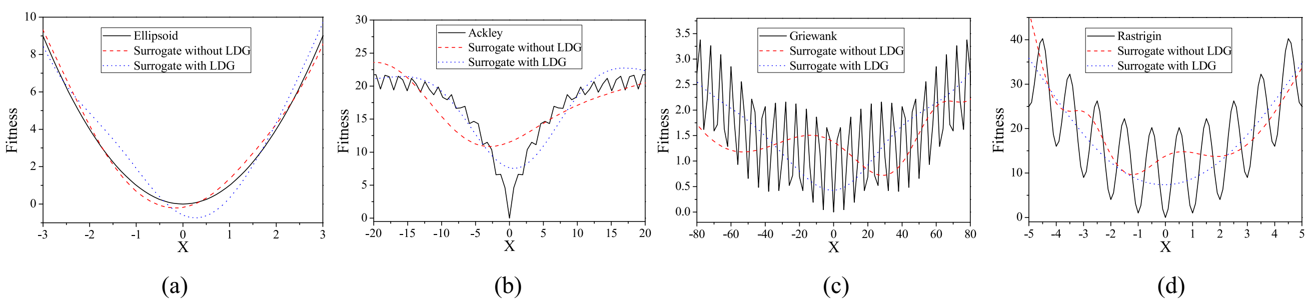


Fig. 3. Approximation result of single-RBF surrogates with or without LDG. (a) Ellipsoid function. (b) Ackley function. (c) Griewank function. (d) Rastrigin function.

17 of the total 20 test problems, respectively, indicating that both the BS and LDG contribute to the promising performance of BDDEA-LDG.

Furthermore, in order to provide more observations about how the LDG works, approximation results of surrogates with or without LDG are plotted in Fig. 3. For fair comparisons, the two surrogates are configured as single RBFNN with the same parameters and use the same amount (11-D) of offline data for model training. To plot clearer and more obvious differences of approximations, LDG is performed ten times and experiments are carried out on 1-D problems. In addition, the Rosenbrock function is not employed, for it will degenerate to a simple convex quadratic function when the dimension decreases to 1. In Fig. 3, on functions with multiple local optima, surrogates with LDG can obtain smoother approximated curves, which can be easier for EAs to optimize. Furthermore, on Ackley, Griewank, and Rastrigin functions, the global optimum of surrogates using LDG is closer to the real one. The above visualizations suggest the advantages of LDG.

In addition, the experiments are also conducted to test the effectiveness of BS and LDG on other surrogate models. Table S.III in the supplementary material compares the optimization results of three models: 1) Kriging model (also known as Gaussian process model); 2) Kriging model using LDG; and 3) Kriging model using both BS and LDG. They are denoted as Boosting Kriging with LDG (BKriging-LDG), Kriging model with LDG but without BS (Kriging-LDG-w/o-BS), and Kriging model without both BS and LDG (Kriging-w/o-BS-LDG), respectively. These three models are obtained as follows. First, a Kriging model, K_{offline} is built on the offline data. Second, K_{offline} is employed to select data to perform LDG and then build the second Kriging model, K_{LDG}, on the data set containing both offline and synthetic data. Subsequently, the Kriging-LDG-w/o-BS will employ the K_{LDG}, while Kriging-w/o-BS-LDG will use the K_{offline}. As for BKriging-LDG, only once LDG is carried out because the training time of Kriging is long and will increase rapidly as the data size increases. That is, the average prediction of K_{offline} and K_{LDG} is adopted in BKriging-LDG as the final prediction. The parameters of both the K_{offline} and K_{LDG} are configured the same according to [13] and the employed optimizers are the same GA-SBX. The multiple comparisons among these four models are performed by Friedman test with the Bergmann–Hommel *post-hoc* test(significance level

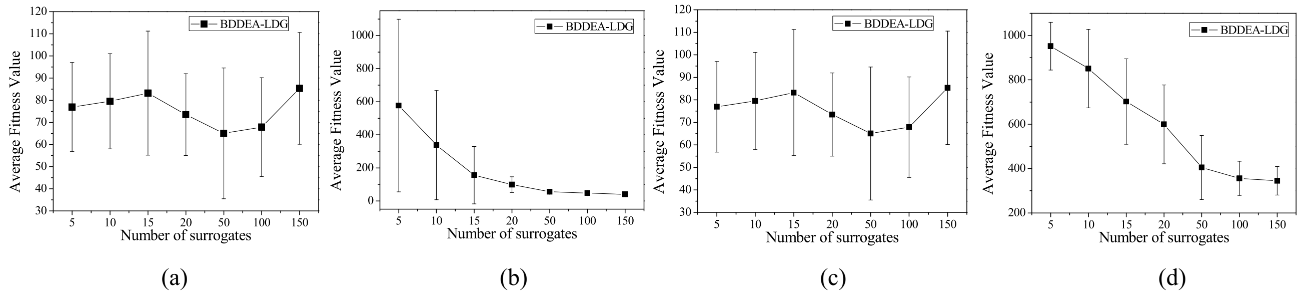


Fig. 4. Average fitness value obtained by BDDEA-LDG with different surrogate number. (a) Ellipsoid function. (b) Ackley function. (c) Griewank function. (d) Rastrigin function.

= 0.05), where their average ranking values are shown in Table S.III in the supplementary material and the p-values are given in Table S.IV in the supplementary material. The results in Table S.III in the supplementary material show that the BS and LDG can be useful for other surrogate models like the Kriging model, because BKriging-LDG can significantly outperform Kriging-LDG-w/o-BS and Kriging-w/o-BS-LDG on 8 and 9 of the 20 problems, respectively. In terms of the average ranking value, BKriging-LDG can also have the best ranking among the three algorithms. Moreover, the LDG has shown to be effective because Kriging-LDG-w/o-BS can obtain a better average ranking value than Kriging-w/o-BS-LDG (2.07 versus 2.30). These show that the BS and LDG can be useful for different kinds of surrogates, including RBFNNs and Kriging models.

To obtain further observations, visualizations of the approximations obtained by the above three models are provided in Fig. S.1 in the supplementary material, which are on 1-D problems. Fig. S.1 in the supplementary material shows that the landscapes approximated by BKriging-LDG can have more accurate positions of the global optima than Kriging-w/o-BS-LDG and Kriging-LDG-w/o-BS, and those approximated by Kriging-LDG-w/o-BS are also better than Kriging-w/o-BS-LDG. These results validate the effectiveness of BS and LDG on the Kriging model.

G. *Influences of Surrogate Number in Boosting Strategy*

This part investigates the influence of surrogate number T. BDDEA-LDG variants with different surrogate numbers, such as 5, 10, 15, 20, 50, 100, and 150, are compared on Ellipsoid and Rastrigin problems at 10 and 100 dimensions.

The results shown in Fig. 4 indicate that the effect of the surrogate number has a strong relationship with the problem dimension. On the one hand, for 10-D problems, the obtained fitness first increases and then decreases along with the increase of the surrogate number. Furthermore, on the Ellipsoid problem, 50 to 100 surrogates are preferred for better results, while on the Rastrigin problem, 20 to 100 surrogates can produce a smaller error. However, the algorithm with 150 surrogates performs poorer than 50 and 100 surrogates on both 10-D Ellipsoid and Rastrigin functions. The reason for the poor performance of 150 surrogates may be the over fitting problem, where the surrogate model approximates too close to the evaluated data but fails to predict the new data correctly. On the other hand, on 100-D problems, solution accuracy improves as the surrogate number increases, indicating that 150 or more surrogates will be better. This suggests that, as the problem complexity and the number of local optima increases rapidly, a surrogate model that overfits in low dimensional problems may be not complex enough for approximating high dimensional problems. For example, in Fig. 4, the algorithm with 150 surrogates overfits in 10-D problems and perform worse than 50 and 100 surrogates. But on 100-D problems, the solution produced by 150 surrogates are more accurate than 50 and 100 surrogates. In conclusion, more surrogates can further enhance BDDEA-LDG on the solution accuracy within a range and the range tends to enlarge as the problem dimension increases.

H. *Influences of Configuration Settings in Localized Data Generation*

This part studies the effect of configurations in LDG. As the selection criterion, the size of the neighborhood for LDG, and the size of data generated by LDG may have effects on the algorithms, these three settings are discussed as follows.

First, different selection criteria for constructing the S in (1) are compared experimentally. As the original criterion used in LDG is diff $= Y_{\text{pre}} - F(x)$, its opposite value and absolute value are employed in comparisons. Also, diff $= -F(x)$ is employed for a baseline, because it is the real fitness value and will not be influenced by surrogate predictions. In the experiments, the parameter settings of each algorithm are the same with BDDEA-LDG except the *diff*. The results provided in Table S.V in the supplementary material presents that the original criterion significantly outperforms others. Table S.V in the supplementary material shows that diff $= F(x) - Y_{\text{pre}}$ performs significantly worse than the original one on 18 of 20 problems. In addition, diff$= \|Y_{\text{pre}} - F(x)\|_1$ and diff $= -F(x)$ are also outperformed by the original one on 18 problems. According to Friedman test with the Bergmann–Hommel *post-hoc* test (significance level = 0.05), the control method, i.e., diff $= Y_{\text{pre}} - F(x)$, obtains the best average ranking and shows significant improvement over the other three criteria. These results indicate that the original criterion can handle this problem well.

Second, the experiments are conducted to investigate the sensitiveness of neighborhood size l, which aims to control the safe region for LDG. In the experiments, BDDEA-LDG was independently configured with 10^{-1}, 10, 10^2, or 10^3 times the original l value computed by (2), which are denoted as $l^* = 10^{-1}l$, $l^* = 10l$, $l^* = 10^2 l$, and $l^* = 10^3 l$, respectively. Also, the original BDDEA-LDG is denoted as $l^* = l$, where the l value is computed by (2). The results reported in Table S.VI in the supplementary material how that the BDDEA-LDG with $l^* = l$ is the best among the five algorithms according to the Wilcoxon's rank-sum test. In Table S.VI in the supplementary material, BDDEA-LDG with $l^* = l$ significantly outperforms all the three BDDEA-LDG variants with larger l^* values on 17 of 20 problems, and only performs worse on the rest three problems. This suggests that the l value obtained by (2) can provide a safer region than larger l values to avoid most noises when performing LDG. Furthermore, when compared with smaller l^* value, the BDDEA-LDG with $l^* = l$ can perform similar to the variant with$l^* = 10^{-1}l$ on most of the problems. Nevertheless, BDDEA-LDG with $l^* = l$ still significantly outperforms BDDEA-LDG with $l^* = 10^{-1}l$ value on five problems while it is only significantly beaten on two problems, showing that the l value obtained by (2) is small enough and will not make more noises than smaller l value. Based on the above, the l value obtained by (2) is very suitable for BDDEA-LDG.

Third, the experiments are performed to study the influences of the size of data generated in LDG. As the original setting is generating 50% of $|TD|$ (i.e., $0.5|TD|$) synthetic data in each LDG, configurations of different data sizes from $0.1|TD|$ to $1.0|TD|$ are tested, where $|TD|$ is the size of the training data set that only contains offline data. Generally speaking, the size of data generated in LDG can have influences on two aspects: 1) the time cost for training surrogates and 2) the optimization performance of the algorithms. Therefore, the experiments are divided into two parts. First, Table S.VII in the supplementary material compares the average time cost for model training of the BDDEA-LDG with different sizes of generated data. Second, Table S.VIII in the supplementary material provides the optimization results obtained by BDDEA-LDG with different sizes of generated data. Moreover, the time costs and optimization results are compared using the Friedman test with the Bergmann–Hommel *post-hoc* test (significance level = 0.05), which are given in Table S.IX in the supplementary material. The results in Table S.VII in the supplementary material show that as the size of generated data increases from $0.1|TD|$ to $1.0|TD|$, the time cost for model training also increases. While in Table S.VIII in the supplementary material, the optimization results show that $0.5|TD|$ can have better overall performance than other settings. For example, $0.5|TD|$ significantly outperforms $0.1|TD|$, $0.2|TD|$, $0.3|TD|$, and $0.4|TD|$ on 17, 17, 17, and 14 problems, respectively. Furthermore, the $0.5|TD|$ obtains the best ranking value (i.e., 3.15) among the ten different settings. In addition, the p-values in Table S.IX in the supplementary material show that $0.5|TD|$ performs significantly better than $0.1|TD|$, $0.2|TD|$, $0.3|TD|$, and $1.0|TD|$, in terms of the optimization results. When considering the multiple comparisons of time cost in Table S.IX in the supplementary material, $0.5|TD|$ also shows significant improvements over $0.8|TD|$, $0.9|TD|$, and $1.0|TD|$, and performs similar to $0.4|TD|$, $0.6|TD|$, and $0.7|TD|$. Therefore, $0.5|TD|$ can be the best setting for balancing the algorithm performance and time cost, which is the recommendation in this article.

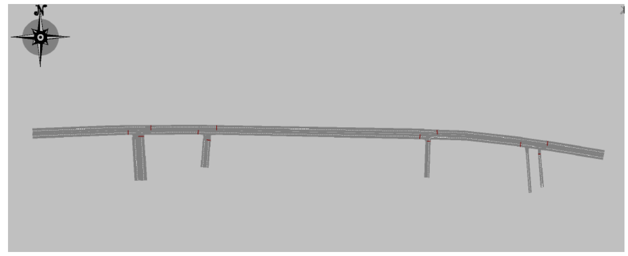


Fig. 5. Diagram of an arterial road with four intersections.

I. Arterial Traffic Signal Timing Optimization

This part employs an arterial traffic signal timing optimization problem to test the proposed algorithm. Due to the substantial increase in vehicle numbers, the traffic congestion phenomenon has received increasing attention,which means traffic demand exceeds the capacity of transportation systems [70]. To alleviate traffic congestions, optimizing traffic signal control is one of the most effective ways, especially, in arterial traffic [71]. However, the evaluation of a signal control plan is not easy and may last a week, or even a month [71]. Otherwise, the signal plans tested well during workdays may not work well on holidays, because the behaviors of the drivers and the traffic flow differ from day to day. Therefore, to shorten the evaluations, the signal control plan is often designed through simulations of professional software, like VISSIM [72], [73]. As the simulations can also be time consuming, arterial traffic signal timing optimization is an ideal place to employ DDEAs.

An arterial traffic signal timing problem can be formulated as follows [73]:

$$\min_{C,g,\theta} \quad OF(C, g, \theta) \tag{7}$$

$$\text{s.t.} \quad C_{\min} \le C \le C_{\max} \tag{8}$$

$$0 \le \theta_z \le C \quad \forall z \in Z \tag{9}$$

$$g_{\min} \le g_{z,i} \le g_{\max} \quad \forall z \in Z \ \forall i \in I \tag{10}$$

$$g_{z,1} + g_{z,1} = g_{z,5} + g_{z,6} \quad \forall z \in Z \tag{11}$$

$$g_{z,3} + g_{z,4} = g_{z,7} + g_{z,8} \quad \forall z \in Z \tag{12}$$

where OF is the objective function of three decision variables including cycle period (C), green splits (g), and offsets (θ), Z is the intersection set (each Z has 8 g), I is the signal set of an intersection (containing green, yellow, and red signals), and $C_{\max}$ and $C_{\min}$ are maximum and minimum cycle length for a complete period, $g_{\max}$ and $g_{\min}$ are the maximum and minimum of a green splits, respectively. Equations (11) and (12) are for the ring-barrier diagram strategy such that the east-west and north-south movements will not contradict each other.

The signal timing problem used in this article is a road with four intersections ($Z = 4$), both of which are T-junctions, as shown in Fig. 5. In this case, the problem dimension is 37, with

TABLE VIII
RESULTS OF THE ARTERIAL TRAFFIC SIGNAL TIMING PROBLEM

Algorithm		Average of Needed Travel Time (seconds)
Offline	BDDEA-LDG	**3.01E+02±1.57E+01**
	DDEA-SE	3.17E+02±2.01E+01
Online	CAL-SAPSO	3.12E+02±3.21E+01
	GPEME	3.35E+02±1.64E+01
Random sample		4.57E+02±6.52E+01
GA-SBX		3.79E+02±4.28E+01

four variables of θ, 32 variables of g, and one variable of C, respectively. In addition, $C_{\max}$ and $C_{\min}$ are set as 120 and 60 s, while $g_{\max}$ and $g_{\min}$ are configured as 40 and 10 s. The objective function is defined as the average travel time for each vehicle, which can be simulated by VISSIM [73]. As the timing of the signals in VISSIM has precision limits, the solution value of each dimension will be rounded off before simulations. To simulate traffic congestions, 2.2×10^4 vehicles with different behaviors and characteristics were generated according to a predefined distribution in VISSIM. Those vehicles were set with different starting points and destinations. To ensure all the vehicles can reach their destinations, each simulation would last for 10^4 simulation seconds before calculating the result.For the comparisons, SA-COSO and MGP-SLPSO were not employed because 37 is not a high dimension. To validate the effectiveness of DDEAs, the results obtained by the random sampling method and GA-SBX are also recorded. For fair comparisons, all the algorithms can only use 407 evaluations in total, namely, 11 times the problem dimension 37. In addition, to reduce accidental error, each algorithm performs 25 independent runs and the average results were used for comparisons.

Table VIII provides the experimental results with the best result marked in bold. In Table VIII, the BDDEA-LDG can obtain the best results while the DDEA-SE and the CAL-SAPSO perform similarly, which suggests the advantages of the BDDEA-LDG. Furthermore, all DDEAs outperform the GA-SBX and the random sample method, suggesting the effectiveness of DDEAs in solving this problem. In summary, the performance of our proposed algorithms has been verified by the arterial traffic signal timing problem.

V. CONCLUSION

Although DDEAs have shown efficiency in solving real-world optimization problems, there are still some difficulties in designing powerful DDEAs, especially, in data utilization and model management. In this article, a BDDEA-LDG algorithm is proposed by combining the model managements and data generation methods. It employs the BS to boost the surrogate performance according to the problems at hand, so that it can obtain suitable surrogate models for different problems. Furthermore, the LDG is proposed to alleviate the data shortage and cooperates with the BS through generating data. In addition, to make a balance between execution time and accuracy, this article empirically studies the tradeoff between the optimization procedure and the model management of BDDEA-LDG, which benefits the algorithm performance. To access the effectiveness of the proposed methods, the experiments and comparisons are conducted on widely used benchmarks and an arterial traffic signal timing optimization problem. The results show that the proposed algorithms are able to outperform state-of-the-art algorithms when given the same computational budgets, suggesting the efficiency of the proposed methods.

For future work, the algorithm proposed in this article will be applied to solve problems with more complicated challenges, such as large-scale [74], multi/many-objective [75], multimodal [76], dynamics [77], and constraint [78]. Moreover, the BS and LDG will be extended to more different types of surrogate models to further study their efficiency in improving the algorithm performance. In addition, researches will be conducted on combining the proposed strategies (i.e., BS and LDG) with other different optimization algorithms (e.g., PSO and DE [79]), so as to obtain more advanced DDEAs.

REFERENCES

[1] D. Dasgupta and Z. Michalewicz, *Evolutionary Algorithms in Engineering Applications*. Berlin, Germany: Springer, 2013.

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

[3] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: A survey," *Control Eng. Pract.*, vol. 10, no. 11, pp. 1223–1241, 2002.

[4] 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, doi: 10.1109/TCYB.2018.2869674.

[5] 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.*, San Sebastian, Spain, 2017, pp. 1541– 1548.

[6] T. Chugh, T. Kratky, K. Miettinen, Y. Jin, and P. Makonen, "Multiobjective shape design in a ventilation system with a preference-driven surrogate-assisted evolutionary algorithm," in *Proc. Conf. Genet. Evol. Comput.*, 2019, pp. 1147–1151.

[7] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Comput. Intell. Mag.*, vol. 4, no. 3, pp. 62–76, Aug. 2009.

[8] 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.

[9] 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. Processes*, vol. 32, no. 10, pp. 1172–1178, Jan. 2017.

[10] 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, Dec. 2016.

[11] 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.*, Athens, Greece, Dec. 2016, pp. 1–8.

[12] T. Chugh, C. Sun, H. Wang, and Y. Jin, "Surrogate-assisted evolutionary optimization of large problems," in *High-Performance Simulation-Based Optimization*. Cham, Switzerland: Springer Int., 2020, pp. 165–187.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.*, Edinburgh, UK, 2005, pp. 2832–2839.

[21] X. Wang, G. G. Wang, B. Song, P. Wang, and Y. Wang, "A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 815–827, Oct. 2019, doi: 10.1109/TEVC.2019.2890818.

[22] 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.

[23] 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.

[24] Z. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives," *IEEE Comput. Intell. Mag.*, vol. 9, no. 4, pp. 62–74, Nov. 2014.

[25] 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.

[26] S. Nguyen, M. Zhang, and K. C. Tan, "Surrogate-assisted genetic programming with simplified models for automated design of dispatching rules," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2951–2965, Sep. 2017.

[27] I. Voutchkov, A. Keane, A. Bhaskar, and T. M. Olsen, "Weld sequence optimization: The use of surrogate models for solving sequential combinatorial problems," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 30, pp. 3535–3551, 2005.

[28] B. Yuan, B. Li, T. Weise, and X. Yao, "A new memetic algorithm with fitness approximation for the defect-tolerant logic mapping in crossbar-based nanoarchitectures," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 846–859, Dec. 2014.

[29] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: Chapman and Hall, 2012.

[30] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. Cambridge, MA, USA: MIT Press, 2012.

[31] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[32] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Comput.*, vol. 23, no. 9, pp. 3137–3166, 2019.

[33] A. Mazumdar, T. Chugh, K. Miettinen, and M. López-Ibáñez, "On dealing with uncertainties from Kriging models in offline data-driven evolutionary multiobjective optimization," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2019, pp. 463–474.

[34] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 1000–1014, Dec. 2019.

[35] 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.

[36] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Trans. Syst. Man. Cybern. B, Cybern.*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012.

[37] 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.

[38] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

[39] 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.

[40] C. Wang, C. Xu, X. Yao, and D. Tao, "Evolutionary generative adversarial networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 921–934, Dec. 2019, doi: 10.1109/TEVC.2019.2895748.

[41] D. Büche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *IEEE Trans. Syst. Man, Cybern. C*, vol. 35, no. 2, pp. 183–194, May 2005.

[42] 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.

[43] L. Willmes, T. Back, Y. Jin, and B. Sendhoff, "Comparing neural networks and kriging for fitness approximation in evolutionary optimization," in *Proc. IEEE Congr. Evol. Comput.*, Canberra, ACT, Australia, 2003, pp. 663–670.

[44] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Proc. Conf. Genet. Evol. Comput.*, 2004, pp. 688–699.

[45] S. Z. Martínez and C. A. CoelloCoello, "MOEA/D assisted by RBF networks for expensive multi-objective optimization problems," in *Proc. Conf. Genet. Evol. Comput.*, 2013, pp. 1405–1412.

[46] D. Lim, Y. Jin, R. Cheng, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.

[47] 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.

[48] 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 Criteria Decis. Anal.*, vol. 24, nos. 1–2, pp. 5–24, 2017.

[49] M. Hüsken, Y. Jin, and B. Sendhoff, "Structure optimization of neural networks for evolutionary design optimization," *Soft Comput.*, vol. 9, no. 1, pp. 21–28, 2005.

[50] J. Branke and C. Schmidt, "Faster convergence by means of fitness estimation," *Soft Comput.*, vol. 9, no. 1, pp. 13–20, 2003.

[51] M. Binois, D. Ginsbourger, and O. Roustant, "Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 386–394, 2015.

[52] W. Ponweiser, T. Wagner, and M. Vincze, "Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, China, 2008, pp. 3515–3522.

[53] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive Black-box functions," *J. Global Optim.*, vol. 13, pp. 455–492, 1998.

[54] A. A. M. Rahat, R. M. Everson, and J. E. Fieldsend, "Alternative infill strategies for expensive multi-objective optimisation," in *Proc. Conf. Genet. Evol. Comput.*, 2017, pp. 873–880.

[55] 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.

[56] L. Bull, "On model-based evolutionary computation," *Soft Comput.*, vol. 3, no. 2, pp. 76–82, 1999.

[57] Y. Jin, M. Olhofer, and B. Sendhoff, "On evolutionary optimization with approximate fitness functions," in *Proc. Conf. Genet. Evol. Comput.*, 2000, pp. 786–793.

[58] 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.

[59] 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.

[60] 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.

[61] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[62] H. Chen, P. Tiòo, and X. Yao, "Predictive ensemble pruning by expectation propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 7, pp. 999–1013, Jul. 2009.

[63] 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.

[64] Z.-J. Wang *et al.*, "Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 894–908, Dec. 2018.

[65] 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.

[66] K. Deb and H.-G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evol. Comput.*, vol. 9, no. 2, pp. 197–221, Jun. 2001.

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

[68] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[69] 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.

[70] A. A. Malikopoulos, S. Hong, J. Lee, B. B. Park, and S. Ryu, "Optimal control for speed harmonization of automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2405–2417, Jul. 2019, doi: 10.1109/TITS.2018.2865561.

[71] J. Zhang, F.-Y. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[72] J. Garcia-Nieto, A. C. Olivera, and E. Alba, "Optimal cycle program of traffic lights with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 823–839, Dec. 2013.

[73] S. Dabiri and M. Abbas, "Arterial traffic signal optimization using particle swarm optimization in an integrated VISSIM-MATLAB simulation environment," in *Proc. 19th Int. Conf . Intell. Transp. Syst.*, Rio de Janeiro, Brazil, 2016, pp. 766–771.

[74] 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.*, early access, doi: 10.1109/TCYB.2019.2933499.

[75] 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.

[76] 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.*, early access, doi: 10.1109/TEVC.2019.2944180.

[77] X.-F. Liu *et al.*, "Neural network-based information transfer for dynamic optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, doi: 10.1109/TNNLS.2019.2920887.

[78] X. Zhang, K. Du, Z.-H. Zhan, S. Kwong, T. Gu, and J. Zhang, "Cooperative coevolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties," *IEEE Trans. Cybern.*, early access, doi: 10.1109/TCYB.2019.2937565.

[79] 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.

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, and their applications in real-world problems, and in environments of distributed computing and big data.

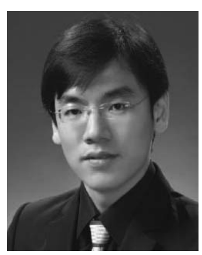
Zhi-Hui Zhan (Senior Member, IEEE) received the bachelor's and the Ph.D. degrees in computer science from the Sun Yat-Sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently the Changjiang Scholar Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His current research interests include evolutionary computation algorithms and their applications.

Dr. Zhan's doctoral dissertation was awarded the IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation. He 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. 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 and *Neurocomputing*.

Chuan Wang received the bachelor's degree in computer science and the master's degree in education from Henan Normal University, Xinxiang, China, in 1999 and 2009, respectively.

He is currently an Associate Professor with the College of Software, Henan Normal University. His current research include computational intelligence and its applications on intelligent information processing and big data.



Hu Jin (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2011.

He is currently an Associate Professor with the Division of Electrical Engineering, Hanyang University, Ansan, South Korea. His research interests include wireless communications, Internet of Things, and machine learning.

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

He is currently a Visiting Scholar with Hanyang University, Ansan, South Korea. His current research interests include computational intelligence algorithms and their applications.

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