Offline Data-Driven Evolutionary Optimization Using Selective Surrogate Ensembles

Handing Wang, *Member, IEEE,* Yaochu Jin, *Fellow, IEEE,* Chaoli Sun, *Member, IEEE,* John Doherty

***Abstract*—In solving many real-world optimization problems, neither mathematical functions nor numerical simulations are available for evaluating the quality of candidate solutions. Instead, surrogate models must be built based on historical data to approximate the objective functions and no new data will be available during the optimization process. Such problems are known as offline data-driven optimization problems. Since the surrogate models solely depend on the given historical data, the optimization algorithm is able to search only in a very limited decision space during offline data-driven optimization. This paper proposes a new offline data-driven evolutionary algorithm to make the full use of the offline data to guide the search. To this end, a surrogate management strategy based on ensemble learning techniques developed in machine learning is adopted, which builds a large number of surrogate models before optimization and adaptively selects a small yet diverse subset of them during the optimization to achieve the best local approximation accuracy and reduce the computational complexity. Our experimental results on the benchmark problems and a transonic airfoil design example show that the proposed algorithm is able to handle offline data-driven optimization problems with up to 100 decision variables.**

***Index Terms*—Offline data-driven optimization, surrogate, evolutionary algorithm, ensemble, radial basis function networks.**

I. Introduction

Evolutionary algorithms (EAs) have been shown to be effective in a number of real-world optimization applications [1], [2]. One assumption of most EAs is that computationally cheap analytical functions are available for calculating the quality of candidate solutions, enabling EAs to afford a large number of fitness evaluations. This assumption does not hold, unfortunately, for many real-world optimization problems, where either computationally intensive numerical simulations or expensive experiments must be performed for fitness evaluations [3], [4]. To solve these expensive optimization problems, it is essential to employ computationally cheap surrogate models to assist EAs in an attempt to reduce the required expensive fitness evaluations [5].

Most existing surrogate-assisted evolutionary algorithms (SAEAs) assume that a small number of expensive real fitness evaluations, either numerical simulations or experiments, can still be conducted, which is known as online data-driven optimization [6]. Thus, the main concern in most existing SAEAs is to properly update the surrogate model by making the best use of the allowed expensive real fitness evaluations, known as evolution control or model management [7]. Many regression or classification techniques can be used as surrogate models in SAEAs, such as radial basis function networks (RBFNs) [8], [9], Kriging models [10], [11], [12], [13], polynomial regression (PR) models [14], among many others. To improve the accuracy in fitness approximation, surrogate ensembles have also been used [15], [16], [17], [18], [19], [20].

Many empirical model management strategies have been developed for online data-driven EAs [5], [21], where the main idea is either to enhance the accuracy of the surrogates [22], [23], to ensure correct environmental selection [24], or to encourage exploration [12], [18], [25], [26]. Another idea is to use a combination of global and local surrogate models in which the global model is used to smoothen out the local optimums while the local ones are utilized for exploiting the local details of the fitness landscape [27], [28], [29].

A class of more formal model management strategies are known as infill sampling criteria [30], [31], which help select the next solution to be evaluated using the expensive fitness function. Three main infill sampling criteria have been suggested, namely, maximizing the predicted fitness, maximizing the prediction uncertainty, or combining the previous two criteria, which in principle agree with the empirical model management strategies. Infill criteria, including expected improvement [32], lower confidence bound (LCB) [33], [34], and probability of improvement [35], are most widely used in Kriging or Gaussian process assisted EAs. Most recently, infill criteria have been extended to surrogates consisting of heterogeneous ensembles [36].

While most SAEAs focus on developing model management strategies for online data-driven optimization, relatively little effort has been dedicated to offline data-driven optimization with a few exceptions [6], [37], [38], where no new data can be made available for managing the surrogates. Offline data-driven optimization poses new challenges to SAEAs, and how to address the challenges heavily depends on the problem to be solved and the amount of historical data. For instance, in trauma system design [6], the historical data are

This work was supported in part by an EPSRC grant (No. EP/M017869/1) and in part by the National Natural Science Foundation of China (No. 61590922).

H. Wang is with the Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: wanghanding.patch@gmail.com).

Y. Jin is with the Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: yaochu.jin@surrey.ac.uk). He is also affiliated with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China, and Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. (*Corresponding author*)

C. Sun is with the Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. She is also affiliated with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110004, China. (e-mail: chaoli.sun.cn@gmail.com)

J. Doherty is with the Department of Mechanical Engineering Sciences, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: john.doherty@surrey.ac.uk)

the emergency records collected within one year in Scotland, and the objectives and constraints can be evaluated using the data. Since a large amount of data is available and consequently the main challenge is to reduce the computation time for fitness evaluations, a model management strategy was proposed to adjust the fidelity of the surrogate model according to the optimization process [6]. By contrast, only a small amount of historical data from manufacturing processes is available for blast furnace optimization [37]. As the data is very noisy, the data must be preprocessed before being used for constructing the surrogate. In another example of optimization of fused magnesium furnaces [38], only historical data from manufacturing processes is available for optimization. The idea is to construct a smooth global PR model before optimization starts and to use this model as the real fitness function for managing local surrogates during the optimization.

In this work, we aim to design a generic and problem-independent offline data-driven EA. The main challenge here is to make full use of the available historical data to guide the evolutionary search. To this end, a large number of surrogates are generated offline using the bagging technique [39], [40] and a subset of them is adaptively selected for fitness estimation as the evolutionary optimization process proceeds. We term the proposed algorithm data-driven evolutionary algorithm using selective ensemble (DDEA-SE).

The rest of this paper is organized as follows. In Section II, main challenges in data-driven evolutionary optimization are discussed together with a short review of the related work. Then, bagging is briefly introduced in Section III. Section IV describes the details of the proposed algorithm, focusing on the generation of ensembles using bagging and selection of the ensemble subset. To further analyze the behavior of the proposed algorithm, experimental results on benchmark problems and an example of transonic wing system design are presented in Sections V and VI. Section VII concludes the paper and suggests a few possible future research directions for offline data-driven evolutionary optimization.

II. Offline Data-Driven Evolutionary Optimization

A wide range of real-world optimization problems can be solved only using offline data-driven optimization approaches, as no new data can be made available during the optimization [6], [37], [38]. As shown in Fig. 1, offline data-driven EAs can be divided into three main parts, i.e., data collection, surrogate modeling and management, and optimization. At first, data is collected, and pre-processed if necessary. Before optimization starts, the optimization problem needs to be properly formulated, including the specification of the fitness and constraint functions to be approximated by the surrogates. Then, global surrogate models are built using the historical data. Finally, an optimizer performs optimization by searching the surrogates created offline, although local surrogates can also be constructed during the optimization using the historical data.

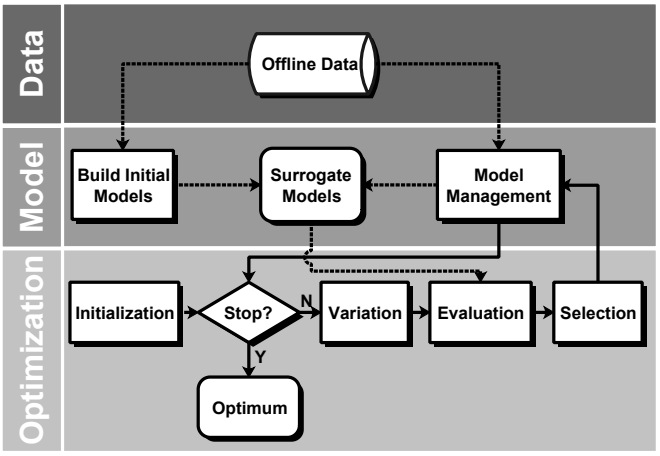

Fig. 1. A diagram of a generic offline data-driven EA.

A. Main Challenges

The available data may pose a challenge to the model management strategy, including the strategy for model selection and the infill criterion, regardless whether offline or online data-driven EAs are used. In contrast to online data-driven EAs, however, offline data-driven EAs have no chance to sample new data to improve the quality of the surrogate models or to validate the found optima. These make offline data-driven EAs more challenging than online data-driven EAs, particularly when the data is imbalanced [41], [42], [43], noisy [44], time-varying [45] or heterogeneous [46].

As the surrogate models cannot be updated during the optimization in offline data-driven EAs, the quality of the surrogates created before the optimization starts becomes especially important in offline data-driven EAs. Therefore, the main challenge lies in the construction of surrogates of sufficiently high quality in case only a limited amount of data are available.

B. Countermeasures

To address the above challenge, effective countermeasures that enhance the quality of data or models must be taken in designing offline data-driven EAs. On the one hand, enhancing the quality of the offline data can indirectly improve the quality of the surrogate models. On the other hand, surrogate models built from very limited data are expected to be able to guide the search properly. Although not much research on offline data-driven EAs has been reported, the following ideas can be used to handle the aforementioned challenges.

- Pre-processing the data. The non-ideal nature of the offline data may seriously degrade the quality of the surrogate models. Thus, pre-processing the offline data is indispensable to reduce noise or to remove outliers, e.g., in offline optimization of blast furnaces [37].
- Creating synthetic data. As lack of data is one main challenge in offline data-driven EAs, one straightforward idea is to generate a certain amount of synthetic data to augment the available historical data for updating the surrogate models. Synthetic data can be generated using

a surrogate model [38] or resampling the given data [39], [42], [43].
- Transferring knowledge from other optimization problems. Multi-tasking optimization [47], [48], [49] provides an effective means to transfer knowledge between different problems to speed up optimization. Thus, transfer learning [50] can be extended to SAEAs to alleviate the issue of data paucity [51].
- Employing advanced machine learning techniques. For example, semi-supervised learning [52] can be used to address data paucity [53], ensemble learning [54] can be employed to enhance the prediction performance of surrogate models [55], and clustering techniques can be adopted to reduce the amount of data to save computation time for each fitness evaluation in an offline data-driven trauma system design application [6].

III. Preliminaries of Selective Bagging

Ensemble learning refers to a class of machine learning methods that construct a set of base learners and combine them to create a strong learner [54]. Ensembles have been shown to have advantages over single learners in terms of accuracy and robustness [56]. Bootstrap aggregating [39] (bagging for short) and boosting [57] are two popular ensemble generation methods. Bagging is a parallel ensemble method minimizing variance while boosting is a sequential ensemble method minimizing bias [58]. In SAEAs, bias introduced by surrogates is less critical as long as the ranking of candidate solutions is correct. Out of this reason, this work adopts bagging for generating surrogate ensembles. To further improve the approximation quality of ensembles, a subset of base learners are selected [59] for calculating the output. Bagging algorithms with model selection strategies are known as selective bagging. Fig. 2 is a diagram showing the process of selective bagging, which consists of bootstrap sampling, model training, model selection, and model combination [57].

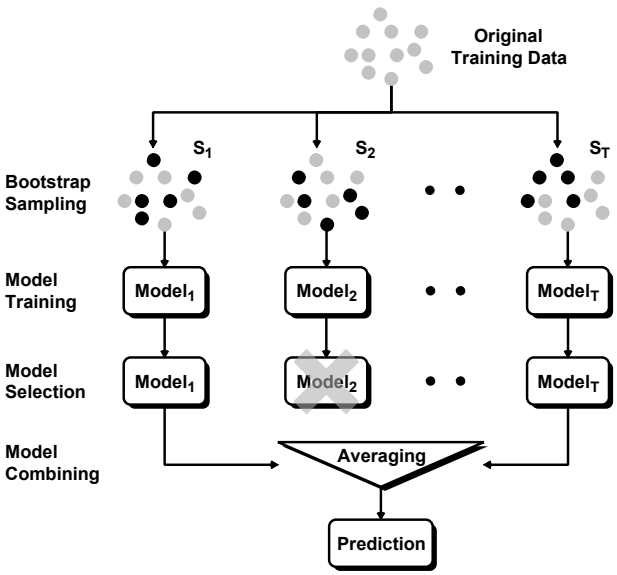

Fig. 2. A diagram of selective bagging.

A. Model Generation

Model generation in selective bagging is composed of bootstrap sampling and model training. First, bootstrap sampling [60] is performed for T independent times to generate T data subsets (S_1, S_2,...,S_T) resampled from the original data. As shown in Fig. 2, each data subset contains a random portion of the original data, which is denoted by black dots. Then T different models are generated, each using one of the T datasets.

It is well known that the accuracy of a bagging ensemble converges as the size of the ensemble (T) increases [57]. Further, since a higher degree of ensemble diversity is expected to deliver better performance, highly nonlinear models that have a large change in their output in response to a small change in the input are usually preferred in bagging [54]. Those data points left out in each data subset, termed out-of-bag samples [61], result in diversity in ensembles. For bootstrap sampling without replacement [62], typically half of the original dataset size (known as half-sampling [40]) is used as the number of out-of-bag samples [63], leading to a well-performing bagging ensemble [64].

B. Model Combination

Model combination in selective bagging consists of model selection and averaging. Before combining the models, only Q of T ($Q < T$) models are selected to produce the ensemble output. An illustrative example is shown in Fig. 2, where the second model is not selected for generating the final output of the ensemble. In this work, the final output of the ensemble is the plain average of the outputs of the selected models.

The model selection strategies play an important role in selective bagging, which affect the accuracy, diversity and computational efficiency. In fact, the process can be formulated as a combinatorial optimization problem where the decision variables are T models and the objective is the accuracy and/or diversity. Existing model selection strategies can be classified into two categories depending on whether global or local search is employed [65]. Global search strategies include sparse optimization [66], [67], genetic algorithms [68], and clustering [69]. By contrast, local search based model selection strategies are greedy, which successively add models by starting from an empty set, or successively delete models from the full set of models. The criterion to evaluate whether a model should be selected or not can be based on complementarity, orientation, or margin distance [70]. It has been shown that the local search based model selection strategies are computationally more efficient than the global search based strategies [71].

IV. Proposed Algorithm

Offline data-driven EAs distinguish themselves from online data-driven EAs in many aspects. Whereas online data-driven EAs can use various infill sampling criteria [30], [31] to include additional training data for updating the surrogates during the optimization, offline data-driven EAs have no access to the real fitness evaluations and no model update can be carried out. In addition, online data-driven EAs are able to

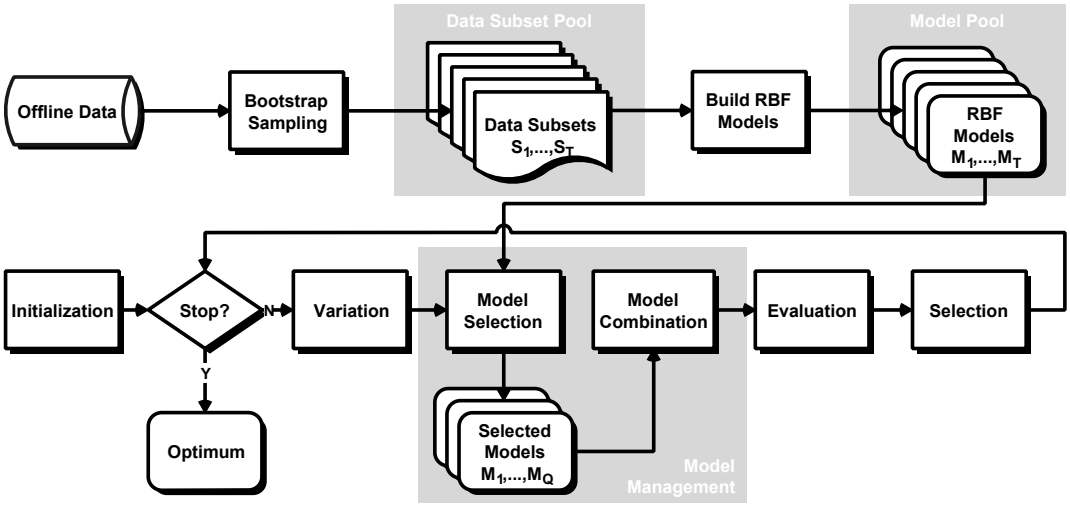

Fig. 3. A generic diagram of DDEA-SE.

validate the optimums found so far during the optimization, but unfortunately, offline data-driven EAs have no opportunity to validate the solutions before they are actually implemented. As a result, offline data-driven EAs should focus on building high-quality surrogate models based on the offline data. To address the above challenges, we proposed a novel offline data-driven EA assisted by a selective ensemble (DDEA-SE).

A. The Framework

A generic diagram of DDEA-SE is shown in Fig. 3. Before running the optimizer (a canonical EA), offline data is created, from which T subsets (S_1, S_2,..., S_T) are generated using bootstrap. Then, T models (M_1, M_2,...,M_T) are independently built based on T subsets. During the optimization, DDEA-SE selects Q ($Q \leq T$) models from T surrogates using a model selection strategy, and the fitness values are estimated by combining those Q models. When the stopping criterion is met, DDEA-SE outputs the final optimal solution.

In the following, we will present the details for building the surrogate ensemble via bagging and surrogate management.

B. Ensemble Generation

Before initializing the population for optimization, DDEA-SE creates training data subsets using bootstrap sampling and builds surrogate ensembles using a proper learning algorithm. As recommended in [57], highly nonlinear models are preferred as base learners in bagging. Thus, we employ RBF networks, which are highly nonlinear, as basic learners to build the surrogate model pool. Accordingly, a large model pool size (T) can be used.

As discussed in [64], the optimal number of out-of-bag samples may be problem-dependent, although half-sampling has been widely used by default. In this work, we employ a probability-dependent sampling instead of using the standard half-sampling. To generate a data subset S_i, every data point in the offline data has a probability of 0.5 to be included in S_i. As a result, the size of S_i is not fixed as in half-sampling, which in principle can promote ensemble diversity.

After T datasets are generated, T RBF models are trained separately using the T datasets S_1, S_2,...,S_T. Each RBF model contains d neurons (Gaussian radial basis functions) in the hidden layer, where d is the number of the decision variables. The whole process of preparing the data subsets and training the pool of surrogate models are shown in Algorithm 1.

Algorithm 1 Pseudo code of setting up the surrogate ensemble in DDEA-SE.

Input: $D_{offline}$-the offline data, d-the dimension of $\mathbf{x}$, T-the size of the model pool.

```
for i = 1 : T do
    Set S_i empty.
    for each data points in D_offline do
        if U(0,1) < 0.5 then
            Add this point to S_i.
        end if
    end for
end for
for i = 1 : T do
    Train an RBF model M_i based on S_i.
end for
```

Output: the data subset pool (S_1, S_2,..., S_T) and model pool (M_1, M_2,..., M_T).

C. Model Management

As the experimental results in [57] show, the accuracy of bagging enhances as the ensemble size increases, when the ensemble size is smaller than 100. However, the accuracy does not necessarily continue to improve when the ensemble size further increases. This finding indicates that it is helpful to reduce the ensemble size without degrading the accuracy by using a model selection strategy [54].

Existing model selection strategies are guided by the global ensemble accuracy. We cannot simply adopt these strategies

as the surrogate management strategy in DDEA-SE, since the population in each generation is distributed in a local area. To address the issue, we propose two different strategies for selecting a subset of bagging models in each generation.

- **Fixed subset size selection strategy**: The number of selected models (base learners) is fixed in the whole optimization process.
- **Adaptive subset size selection strategy**: The number of selected models is adaptively changed according to the distribution of the population.

For the strategy selecting a fixed number of models in each generation, the model can be randomly or adaptively selected from the model pool. The surrogate ensemble can be seen as a global model when the base learners are selected randomly, while the surrogate ensemble becomes local when the base learners are selected considering a particular local region of interest in the search space. Here, we adaptively select a subset of bagging models as the population moves around in the search space. The main idea is to use the best individual (estimated by the surrogates) in the current generation as a reference for selecting the diverse models in the interesting regions for the next generation, which can be seen as a best strategy [5].

More specifically, the fixed subset size selection will be applied to select Q models from T models. Let $\mathbf{x_b}$ be the best individual according to the surrogate ensemble consisting of Q RBF models, then the fitness value according to the i-th ($1 \leq i \leq T$) individual RBF model is calculated, denoted by P_i. This is followed by sorting the T RBF models (denoted by M_1, M_2,...,M_T) according to the estimated fitness, denoted by P_1, P_2,...,P_T. Afterwards, the sorted RBF models are equally divided into Q groups. Finally, one RBF model is randomly selected from each of the Q groups to form a new surrogate ensemble to be used for fitness estimation in the next generation. This way, a set of diverse models local to the current population will be selected so that the locally most accurate fitness estimation can be achieved. The details of the fixed subset size selection strategy are presented in Algorithm 2.

Algorithm 2 Pseudo code of the fixed subset size selection strategy.

Input: Q: the ensemble size after model selection, $\mathbf{x_b}$: the current predicted best solution, M_1, M_2,..., M_T: the model pool.

1: **if** it is the first generation **then**
2: Randomly choose Q models from the pool.
3: **else**
4: Using (M_1, M_2,...,M_T) to predict $\mathbf{x_b}$.
5: Sort T RBF models based on their predictions on $\mathbf{x_b}$.
6: Equally divide T sorted RBF models into Q groups.
7: **for** each group **do**
8: One random model is selected to construct the ensemble.
9: **end for**
10: **end if**

Output: Q selected RBF models.

To elaborate Algorithm 2, we take a model pool having six models (M_1, M_2,..., M_6) as an example. The estimated fitness of $\mathbf{x_b}$ is $P_1 = 2.1$, $P_2 = 2.3$, $P_3 = 2.0$, $P_4 = 1.9$, $P_5 = 2.2$, and $P_6 = 2.4$, respectively. To select three models for the surrogate ensemble in the next generation, these models are sorted based on their estimated fitness value and clustered into three groups denoted by (M_3, M_4), (M_1, M_5), and (M_2, M_6). Then, one model is randomly chosen in each group. Thus, M_4, M_1, and M_6 can be one possible output of Algorithm 2.

The ensemble size Q is a parameter to be specified, affecting both the accuracy and computational complexity. To investigate the relationship between the ensemble size and fitness estimation accuracy, we examine the change of the root mean square error (RMSE) of the ensemble over the ensemble sizes up to 5000 on both uni- and multi-modal test problems (Ellipsoid and Rastrigin) with 10, 30, 50, and 100 decision variables. The experiments are conducted following the steps below:

- Generate 10000 random samples as the test dataset.
- Generate $11d$ solutions using the Latin hypercube sampling (LHS) [72] and calculate their fitness using the real objective function. These solutions are used as the offline training data.
- Build 5000 RBF models from the offline training dataset according to Algorithm 1.
- Calculate the RMSEs of the ensemble on the test dataset by sequentially adding RBF models to the ensemble.

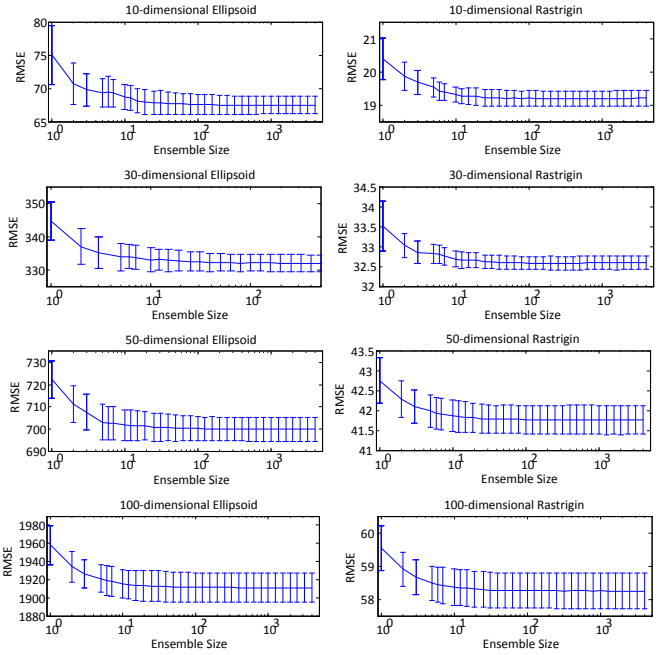

Fig. 4. Change of the average RMSE of the bagging ensembles over the ensemble size on the Ellipsoid and Rastrigin functions with up to 100 dimensions.

The RMSE averaged over 20 independent runs as the ensemble size increases is shown in Fig. 4, from which we note that the error profiles on both uni- and multi-modal problems with different numbers of decision variables are quite similar. It is straightforward that the computational cost of

the bagging ensemble linearly grows as the ensemble size increases. By contrast, the RMSE of the bagging ensemble decreases as the ensemble size increases in the beginning, but drops very slowly when the size is larger than 100. Based on this observation, we set Q to be 100 and T to be 2000 in DDEA-SE, hoping to achieve sufficiently good approximation accuracy with relatively low computational cost.

A fixed Q in Algorithm 2 might be unsuited for the whole optimization process, since the population tends to converge as the search proceeds. When the individuals are distributed in a small area, the surrogate ensemble should be able to capture local details of the fitness landscape by decreasing the number of selected models. We use an average distance (D_g) of the population in the g-th generation to the best individual $\mathbf{x_b}$ to measure the population distribution in the decision space. The subset size Q_g in the g-th generation is adjusted as below:

$$Q_g = \left\lfloor T\frac{D_g}{D_0} \right\rfloor, \tag{1}$$

where D_0 is the average distance of the initial population to its best individual. Thus, the smaller the local region in which the population are distributed, the smaller number of models will be selected in DDEA-SE.

Three different model selection strategies are designed for DDEA-SE. The first strategy randomly selects a fixed number of models, the second strategy selects a fixed number of models according to the location of the best solution, and the third strategy selects an adaptive number of models according to the population distribution and the location of the best solution. As the initial population of DDEA-SE is distributed across the whole search space, the surrogate ensemble is expected to be able to describe the global fitness landscape in the decision space. Therefore, Q models are randomly selected from T models for fitness estimation in the first generation. From the second generation onward, one of the three strategies presented will be applied to select models.

V. Experimental Results on Benchmark Problems

In this section, we will empirically analyze the performance of the proposed algorithm. The basic EA adopted in the proposed algorithm is a real-coded genetic algorithm with the simulated binary crossover (SBX) ($\eta = 15$), polynomial mutation ($\eta = 15$), and tournament selection. Further, the activation function of the RBF models is the Gaussian radial basis functions and there are d nodes (neurons) in the hidden layer, where d is the dimension of the decision space. The centers of the Gaussian functions of the RBF models are specified using the k-means clustering algorithm, the widths are set to be the maximum distance between the centers, and the weights from the hidden nodes to the output node are determined using the pseudo-inverse method [73].

In the experiments, we use five benchmark problems [34] of a dimension up to 100 decision variables, as presented in Table I. Here, we consider these benchmark problems to be computationally expensive and play the role of ground truth for examining the performance of the proposed offline data-driven EA. It should be emphasized that offline data-driven EAs cannot sample any new data during the optimization. Therefore, the real objective function will be used for performance assessment only and the data created for performance assessment are not available to the EA.

TABLE I
Test problems.

Problem	d	optimum	Characteristics
Ellipsoid	10,30,50,100	0.0	Uni-modal
Rosenbrock	10,30,50,100	0.0	Multi-modal
Ackley	10,30,50,100	0.0	Multi-modal
Griewank	10,30,50,100	0.0	Multi-modal
Rastrigin	10,30,50,100	0.0	Multi-modal

A. Empirical results

1) Comparison of Surrogate Management Strategies: In this work, three different model selection strategies are proposed as the surrogate management method in DDEA-SE. In this subsection, we examine the influence of the different surrogate management methods on the performance of DDEA-SE. Therefore, the following four DDEA-SE variants with or without these strategies are compared:

- DDEA-SE-random: the proposed algorithm randomly selecting Q models from T models in each generation ($T = 2000$ and $Q = 100$),
- DDEA-SE-fixed: the proposed algorithm selecting Q models from T models according to $\mathbf{x_b}$ in the current generation ($T = 2000$ and $Q = 100$),
- DDEA-SE-adaptive: the proposed algorithm selecting an adaptive number of models from T models according to $\mathbf{x_b}$ in the current generation ($T = 2000$),
- DDEA-E: the proposed algorithm without using any surrogate management method ($T = 2000$).

In the comparisons, the four compared algorithms all use a population size of 100 and terminate after running 100 generations. We test the four compared algorithms on $11d$ offline data of Ellipsoid and Rastrigin functions ($d = 10, 30, 50, 100$) using LHS. For each instance, each algorithm repeats for 20 times. The obtained optimal solution and runtime are presented in Table II. From the table, we can see that the DDEA-SE variants perform very similarly on the uni-modal Ellipsoid function. Also, the runtime of all algorithms increases as the dimension increases, however, the runtime of DDEA-E ($T = 2000$) grows much faster than other three algorithms, resulting in almost 10 times of runtime compared to that of DDEA-SE-fixed on the 100-dimensional test problems. We use the Friedman test with the Bergmann-Hommel post-hoc test [74] to analyze the results in Table II, and the p-values are shown in Table III. DDEA-SE-fixed significantly outperforms DDEA-SE-random and DDEA-SE-adaptive, but slightly outperforms DDEA-E ($T = 2000$). Comparing the p-values of DDEA-SE-fixed and DDEA-E ($T = 2000$) on running time, we find that DDEA-SE-fixed needs much shorter running time than DDEA-E ($T = 2000$).

From the above results, we observe that the model management strategy in DDEA-SE-fixed is able to significantly reduce the computation time without degrading the performance. DDEA-SE-random uses a global surrogate ensemble during the whole algorithm, which is the reason for its poor

TABLE II
RESULTS OBTAINED BY DDEA-SE VARIANTS ON ELLIPSOID AND RASTRIGIN PROBLEMS.

$11d$ LHS		Obtained optimum				Execution time (s)			
P	d	DDEA-SE-random	DDEA-SE-fixed	DDEA-SE-adaptive	DDEA-E	DDEA-SE-random	DDEA-SE-fixed	DDEA-SE-adaptive	DDEA-E
Ellipsoid	10	1.0±0.1	1.0±0.1	1.0±0.2	1.0±0.1	6.5±0.3	24.1±0.1	21.5±1.0	87.7±0.2
	30	3.9±0.4	4.2±0.6	4.9±0.9	3.9±0.3	20.9±0.4	42.2±0.1	75.0±2.9	291.6±1.2
	50	15.7±2.9	11.6±2.0	14.3±2.8	13.7±3.2	82.4±8.7	73.9±0.8	264.8±12.9	747.9±10.5
	100	328.7±63.7	317.2±74.4	323.7±76.7	319.4±80.7	279.5±13.3	214.8±4.7	1343.4±62.7	2035.1±10.4
Rastrigin	10	66.6±3.8	34.0±4.6	65.9±8.9	66.5±2.2	11.0±1.2	41.5±0.3	18.5±2.9	88.0±0.5
	30	178.0±7.8	116.8±7.2	183.3±12.6	180.6±5.5	49.6±3.7	70.1±3.6	73.2±6.9	294.4±6.7
	50	207.6±18.9	189.5±16.4	218.8±20.2	197.2±15.4	128.6±28.7	84.5±8.1	335.7±32.3	671.0±17.8
	100	840.2±78.3	833.8±70.2	858.1±69.5	838.2±65.0	324.5±85.4	282.0±110.7	1323.4±274.9	2291.7±37.5
Average rank		3.3	1.5	3.1	2.1	1.5	1.8	2.8	4.0

TABLE III
ADJUSTED p-VALUES OF THE FRIEDMAN TEST WITH THE BERGMANN-HOMMEL POST-HOC TEST (SIGNIFICANCE LEVEL=0.05) FOR THE COMPARISONS OF DDEA-SE VARIANTS. DDEA-SE-RANDOM, DDEA-SE-FIXED, AND DDEA-SE-ADAPTIVE ARE SHORTED TO RANDOM, FIXED, AND ADAPTIVE.

		Random	Fixed	Adaptive	DDEA-E
Optimum	Random	NA	**0.0067**	0.8465	0.0814
	Fixed	**0.0067**	NA	**0.0118**	0.3329
	Adaptive	0.8465	**0.0118**	NA	0.1213
	DDEA-E	0.0814	0.3329	0.1213	NA
Time	Random	NA	0.6985	0.0528	**0.0001**
	Fixed	0.6985	NA	0.1213	**0.0005**
	Adaptive	0.0528	0.1213	NA	0.0528
	DDEA-E	**0.0001**	**0.0005**	0.0528	NA

performance. DDEA-SE-adaptive was expected to perform better than DDEA-SE-fixed in Section IV-C, but its average rank is larger than that of DDEA-SE-fixed, indicating a worse performance. In fact, the objective function in DDEA-SE can be seen as a dynamic optimization problem, as the surrogate ensemble changes over the generations. However, no strategies handling the changing fitness landscape have been adopted in DDEA-SE. The severity of changes in DDEA-SE-adaptive is larger than that in DDEA-SE-fixed. In other words, DDEA-SE-adaptive deals with harder problems than DDEA-SE-fixed. Therefore, DDEA-SE-fixed performs better than DDEA-SE-adaptive.

To take a closer look at the behavior of the ensemble during the optimization, we show, in Figs. 5 and 6, respectively, the average percentage of the correctly selected individuals (meaning those should be selected when the fitness evaluations are based on the exact fitness function) before and after model selection in DDEA-SE-fixed on the two test problems. This percentage can be viewed as an assessment of the selection accuracy using the surrogates. By selecting Q RBF models, the selection accuracy for uni-modal Ellipsoid is slightly improved. In contrast, the selection accuracy on the multi-modal Rastrigin function has been significantly enhanced at the later stage of the search. These results indicate that the selective ensemble is able to distinguish better solutions from worse ones in the exploitation stage.

2) Comparison of Ensemble Generation Strategies: From the results in Section V-A1, DDEA-SE-fixed is the best-performing variant. We use the strategy to select a fixed number models according to $\mathbf{x_b}$ in DDEA-SE for the following experiments.

The ensemble generation strategy is an important step of DDEA-SE, where every data point in the offline data has

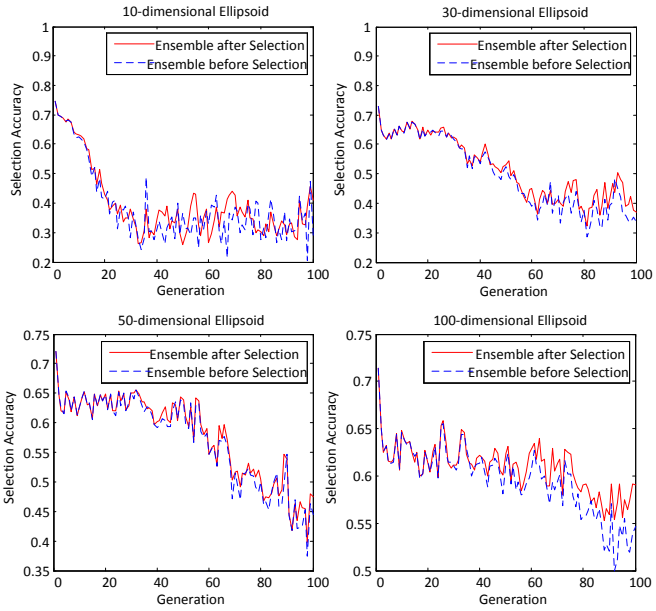

Fig. 5. Average selection accuracy before and after model selection in DDEA-SE-fixed on Ellipsoid problems with different numbers of decision variables.

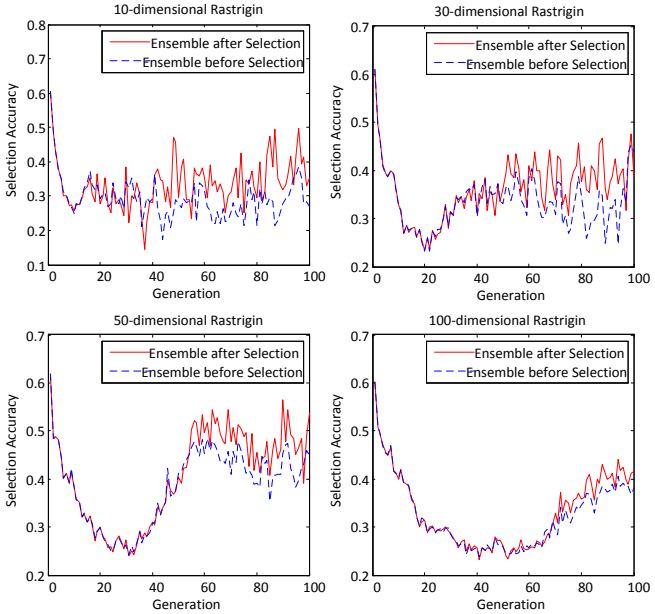

Fig. 6. Average selection accuracy before and after model selection in DDEA-SE-fixed on Rastrigin problems with different numbers of decision variables.

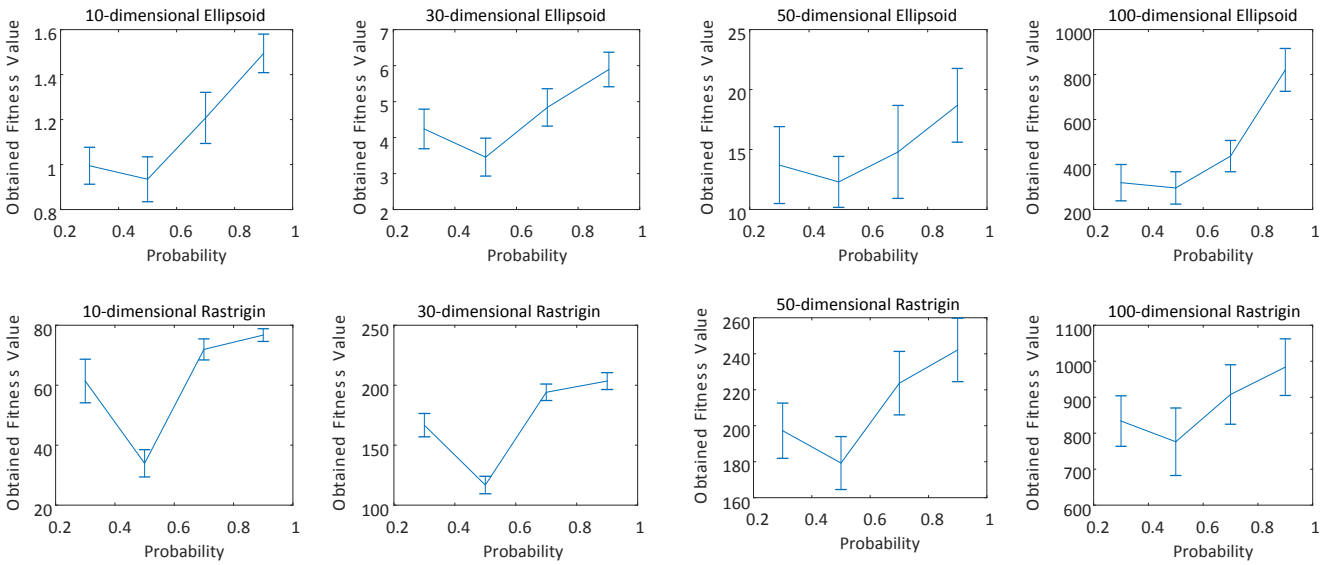

Fig. 7. Optimum obtained by DDEA-SE with the ensemble generation strategies of different probabilities (0.3, 0.5, 0.7, 0.9) on Ellipsoid and Rastrigin problems with different numbers of decision variables.

a probability to be included in the dataset for generating a model. Different probabilities lead to different ensemble generation strategies. In this subsection, we examine the effects of those different probabilities (0.3, 0.5, 0.7, and 0.9) in the ensemble generation strategy on DDEA-SE.

In the comparisons, the four DDEA-SE variants (using different probabilities) all use a population size of 100 and terminate after running 100 generations. We test those four compared algorithms on $11d$ offline data of Ellipsoid and Rastrigin functions ($d = 10, 30, 50, 100$) using LHS. For each offline data, each algorithm repeats for 20 times. The optimal solutions obtained using different probabilities are shown in Fig. 7. For both multi- and uni-modal problems, DDEA-SE has shown the best performance when the probability is 0.5. The reason is that a probability of 0.5 can offer the most diverse data subsets, leading to the generation of the most diverse models. These results indicate that the diversity of the generated models of the ensemble heavily influence the optimization performance. In the following experiments, we set the probability of the ensemble generation strategy to be 0.5 in DDEA-SE.

3) Comparison of Offline Data-Driven EAs: In this subsection, we compare the proposed algorithm with two offline data-driven EAs:

- DDEA-SE: DDEA-SE-fixed with the settings of $Q = 100$ and $T = 2000$,
- DDEA-E: the proposed algorithm without the surrogate management strategy ($T = 100$),
- DDEA-RBF: an EA using a single RBF model as the surrogate built from all offline data.

It has been shown that DDEA-E with 2000 RBF models is computationally very intensive. One natural question is what if we generate a smaller ensemble in the beginning. To answer this question, we compare here a variant of DDEA-E that generates 100 RBF models offline and no model selection is carried out during the optimization.

In the comparisons, the three compared algorithms all use a population size of 100 and terminate after running 100 generations. Unlike online data-driven EAs, offline data-driven EAs are tested on different offline datasets. Therefore, we use two sampling methods (LHS and random sampling) to generate offline data. We test the compared algorithms on three different types of offline data for each test problem: datasets with $11d$ and $5d$ solutions generated by LHS and a dataset with $11d$ solutions generated by random sampling. To avoid possible biases from different datasets, each dataset is generated independently for three times as three instances. For each instance, each algorithm repeats for 20 times.

The comparative results of DDEA-SE, DDEA-E, and DDEA-RBF on the Ellipsoid and Rastrigin test problems are shown in Table IV. From these results, we can see that DDEA-SE and DDEA-E outperform DDEA-RBF on most instances compared in this study. Then we apply the Friedman test with the Bergmann-Hommel post-hoc test (significance level=0.05) [74] to compare these results, where DDEA-SE is the control method. Overall, DDEA-SE performs the best, followed by DDEA-E, and DDEA-RBF performs the worst. In other words, surrogate ensemble improves the performance of the offline data-driven EAs and selective surrogate ensemble can further enhance the performance. However, the compared algorithms behave slightly differently on different datasets. For example, data generated by LHS can result in better performance than randomly generated data, as evidenced by the results of DDEA-RBF using a single surrogate. In addition, DDEA-SE improves its performance when the data size increases from $5d$ to $11d$ and the performance enhancement by using ensemble surrogates becomes more significant as the dimension increases.

To study the scalability of the proposed algorithm, we investigate its performance on 10-, 30-, 50-, and 100-dimensional

TABLE IV

OPTIMAL SOLUTIONS OBTAINED BY DDEA-SE, DDEA-E AND DDEA-RBF, WHERE I# MEANS THE INSTANCE NUMBER OF OFFLINE DATA. THE RESULTS ARE SHOWN IN THE FORM OF MEAN ± STANDARD DEVIATION. THE RESULTS ARE ANALYZED BY THE FRIEDMAN TEST WITH THE BERGMANN-HOMMEL POST-HOC TEST (DDEA-SE IS THE CONTROL METHOD AND THE SIGNIFICANCE LEVEL IS 0.05). THE BEST FITNESS VALUES AMONG ALL THE COMPARED ALGORITHMS FOR EACH PROBLEM ARE HIGHLIGHTED.

Offline Data			$11d$ LHS			$11d$ Rand			$5d$ LHS		
P	d	I#	DDEA-SE	DDEA-E	DDEA-RBF	DDEA-SE	DDEA-E	DDEA-RBF	DDEA-SE	DDEA-E	DDEA-RBF
Ellipsoid	10	1	**1.0±0.1**	1.7±0.7	3.2±2.0	**3.9±0.3**	4.6±1.0	6.0±3.7	**2.6±0.2**	4.2±1.6	7.1±3.0
		2	**0.6±0.1**	1.2±0.6	2.7±1.6	**3.7±0.2**	3.7±1.0	5.6±2.7	**1.1±0.1**	2.8±1.5	5.7±4.9
		3	**1.5±0.1**	2.0±0.9	5.6±2.7	**3.0±0.2**	3.7±1.1	5.0±1.4	**1.2±0.2**	2.3±0.9	4.6±3.2
	30	1	**4.2±0.6**	5.4±1.1	15.8±5.5	**17.4±1.5**	20.3±3.0	33.5±11.5	**5.7±0.5**	9.9±2.6	28.6±14.0
		2	**2.8±0.2**	5.5±1.6	12.4±4.1	**14.2±0.8**	16.4±2.3	28.4±14.6	**9.7±0.8**	14.5±3.1	36.1±15.8
		3	**4.3±0.4**	7.0±1.7	16.0±4.9	**6.3±0.8**	10.1±2.2	20.0±10.5	**7.3±0.8**	12.2±2.0	23.6±8.5
	50	1	**11.6±2.0**	18.5±3.5	54.2±22.9	**25.6±3.9**	33.8±6.8	89.2±36.2	**17.3±2.7**	25.1±5.8	67.9±38.6
		2	**14.3±2.7**	20.4±2.9	52.1±20.7	**30.7±3.9**	37.4±6.4	89.8±45.4	**27.2±3.1**	37.7±6.8	91.9±36.0
		3	**12.1±2.3**	18.6±4.2	65.4±23.4	**28.2±3.7**	34.4±6.1	81.8±29.7	**19.6±3.6**	30.1±6.4	65.9±20.6
	100	1	**317.2±74.4**	371.0±89.2	2186.0±1665.8	**331.6±43.8**	364.2±63.8	1823.1±1235.8	**321.6±53.8**	339.8±67.4	757.6±443.7
		2	**330.8±48.8**	489.0±189.5	2593.2±897.5	**327.7±66.5**	384.6±98.6	2143.5±1123.6	**293.5±65.2**	312.2±73.7	746.5±425.9
		3	**294.9±36.3**	364.0±66.5	1245.5±776.5	**306.7±41.8**	380.6±75.4	1531.4±851.5	**321.8±72.2**	353.2±41.6	597.9±151.0
Rastrigin	10	1	**34.0±4.6**	76.6±11.7	80.1±21.5	**47.3±3.1**	58.9±14.6	78.9±18.5	131.0±5.9	**105.0±24.7**	102.4±20.8
		2	**52.4±4.6**	93.3±16.7	76.7±33.6	**69.9±3.7**	75.4±17.3	101.9±27.8	**82.0±3.9**	96.9±17.1	93.6±27.6
		3	**57.1±1.8**	109.2±14.0	90.8±26.2	**66.8±3.7**	79.5±22.5	92.7±18.5	**76.1±5.1**	79.6±18.1	102.9±21.4
	30	1	**116.8±7.2**	208.3±29.2	286.8±39.2	**214.6±9.6**	233.9±14.4	295.0±31.9	**162.8±8.7**	207.9±35.6	290.4±31.6
		2	**90.5±4.5**	122.4±22.3	191.5±37.8	**179.5±7.3**	195.5±21.4	252.2±23.6	**162.5±11.4**	213.4±36.4	268.4±54.9
		3	**100.8±5.0**	134.5±22.4	238.7±44.2	**162.2±6.1**	190.4±30.2	255.7±47.9	**240.7±12.3**	248.8±33.3	295.4±34.9
	50	1	**189.5±16.4**	233.3±41.0	408.4±74.2	**209.2±18.4**	246.7±35.2	424.2±57.0	**238.6±17.7**	298.1±34.6	422.1±62.1
		2	**158.6±16.0**	233.7±32.8	421.4±41.0	**280.5±26.6**	328.2±35.5	480.5±69.2	**287.0±27.0**	333.4±36.3	470.5±51.9
		3	**180.0±18.1**	263.9±40.8	441.0±48.2	**180.4±18.1**	236.7±33.4	425.5±67.7	**232.1±15.6**	301.4±40.3	426.7±45.5
	100	1	**833.8±70.2**	891.8±103.3	1053.3±57.3	**825.1±88.3**	920.5±96.1	1013.0±71.3	**800.1±65.8**	903.2±64.8	1042.5±64.0
		2	**848.3±82.7**	949.4±75.2	1068.7±96.8	**868.9±73.7**	935.8±88.2	1070.0±73.7	**794.7±110.7**	837.4±66.5	1013.3±82.9
		3	**762.2±99.7**	860.6±94.9	1003.1±74.0	**832.0±67.8**	883.2±75.2	1046.5±69.4	**828.2±82.8**	874.9±48.7	1015.6±86.1
Average rank			1.0	2.1	2.9	1.0	2.0	3.0	1.1	2.0	2.9
Adjusted p-value			NA	**0.0002**	**0.0000**	NA	**0.0005**	**0.0000**	NA	**0.0009**	**0.0000**

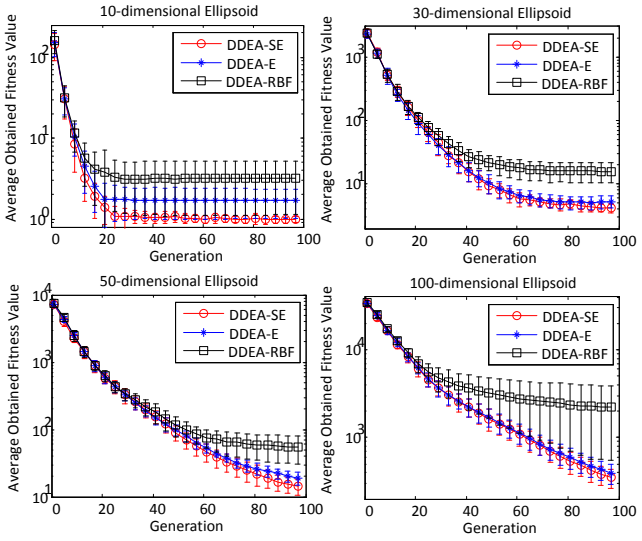

Fig. 8. Average convergence profiles of DDEA-SE, DDEA-E and DDEA-RBF on Ellipsoid test problems with different numbers of decision variables.

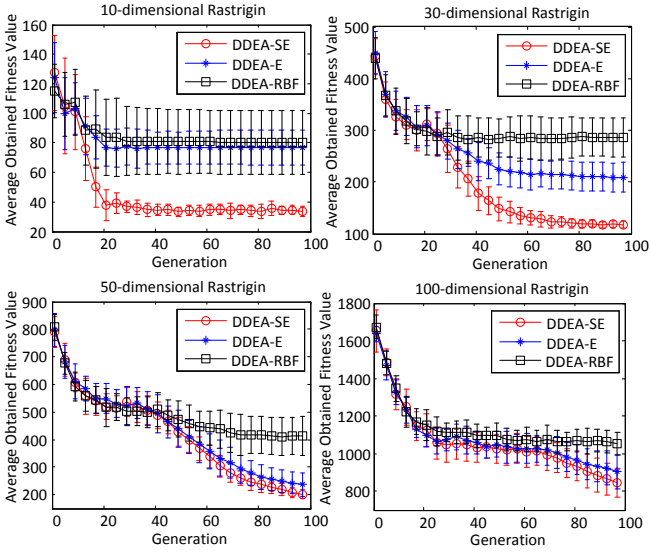

Fig. 9. Average convergence profiles of DDEA-SE, DDEA-E and DDEA-RBF on Rastrigin test problems with different numbers of decision variables.

Ellipsoid and Rastrigin problems when $11d$ data are sampled using LHS. The results are presented in Figs. 8 and 9, respectively. Note that in Figs. 8 and 9, the best solution in each generation is evaluated using the real objective function. From these results, we can see that DDEA-SE outperforms DDEA-RBF and DDEA-E on the 10-dimensional Ellipsoid problem. However, the performance of DDEA-SE becomes less advantageous as the number of decision variables increases. By contrast, DDEA-SE outperforms both DDEA-RBF and DDEA-E on the 10-dimensional Rastrigin problem. On the 30-dimensional Rastrigin problem, both DDEA-E and DDEA-SE significantly outperform DDEA-RBF, where the advantage of ensemble becomes more obvious. Both DDEA-E and DDEA-SE performs comparably well but much better than DDEA-RBF on the 50- and 100-dimensional Rastrigin problems.

From the above results, we can make the following observations. First, surrogate ensembles help improve the performance of data-driven EAs in general compared with a single surrogate. Second, selective ensembles are able to further enhance the performance of offline data-driven EAs while significantly

reducing the computation time. Finally, EAs assisted by a selective ensemble are likely to perform much better on multi-modal problems than EAs assisted by a non-selective ensemble or a single surrogate.

4) Scalability on Size of Offline Data: In this subsection, we compare DDEA-SE, DDEA-E and DDEA-RBF on the Rastrigin problem for different data sizes (100, 300, 500, 700, and 1000) generated using LHS. In the experiment, all the compared algorithms repeat for 20 independent times. Note, however, that for 50- and 100-dimensional problems, at least 300 data samples are considered.

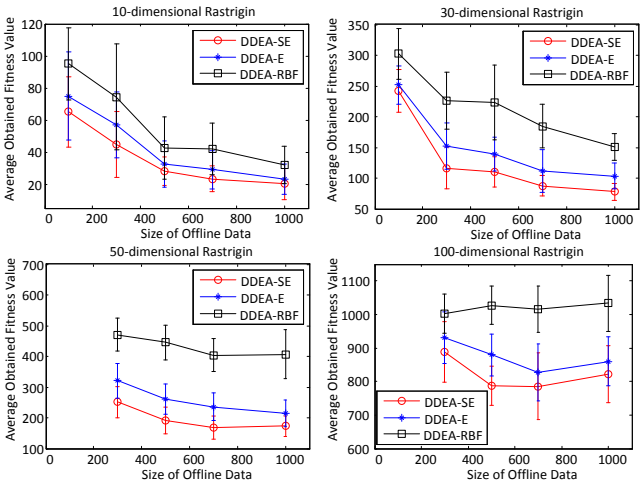

Fig. 10. Average fitness obtained by DDEA-SE, DDEA-E, and DDEA-RBF on the Rastrigin problems with different offline data sizes.

The average fitness obtained by DDEA-SE, DDEA-E and DDEA-RBF on the Rastrigin problems with different data sizes are plotted in Fig. 10. For 10-, 30- and 50-dimensional Rastrigen problems, the performance of all compared algorithms consistently enhances as the data size increases, although it is noticed the performance improvement becomes less significant when the size is larger than 700. Note also that all algorithms perform even worse on the 100-dimensional Rastrigin problem, when the size of data is increased to 1000. The performance degradation can be also observed from Table IV, which might be attributed to the fact that the Rastrigin function is a multi-modal function and a large data size may enable the surrogates to capture more local optima, leading to worse search performance.

B. Comparison with Online Data-Driven EAs

To further examine the performance of the proposed offline data-driven EA, we compare DDEA-SE with a few online data-driven EAs on the test problems listed in Table I, although such comparisons may not be completely fair. Since DDEA-SE is ensemble-based, we choose one ensemble-assisted and two single surrogate-assisted data-driven EAs as compared algorithms: committee-based active learning for surrogate-assisted particle swarm optimization algorithm (CAL-SAPSO) [18], Gaussian process surrogate model assisted evolutionary algorithm for medium-scale expensive problems (GPEME) [34], and surrogate-assisted cooperative swarm optimization algorithm (SA-COSO) [28]. The characteristics of these three algorithms are briefly discussed below.

- CAL-SAPSO is an online ensemble-assisted data-driven EA assisted by multiple surrogates, namely, PR, RBF, and Kriging models, using an active learning-based surrogate management strategy.
- GPEME is an online single surrogate-assisted data-driven EA assisted by a Kriging model with the LCB-based infill sampling criterion as its surrogate management strategy.
- SA-COSO is an online data-driven EA assisted by a single RBF model with two swarms in its surrogate management strategy.

TABLE V
OPTIMAL SOLUTIONS OBTAINED BY DDEA-SE, CAL-SAPSO AND GPEME, WHERE THE RESULTS ARE ANALYZED BY THE FRIEDMAN TEST WITH THE BERGMANN-HOMMEL POST-HOC TEST (DDEA-SE IS THE CONTROL METHOD AND THE SIGNIFICANCE LEVEL IS 0.05). THE BEST FITNESS VALUES AMONG THE COMPARED ALGORITHMS FOR EACH PROBLEM ARE HIGHLIGHTED.

Problem	d	DDEA-SE	CAL-SAPSO online	CAL-SAPSO offline	GPEME online	GPEME offline
Ellipsoid	10	1.0±0.5	0.9±0.9	**0.0±0.0**	37.8±15.3	129.8±34.3
	30	5.0±1.5	**4.0±1.1**	30.2±10.8	1228.6±223.6	2013.3±246.1
Rosenbrock	10	29.1±6.3	**16.0±3.4**	157.2±120.4	186.0±66.9	625.6±196.7
	30	53.5±4.5	**51.0±11.5**	184.5±28.1	2441.0±809.5	4998.6±646.9
Ackley	10	**6.3±1.3**	20.1±0.2	18.2±0.8	13.8±2.5	19.2±0.5
	30	**4.8±0.5**	16.2±0.4	12.6±2.3	19.5±0.4	20.4±0.1
Griewank	10	1.3±0.1	1.1±0.1	**0.0±0.0**	27.2±11.3	103.6±31.4
	30	1.3±0.1	**1.0±0.0**	2.6±0.8	283.6±52.5	488.6±29.9
Rastrigin	10	1.0±0.5	0.9±0.9	**0.0±0.0**	37.8±15.3	129.8±34.3
	30	5.0±1.5	**4.0±1.1**	30.2±10.8	1228.6±223.6	2013.3±246.1
Average rank		1.9	2.0	2.8	3.5	4.8
Adjusted p-value		NA	0.8875	0.2031	**0.0237**	**0.0000**

Since CAL-SAPSO and GPEME were not meant for high-dimensional problems, we compare them with DDEA-SE only on 10- and 30-dimensional problems. For the problems with 50 and 100 decision variables, we compare DDEA-SE with SA-COSO. The parameter settings (including the hyperparameter optimization for their surrogate models) of CAL-SAPSO and GPEME are exactly the same as in [18], those for SA-COSO are taken from [28]. In addition to the original versions of those three online SAEAs, we compare their offline versions. The offline versions of CAL-SAPSO, GPEME, and SA-COSO start with training the surrogate using all allowed computational budget and stop once the first real fitness evaluation is required. This means that the surrogate models in the offline versions are better than those in the original online versions before the optimization starts. In this section, all the compared algorithms repeats 20 times. $11d$ real fitness evaluations are allowed for all compared algorithms.

The results of DDEA-SE, CAL-SAPSO and GPEME on the 10- and 30-dimensional problems are given in Table V. The results are analyzed by the Friedman test with the Bergmann-Hommel post-hoc test (significance level=0.05) [74], where DDEA-SE is the control method. From the Friedman test, we can see that DDEA-SE significantly performs better than GPEME. DDEA-SE is the best-performing algorithm on two test problems, CAL-SAPSO (online) is the best-performing algorithm on five test problems, and CAL-SAPSO (offline) is the best-performing algorithm on three 10-dimensional problems.

For 10-dimensional problems, $11d$ samples are sufficient to train a single well-performing surrogate model, which is the reason why CAL-SAPSO (offline) has the best performance. However, for 30-dimensional problems, $11d$ samples become insufficient for training surrogate models, thus the performance of CAL-SAPSO (offline) dramatically degenerates. With the help of active sampling or ensemble surrogate, CAL-SAPSO (online) and DDEA-SE outperform CAL-SAPSO (offline) on 30-dimensional problems. Note that CAL-SAPSO (online) actively samples part of the data during the optimization, while DDEA-SE collects all samples offline before the optimization starts. Nevertheless, DDEA-SE can still achieve relatively good performance on low-dimensional problems.

TABLE VI
OPTIMAL SOLUTIONS OBTAINED BY DDEA-SE AND SA-COSO, WHERE THE RESULTS ARE ANALYZED BY THE FRIEDMAN TEST WITH THE BERGMANN-HOMMEL POST-HOC TEST (DDEA-SE IS THE CONTROL METHOD AND THE SIGNIFICANCE LEVEL IS 0.05). THE BEST FITNESS VALUES AMONG ALL THE COMPARED ALGORITHMS FOR EACH PROBLEM ARE HIGHLIGHTED IN BOLDFACE.

Problem	d	DDEA-SE	SA-COSO online	SA-COSO offline
Ellipsoid	50	**15.4±3.8**	226.8±66.4	179.8±44.5
	100	**312.2±59.1**	957.9±236.4	931.2±219.4
Rosenbrock	50	**84.0±6.3**	615.9±216.1	565.1±112.9
	100	**250.6±37.4**	2078.9±447.8	2035.8±649.8
Ackley	50	**4.6±0.3**	13.0±0.9	13.1±0.9
	100	**7.0±0.5**	15.9±0.6	15.5±0.5
Griewank	50	**1.9±0.2**	27.2±5.6	24.7±5.5
	100	**17.3±3.0**	74.2±16.5	57.8±15.7
Rastrigin	50	**181.8±32.0**	417.7±34.2	422.2±39.1
	100	**809.8±102.2**	821.6±69.0	857.1±67.1
Average rank		1.0	2.7	2.3
Adjusted p-value		NA	**0.0001**	**0.0037**

The results obtained by DDEA-SE and SA-COSO on the 50- and 100-dimensional problems are shown in Table VI. The results are analyzed by the Friedman test with the Bergmann-Hommel post-hoc test (significance level=0.05) [74], where DDEA-SE is the control method. Surprisingly, DDEA-SE significantly outperforms both SA-COSOs. Although the model in offline SA-COSO is better than that in online SA-COSO, the improvement of offline SA-COSO is not significant. This indicates that the use of ensemble surrogates is more reliable than single surrogate, in particular for high-dimensional problems.

From the above experimental results, we can conclude that the performance of DDEA-SE is comparable with two online data-driven EAs on low-dimensional problems and is better than one online data-driven EAs on high-dimensional problems, demonstrating that DDEA-SE is able to perform robustly on different problems, even if compared with online data-driven SAEAs.

VI. APPLICATION TO AIRFOIL DESIGN

In this section, we apply the proposed algorithm to the RAE2822 airfoil test case in the GARTEUR (Group for Aeronautical Research and Technology in Europe) AG52 project [75], where nine European collaborative partners aim to promote research on surrogate-based aerodynamic shape optimization[1]. For the RAE2822 airfoil test case, 70 different geometries in the defined parameterization were given as a starting point. The partners then tried to find an optimal geometry by using their own optimization methods, together with computational fluid dynamic (CFD) simulations to provide quality evaluation for candidate geometries. Then, the optimal candidates found by all the partners were compared and cross validated using the other partner's CFD simulations. Advantages and disadvantages of different optimization methods, surrogate models, model management strategies, and CFD simulators are assessed.

However, CFD simulations are computationally very expensive and directly integrating an EA with a CFD tool is not always straightforward (sometimes the end user is hesitant to give out the code too). In this study, we run CFD simulations (VGK) [76], [77] for the 70 geometries as used in GARTEUR, as the data to verify the performance of offline data-driven EAs.

A. Problem Description

As described in [18], the airfoil design problem has 14 decision variables, which define the geometry of a candidate airfoil design. The objective is to minimize the drag over lift ratio, which is calculated from CFD simulations. The detailed objective functions are described as follows:

$$f_{Airfoil} = \min \frac{1}{2}\left(\frac{D_1}{L_1}\Big/\frac{D_1^b}{L_1^b} + \frac{D_2}{L_2}\Big/\frac{D_2^b}{L_2^b}\right), \qquad (2)$$

where two design conditions are considered, D_i and L_i are the drag and lift coefficients in design condition i, D_i^b and L_i^b are the drag and lift coefficients of baseline design in design condition i. Each drag or lift coefficients need to be calculated using CFD simulations. The fitness of the baseline design is normalized to be 1.

B. Results

In the comparisons, we run DDEA-SE, DDEA-E, and DDEA-RBF for 20 independent times. Those three compared algorithms are set as Section V-A3. As the airfoil design optimization problem has been tested on online data-driven EAs in [18], we use those reported results of CAL-SAPSO and GPEME as a reference. Noted that, these online data-driven EAs use the same offline data as the compared algorithms but 84 more online data. To verify the performance of the compared algorithms, all obtained designs are verified using CFD simulations, which are shown in Table VII and the best geometries (in X-Z coordinates) obtained by the compared algorithms are shown in Fig. 11.

TABLE VII
EXACT FITNESS VALUES OBTAINED BY DDEA-SE, DDEA-E, AND DDEA-RBF ON THE RAE2822 AIRFOIL TEST CASE. THE BEST RESULTS ARE HIGHLIGHTED.

Offline algorithm	DDEA-SE	**0.8470±0.0079**
	DDEA-E	0.9473±0.0358
	DDEA-RBF	3.4194±10.4958
Online algorithm	CAL-SAPSO	0.6843±0.0108
	GPEME	0.7781±0.0100

[1]http://www.garteur.org/

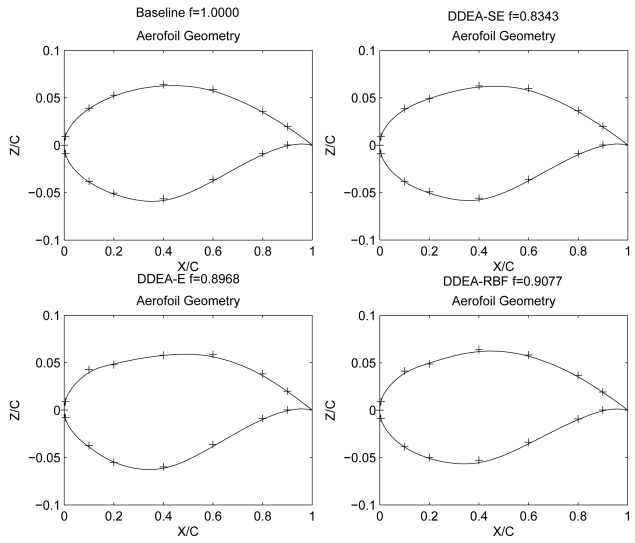

Fig. 11. The baseline design and the best designs obtained by DDEA-SE, DDEA-E and DDEA-RBF.

From Table VII, we can see that the best designs obtained by DDEA-SE, DDEA-E and DDEA-RBF are all better than the baseline design and the one obtained by DDEA-SE is the best. In addition, DDEA-SE has achieved the best average fitness and the minimum variance. This confirms that DDEA-SE performs robustly on this airfoil design optimization problem. It should be noted that the average fitness obtained by DDEA-RBF is much worse than the baseline design mainly because the CFD simulation for one out of the 20 runs has failed, resulting in an abnormally large value for the objective function. Note also that the above results are worse than those reported in [18], which might be due to the fact that all compared algorithms in [18] are online data-driven EAs using 70 offline data and 84 online data.

VII. Concluding Remarks

This paper aims to address offline data-driven optimization problems, which are challenging, widely seen in the real-world, but are largely neglected in the evolutionary optimization community. For offline data-driven optimization problems where only limited data is available, the optimization becomes extremely difficult and it becomes critical to fully exploit the data to guide the search. In this work, we propose a data-driven EA using an adaptive selective ensemble. The proposed algorithm builds a large number of surrogate models on the basis of probability-based sampling of the given data before the optimization starts and adaptively selects a small subset of the models built offline. The experimental results on five benchmark problems demonstrate that the proposed algorithm can deal with various problems with up to 100 decision variables, no matter whether the data are created randomly or sampled using the Latin hypercube method. Additionally, the effectiveness of the proposed algorithm is verified on the RAE2822 airfoil test case.

Despite of the promising results, we must emphasize that the work reported in this paper is still a first step towards solving offline data-driven optimization problems. Several possible improvements could be considered in the future. First, the surrogate management strategy plays an important role in data-driven EAs. More sophisticated surrogate management strategies that more explicitly take into account of the local and global fitness landscapes need to be designed. Second, advanced machine learning techniques such as stacking [78], transfer learning [51], and deep learning should be explored. Fusion of heterogeneous data might be needed for solving more complex real-world problems. Last but not least, the proposed algorithm can be used to deal with the offline part of online data-driven EAs, and the performance improvement should be further studied.

References

[1] D. Dasgupta and Z. Michalewicz, *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, 2013.
[2] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," *Control Engineering Practice*, vol. 10, no. 11, pp. 1223–1241, 2002.
[3] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 62–76, 2009.
[4] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.
[5] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
[6] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 939–952, 2016.
[7] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.
[8] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.
[9] S. Zapotecas Martínez and C. A. Coello Coello, "MOEA/D assisted by RBF networks for expensive multi-objective optimization problems," in *Proceeding of the 15th Annual Conference on Genetic and Evolutionary Computation Conference*. ACM, 2013, pp. 1405–1412.
[10] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2018.
[11] N. Namura, K. Shimoyama, and S. Obayashi, "Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 898–913, 2017.
[12] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou, "Metamodel–assisted evolution strategies," in *Parallel Problem Solving from Nature–PPSN VII*. Springer, 2002, pp. 361–370.
[13] Y. Liu and M. Collette, "Improving surrogate-assisted variable fidelity multi-objective optimization using a clustering algorithm," *Applied Soft Computing*, vol. 24, pp. 482–493, 2014.
[14] 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 *IEEE Congress on Evolutionary Computation*, vol. 3. IEEE, 2005, pp. 2832–2839.
[15] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Proceedings of the 6th Annual Conference on Genetic and Evolutionary Computation*. Springer, 2004, pp. 688–699.
[16] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, "Ensemble of surrogates," *Structural and Multidisciplinary Optimization*, vol. 33, no. 3, pp. 199–216, 2007.

[17] F. A. Viana, V. Picheny, and R. T. Haftka, "Conservative prediction via safety margin: design through cross-validation and benefits of multiple surrogates," in *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2009, pp. 741–750.

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

[19] K. S. Bhattacharjee, H. K. Singh, T. Ray, and J. Branke, "Multiple surrogate assisted multiobjective optimization using improved pre-selection," in *IEEE Congress on Evolutionary Computation*. IEEE, 2016, pp. 4328–4335.

[20] D. Villanueva, R. T. Haftka, R. Le Riche, and G. Picard, "Locating multiple candidate designs with dynamic local surrogates," in *10th World Congress on Structural and Multidisciplinary Optimization (WCSMO-10)*, 2013.

[21] R. Allmendinger, M. T. M. Emmerich, J. Hakanen, Y. Jin, and E. Rigoni, "Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case," *Journal of Multi-Criteria Decision Analysis*, vol. 14, pp. 5–25, 2017.

[22] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[23] P. B. Nair, A. J. Keane, and R. Shimpi, "Combining approximation concepts with genetic algorithm-based structural optimization procedures," in *Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 1998, pp. 1741–1751.

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

[25] J. Branke and C. Schmidt, "Faster convergence by means of fitness estimation," *Soft Computing*, vol. 9, no. 1, pp. 13–20, 2005.

[26] M. Binois, D. Ginsbourger, and O. Roustant, "Quantifying uncertainty on pareto fronts with gaussian process conditional simulations," *European Journal of Operational Research*, vol. 243, no. 2, pp. 386–394, 2015.

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

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

[29] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 1, pp. 66–76, 2007.

[30] W. Ponweiser, T. Wagner, and M. Vincze, "Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models," in *IEEE Congress on Evolutionary Computation*. IEEE, 2008, pp. 3515–3522.

[31] I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, and L. Knockaert, "Surrogate-based infill optimization applied to electromagnetic problems," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 20, no. 5, pp. 492–501, 2010.

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

[33] D. Büche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 2, pp. 183–194, 2005.

[34] B. Liu, Q. Zhang, and G. G. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.

[35] A. A. Rahat, R. M. Everson, and J. E. Fieldsend, "Alternative infill strategies for expensive multi-objective optimisation," in *Proceeding of the 19th Annual Conference on Genetic and Evolutionary Computation Conference*. ACM, 2017, pp. 873–880.

[36] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble based infill criterion for evolutionary multi-objective optimization of expensive problems," *IEEE Transactions on Cybernetics*, 2018, accepted.

[37] 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," *Materials and Manufacturing Processes*, vol. 32, no. 10, pp. 1172–1178, 2017.

[38] D. Guo, T. Chai, J. Ding, and Y. Jin, "Small data driven evolutionary multi-objective optimization of fused magnesium furnaces," in *IEEE Symposium Series on Computational Intelligence*. Athens, Greece: IEEE, December 2016.

[39] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[40] J. H. Friedman and P. Hall, "On bagging and nonlinear estimation," *Journal of Statistical Planning and Inference*, vol. 137, no. 3, pp. 669–683, 2007.

[41] S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," *IEEE Transactions on Reliability*, vol. 62, no. 2, pp. 434–443, 2013.

[42] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1356–1368, 2015.

[43] A. Fernandez, S. del Rio, N. V. Chawla, and F. Herrera, "An insight into imbalanced big data classification: outcomes and challenges," *Complex & Intelligent Systems*, vol. 3, no. 2, pp. 105–120, 2017.

[44] H. Wang, Q. Zhang, L. Jiao, and X. Yao, "Regularity model for noisy multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 9, pp. 1997–2009, 2016.

[45] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 459–472, 2006.

[46] S. Castano and V. De Antonellis, "Global viewing of heterogeneous data sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 2, pp. 277–297, 2001.

[47] A. Gupta, Y. S. Ong, and L. Feng, "Multifactorial evolution: toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2016.

[48] A. Gupta, J. Mańdziuk, and Y.-S. Ong, "Evolutionary multitasking in bi-level optimization," *Complex & Intelligent Systems*, vol. 1, no. 1-4, pp. 83–95, 2015.

[49] J. Luo, A. Gupta, Y. S. Ong, and Z. Wang, "Evolutionary optimization of expensive multiobjective problems with co-sub-pareto front gaussian process surrogates," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–14, 2018, doi=10.1109/TCYB.2018.2811761.

[50] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[51] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized multi-tasking for evolutionary optimization of expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–1, 2017, doi: 10.1109/TEVC.2017.2785351.

[52] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. The MIT Press, 2006.

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

[54] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression: A survey," *ACM Computing Surveys*, vol. 45, no. 1, p. 10, 2012.

[55] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Genetic and Evolutionary Computation Conference*. Springer, 2004, pp. 688–699.

[56] N. García-Pedrajas, C. Hervás-Martínez, and D. Ortiz-Boyer, "Cooperative coevolution of artificial neural network ensembles for pattern classification," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 271–302, 2005.

[57] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[58] Y. L. Suen, P. Melville, and R. J. Mooney, "Combining bias and variance reduction techniques for regression trees," in *ECML*. Springer, 2005, pp. 741–749.

[59] D. Hernández-Lobato, G. Martínez-Muñoz, and A. Suárez, "Pruning in ordered regression bagging ensembles," in *International Joint Conference on Neural Networks, 2006. IJCNN'06.* IEEE, 2006, pp. 1266–1273.

[60] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.

[61] L. Breiman, "Out-of-bag estimation," 1996.

[62] P. Büchlmann and B. Yu, "Analyzing bagging," *Annals of Statistics*, vol. 30, no. 4, pp. 927–961, 2002.

[63] A. Buja and W. Stuetzle, "Observations on bagging," *Statistica Sinica*, pp. 323–351, 2006.

[64] G. Martínez-Muñoz and A. Suárez, "Out-of-bag estimation of the optimal sample size in bagging," *Pattern Recognition*, vol. 43, no. 1, pp. 143–152, 2010.
[65] N. Li, Y. Yu, and Z.-H. Zhou, "Diversity regularized ensemble pruning," *Machine Learning and Knowledge Discovery in Databases*, pp. 330–345, 2012.
[66] H. Chen, P. Tiňo, and X. Yao, "Predictive ensemble pruning by expectation propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 7, pp. 999–1013, 2009.
[67] N. Li and Z.-H. Zhou, "Selective ensemble under regularization framework," *Multiple Classifier Systems*, pp. 293–303, 2009.
[68] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.
[69] A. Lazarevic and Z. Obradovic, "Effective pruning of neural network classifier ensembles," in *International Joint Conference on Neural Networks, 2001. Proceedings. IJCNN'01.*, vol. 2. IEEE, 2001, pp. 796–801.
[70] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 245–259, 2009.
[71] D. Hernández-Lobato, G. Martínez-Muñoz, and A. Suárez, "Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles," *Neurocomputing*, vol. 74, no. 12, pp. 2250–2264, 2011.
[72] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
[73] K.-L. Du and M. Swamy, "Radial basis function networks," in *Neural Networks and Statistical Learning*. Springer, 2014, pp. 299–335.
[74] 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 and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
[75] E. Andres, D. Gonzalez, M. Martin, E. Iuliano, D. Cinquegrana, G. Carrier, J. Peter, D. Bailly, O. Amoignon, P. Dvorak, D. Funes, P. Weinerfelt, L. Carro, S. Salcedo, Y. Jin, J. Doherty, and H. Wang, "GARTEUR AD/AG-52: Surrogate-based global optimization methods in preliminary aerodynamic design," in *EUROGEN*, 2017.
[76] P. Ashill, R. Wood, and D. Weeks, "An improved, semi-inverse version of the viscous garabedian and korn method (VGK)," *RAE TR*, vol. 87002, 1987.
[77] M. Freestone, "VGK method for two-dimensional aerofoil sections part 1: principles and results," ESDU 96028, Tech. Rep., 2004.
[78] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.