Hindawi
Journal of Nanomaterials
Volume 2022, Article ID 4618832, 11 pages
https://doi.org/10.1155/2022/4618832

Research Article

A Design Optimization Method with Sparse Scattered Data and Evolutionary Computation

Yuxiang Liu,[1] **Shipei He**,[2] **Wei Liu**,[2] **and Xihong Chen**[1]

[1]*Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China*
[2]*Jiangnan Institute of Electromechanical System Design, Guizhou 590009, China*

Correspondence should be addressed to Yuxiang Liu; 2018210798@mail.chzu.edu.cn

Received 26 February 2022; Revised 27 April 2022; Accepted 11 May 2022; Published 30 May 2022

Academic Editor: Awais Ahmed

Engineering design can be regarded as an iterative optimization process. This process is difficult because of two main problems: the first is that computer-aided engineering (CAE) is time-consuming in terms of evaluating design solutions, while the second is the high dimensionality of design solutions. In the research community, a surrogate model is proposed to deal with the first problem while an evolutionary algorithm is adopted for the second. In this work, we develop a new method with only sparse scattered data, which is very common in many practical scenarios. The surrogate model can also assign a penalty factor for the predicted value, and this penalty factor can be used as one of the targets of the evolutionary algorithm to balance global exploration and local exploit. We also adopt a new evolutionary strategy, which can search high-dimensional space. Three groups of experiments are conducted to validate the proposed methods. The experimental results show that the surrogate model can predict performance and the corresponding penalty factor, the evolutionary strategy is better in terms of searching high-dimensional space compared with other evolutionary strategies, and the whole method can generate new design solutions that are near to the known design solutions. The experimental results show that this method can be used in practical scenarios, especially where they only have sparse scattered data.

1. Introduction

Engineering design is a complex process involving different design activities, and it can be generally regarded as the iteration of design and validation. Currently, with the help of well-developed computer-aided design (CAD) and computer-aided engineering (CAE) systems, the design model can be parameterized and the validation process can be simulated computationally. Therefore, engineering design can be regarded as an optimization problem [1], and different aspects of product can be optimized, such as shape optimization [2] and reliability optimization [3].

$$\hat{x} = \arg\max\ f(\mathbf{x}), \tag{1}$$

$$\mathbf{x} \in \Omega, \tag{2}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_n]$ is a parameter vector representing a design solution, Ω is the feasible solution space of $\mathbf{x}$, $\hat{x}$ is the optimal solution, and $f(\mathbf{x})$ is the evaluation function of design solutions. CAE has played the role of $f(\mathbf{x})$ successfully in engineering design for many years. However, two characters of such function make the optimization extremely complex and difficult. The first is derivative unavailable, which means the gradient-descent methods are invalid [4]; the second is time-consuming [5], which means the evaluation of the function requires extensive computational resources. In addition to above difficulties, the high dimensionality of the design space is another problem making the optimization difficult.

A response surface model (RSM) [6] is a method adopted to deal with the first two difficulties, and it is a statistical method that explores the relationships between design variables $\mathbf{x}$ and one or more response variables $f(\mathbf{x})$ [7]. The RSM method is also called as a surrogate model, metamodels, or emulators in different scenarios [8], and the underlying idea is to replace the computationally expensive model with a computationally cheaper one [8, 9]. In this

work, we use the terminology of "surrogate model" referring to the latter one.

Many technical methods can be used to build a surrogate model, such as linear or nonlinear interpolation, Kriging [10], neural network, radial basis function [11], and Gaussian process [12]. Although these methods have been successfully applied in solving many engineering problems, some remaining problems are hindering the applications of the surrogate model in engineering design. In many scenarios of applying the surrogate model successfully, either the available data is sufficient or the method of data collection is deliberately designed, such as "grid" data or "orthogonal" data. However, there are many different scenarios where the data is "sparse," which means the amount of available data is small, and "scattered," which means the data is located randomly in the feasible design space.

The main contribution of this paper is to propose a surrogate model method and a high-dimensional design space exploration method, respectively. We first attempt to build a surrogate model based on "sparse scattered" data and then use a new evolutionary strategy to explore and exploit the high-dimensional space. The combination of the two can realize the rapid generation from sparse scattered point data to a design scheme. The rest of the paper is structured as follows. Section 2 provides some related works for this study. Section 3 explains the technical details of both building a surrogate model and the evolutionary strategy. Section 4 conducts several groups of experiments to validate the proposed method. Section 5 summarizes this research and identifies some possible future works.

2. Related Works

In this section, some existing key techniques related to this work will be explained, including the surrogate model and evolutionary strategies.

2.1. Surrogate Model. There are many methods for building a surrogate model, and these methods can be categorized based on different criteria. From the perspective of data characteristics, there are methods for both grid data and scattered data. If the data is collected through Design of Experiment (DOE) [13], we can get regular grid data, and the surrogate model is easy to build. For low-dimensional problem, the Delaunay triangulation [14], natural neighbor interpolation [15], spline interpolation [16], and so on can be used to build the surrogate model, while for high-dimensional problems, the simple nearest-neighbor interpolation [17], Kriging [10], and so on can be used to build the surrogate model. Zhou et al. presented the nearest-neighbor value (NNV) interpolation algorithm for the improved novel enhanced quantum representation of digital images (INEQR). Experiments show that the proposed interpolation method has higher performance in high-resolution image recognition [18]. Qian et al. proposed a general sequential constraint updating approach based on the confidence intervals from the Kriging surrogate model (SCU-CI). Results illustrate that the proposed SCU-CI approach can generally ensure the feasibility of the optimal solution under a reasonable computational cost [19]. If the data is collected randomly, we can only get irregular grid data, which are also called scattered data. In this situation, the surrogate model becomes difficult to be constructed. There are already some methods to deal with this problem, such as triangulated irregular network, radial basis function [20, 21], and Kriging [10]. de Oliveira et al. proposed a new approach for occlusion detection—the surface-gradient-based method (SGBM) applied to a triangulated irregular network (TIN) representation. Experimental results demonstrated the feasibility of the SGBM for occlusion detection in the true orthophoto generation [22]. She et al. present a novel battery aging assessment method based on the incremental capacity analysis (ICA) and radial basis function neural network (RBFNN) model [23].

From the perspective of key techniques, the methods can also be categorized as interpolation and fitting. The former attempts to build a hypersurface that exactly passes the existing data, while the latter treats existing data as noise-contained data and attempts to find a hypersurface that minimizes the errors, and the hypersurface is unnecessary passing existing data. From the perspective of complexity, the methods can be categorized as a linear model or nonlinear model.

Although the surrogate models have been used widely in many domains, the successful applications always rely on either grid data or sufficient data. However, in many practical scenarios, the existing data is neither grid data nor sufficient. The data is sparse and scattered, and methods that fit for such situations should be developed.

2.2. Evolutionary Computation. Evolutionary computation is a commonly used method to search optimal solutions from design space. Most of the evolutionary computation methods follow the same framework, but they improve the algorithm by developing different evolutionary strategies, including crossover, mutant, and selection.

Many evolutionary strategies are developed to improve the algorithm from two perspectives. The first perspective is developing a new crossover and mutant method to generate new offspring, such as differential evolution- (DE-) based method [24, 25], immune-based method [26], particle swarm optimization- (PSO-) based method [27], and probabilistic model-based method [28]. The second perspective is developing a new selection method, such as decomposition-based method [29], preference-based method [30], indicator-based method [31], and hybrid method [32]. Although the above improvements contribute to solving many design problems, there is also requirement for developing evolutionary strategies for optimizing a high-dimensional problem.

3. Methodology

In this section, we will explain the proposed method in detail. Generally, the method has two main parts, including surrogate model construction and design solution searching. In the first part, a surrogate model, which receives design solutions as input and return performances as output, will be constructed based on a two-stage interpolation process. In the second part, we adopt a new

mutant strategy, to search the high-dimensional space. The two parts will be detailed in the following two sections, respectively.

3.1. Surrogate Model Construction. We first define the data structure used to build a surrogate model and then explain the underlying ideas of the two-stage interpolation process, and finally, we explain the technical detail of this method. The mathematical symbols used in this work are summarized in Table 1.

3.1.1. Data Structure. Typically, there are two kinds of data in many engineering design problems, and they are design solution and the corresponding performance. A design solution can be represented by a vector $\mathbf{x} = [x_1, x_2, \cdots, x_d]$, in which $x_1, x_2, \cdots, x_d$ are parameters that determine the design solution, and d is the dimensionality of the design solution $\mathbf{x}$. A performance is a measurement of design solution $\mathbf{x}$, and design solution $\mathbf{x}$ commonly has several performances for evaluation. Besides, the performance can be different under different working conditions$\mathbf{c} = [c_1, c_2, \cdots, c_l]$, in which $c_1, c_2, \cdots, c_l$ are parameters that determine the working conditions, such as temperature and velocity which can jointly define a two-dimensional $l = 2$ working condition. Therefore, the performance can be represented by a $k \times m$ matrix where k is the kinds of performances and m is the total number of working conditions.

$$\mathbf{p} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{km} \end{bmatrix}. \quad (3)$$

Generally, we have a $n \times k \times m$ matrix as database D for building the surrogate model, where n is the number of known design solutions. As shown in Figure 1, we know the k kinds of performance under m different working conditions of n known design solutions. It is noteworthy to say that the n known design solutions are scattered in the high-dimensional space.

3.1.2. Underlying Considerations and Assumptions. Based on the above data, the goal is to construct a surrogate model that receives a design solution $\mathbf{x}'$ and a group of working conditions $\mathbf{c}'$ as input and outputs the corresponding performances of the design solution $\mathbf{x}'$ under working conditions $\mathbf{c}'$. It is noteworthy that the design solutions $\mathbf{x}'$ and working conditions $\mathbf{c}'$ are generally not contained in the database D.

We can regard the surrogate model as a function $\mathbf{p} = f(\mathbf{x}, \mathbf{c})$. This function maps value from $d + l$ dimensional space R^{d+l} to k dimensional space R^k, and this high-dimensional mapping requires more data to train. Since we only have sparse scattered data, two critical assumptions are drawn before learning the surrogate model.

Table 1: The mathematical symbols used in this work.

#	Symbols	Explanation
1	$\mathbf{x}$	A design solution
2	x	A parameter of the design solution $\mathbf{x}$
3	d	The dimensionality of the design solution $\mathbf{x}$
4	$\mathbf{c}$	A working condition
5	c	A parameter of the working condition $\mathbf{c}$
6	l	The dimensionality of the working condition $\mathbf{c}$
7	$\mathbf{p}$	The performance of a design solution
8	k	The total kinds of performance
9	m	The total number of working conditions
10	n	The total number of known design solutions
11	D	The training database

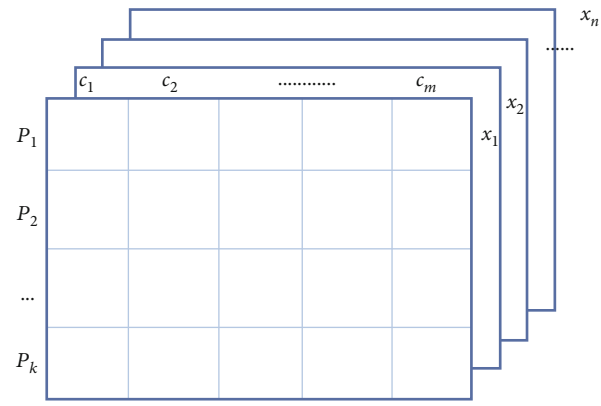

Figure 1: The data structure of the existing training data.

Assumption 1. For different design solutions, the mappings from working condition to performance are identical.

Based on Assumption 1, we do not need to learn a dedicated mapping (maps from working condition to performance) for different design solutions, which is necessary if the surrogate model is built, like $\mathbf{p} = f(\mathbf{x}, \mathbf{c})$. Therefore, we can divide the surrogate model into two stages, of which the first stage maps design solution $\mathbf{x}'$ to performances $\mathbf{p}_\mathbf{c}$ under existing working conditions $\mathbf{c}$ by $\mathbf{p}_\mathbf{c} = g(\mathbf{x}')$, while the second stage maps $\mathbf{c}'$ to performance under unknown working condition by $\mathbf{p} = h(\mathbf{c}')$. By this assumption, some dependences among design solutions, working conditions, and performance are ignored, and this loses the data requirement for training the surrogate model.

Assumption 2. The design solutions that are like existing design solutions are more feasible.

Based on Assumption 2, we make the surrogate model to assign a penalty factor for the predicted performance. If the new design solution $\mathbf{x}'$ is far from the known design

solutions, the penalty factor should be high and vice versa. This penalty factor will be used during the design optimization process, and it helps to shrink the searching space and keeps the design solution close to the regions where some design solutions are known.

3.1.3. Two-Stage Interpolation. The surrogate model can be implemented by two-stage interpolation. In both two stages, the inverse distance weighting (IDW) [33] is adopted to implement the interpolation, which computes an unknown value based on the following equation:

$$\hat{y} = \frac{\sum_{i=1}^{Q} w_i \times y_i}{\sum_{i=1}^{Q} w_i}, \tag{4}$$

where Q is the total number of data points used to predict new value $\hat{y}$ and w_i is the weight of the i^{th} data point. In the IDW method, the weight is calculated by

$$w_i = \frac{1}{\|x/-x_{\mathbf{i}}\|^{P/2}}, \tag{5}$$

where $\mathbf{x}'$ is the data point that the performance is unknown; $\mathbf{x}_i$ is the i^{th} data point; $\| \|$ is to calculate the distance between two data points in the high-dimensional space R^n; P is the order the distance, and this is a metaparameter to control the weights.

Based on this, given an unknown design solution $\mathbf{x}'$ and unknown working condition $\mathbf{c}'$, we can first interpolate to get the performances of the design solution $\mathbf{x}'$ under known working conditions $\mathbf{c}$ and then further interpolate to get the performances of the design solution $\mathbf{x}'$ under unknown working conditions $\mathbf{c}'$. We also need to assign penalty factors to the predicted performances of unknown design solutions and working conditions. In this work, we simply use the Euclidean distance as a measurement of penalty factors, and the value can be calculated by

$$\mathrm{PF} = \mathrm{PF}_1 + \mathrm{PF}_2, \tag{6}$$

where PF_1 is the penalty factor of the first stage and it can be obtained by (7), while PF_2 is the penalty factor of the second stage and it can be obtained by (8).

$$\mathrm{PF}_1 = \frac{1}{Q}\sum_{i=1}^{Q} \left\|x' - x_i\right\|^{p/2} \tag{7}$$

where Q is the total number of design solution used to predict the performance; $\mathbf{x}'$ is the data point that the performance is unknown; $\mathbf{x}_i$ is the i^{th} data point used to predict the performance; $\| \|$ is to calculate the distance between two data points in the high n dimensional space R^n; P is the order of the distance, and this is a metaparameter to control the weights.

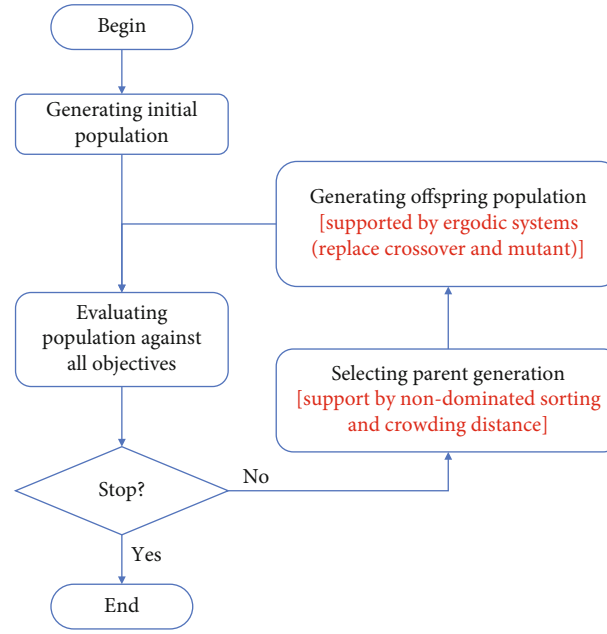

Figure 2: The main framework of ergodic evolution.

$$\mathrm{PF}_2 = \frac{1}{R} x \sum_{j=1}^{R} \left\|c' - c_j\right\|^{p/2} \tag{8}$$

where R is the total number of known working conditions used to predict the performance of unknown working conditions, $\mathbf{c}'$ is the working condition that the performance is unknown, and $\mathbf{c}_j$ is the j^{th} working condition used to predict the performance.

3.2. Design Solution Searching. Based on the method of surrogate model construction in Section 3, a surrogate model can be obtained, which receives design solution and working conditions as input and predicts performances and its corresponding penalty factor as output. Based on this surrogate model, we will adopt an evolutionary algorithm to explore the design space. The penalty factor will be taken as one of the fitness functions during the evolutionary process.

Basically, most evolutionary algorithms follow the same framework. As shown in Figure 2, the critical difference is the operation of "generating offspring population," which is marked by a bold rectangle. The traditional crossover and mutant operations are replaced by a simple ergodic system. The detail can be found in Algorithm 1. As shown in the algorithm, the ergodic system adopted in this work is a logistic map, and it is used to generate a random number for generating new offspring. The ergodic evolution method has the advantages of ergodicity and regularity and performs better in dealing with high-dimensional design space.

4. Experiment

In this section, we conduct three groups of experiments to validate the proposed method. The first group is to test whether the surrogate model can predict the performance;

Generate an initial population. Evaluate the fitness for each individual.
/∗ *D* and *EP* initialization∗/ for i =1 to PS **do**
for j =1 to Dim **do**
/∗ *DR* as a random value∗/ if $rand$ [0, 1) < DR **then**
$D_{i,j} = -1$
else
$D_{i,j} = +1$
end if
$EP_{i,j}$ = rand(0,1)
end for end for
Evaluate the fitness for each offspring individual
Select n individuals as target individuals
/∗Ergodic Search∗/for G =1 to $maxIter$ **do**
for i =1 to PS **do**
k=rand(1,Dim)
for j =1 to Dim **do**
if $rand$[0, 1) < Cr **or** $j == k$ **then** $mutant_{i,j} = target_{i,j} * (1 + D_{i,j} * EP_{i,j})$ $ergodic_{i,j} = mutant_{i,j}$
else
$ergodic_{i,j} = target_{i,j}$
end if end for
/∗Selection∗/
Nondominant sort target vectors of G_{th} and $G-1_{th}$ generations Apply crowding distance
Select n individuals as target individual
end for
/∗ *D* and *EP* update∗/ for i =1 to PS **do**
for j =1 to Dim **do**
$EP_{i,j} = logistic\ map(EP_{i,j})$
if $rand$[0, 1) < DR **then**
$D_{i,j} = -1$
else
$D_{i,j} = +1$
end if end for
end for end for
return the optimum

Algorithm 1:Nondominant sorting- and crowding distance selection-based ergodic evolution algorithm. PS: population size; Dim: dimension; D: direction factor; DR: direction factor rate; EP: ergodic parameter; G: generation; maxIter: maximum generation; i: index of individual; j: index of dimension; Cr: mutant rate; targeti,j: individual to generate offspring.

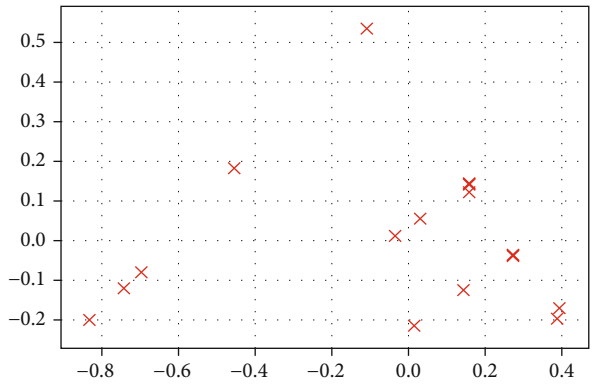

Figure 3: The 2-dimensional representation of the 20 known design solutions.

the second group is to test whether the ergodic evolution can search the high-dimensional space; the last group is to test the method as a whole and to validate the feasibility of applying it in practice.

Table 2: Experiment result of this surrogate model (first stage).

P	Q	15	10	5	3	2
1		0.01440	0.01371	0.01310	0.01340	0.01344
2		0.01488	0.01443	0.01362	0.01398	0.01359
3		0.01511	0.01462	0.01397	0.01398	0.01365
4		0.01518	0.01471	0.01406	0.01440	0.01369
5		0.01522	0.01476	0.01412	0.01445	0.01371

Table 3: Experiment result of the surrogate model (second stage).

P	R	10	5	3	2	1
1		0.02360	0.01690	0.01533	0.01289	0.01879
2		0.02619	0.01766	0.01533	0.01289	0.01879
3		0.02717	0.01781	0.01533	0.01289	0.01879
4		0.02761	0.01787	0.01533	0.01289	0.01879
5		0.02775	0.01790	0.01533	0.01289	0.01879

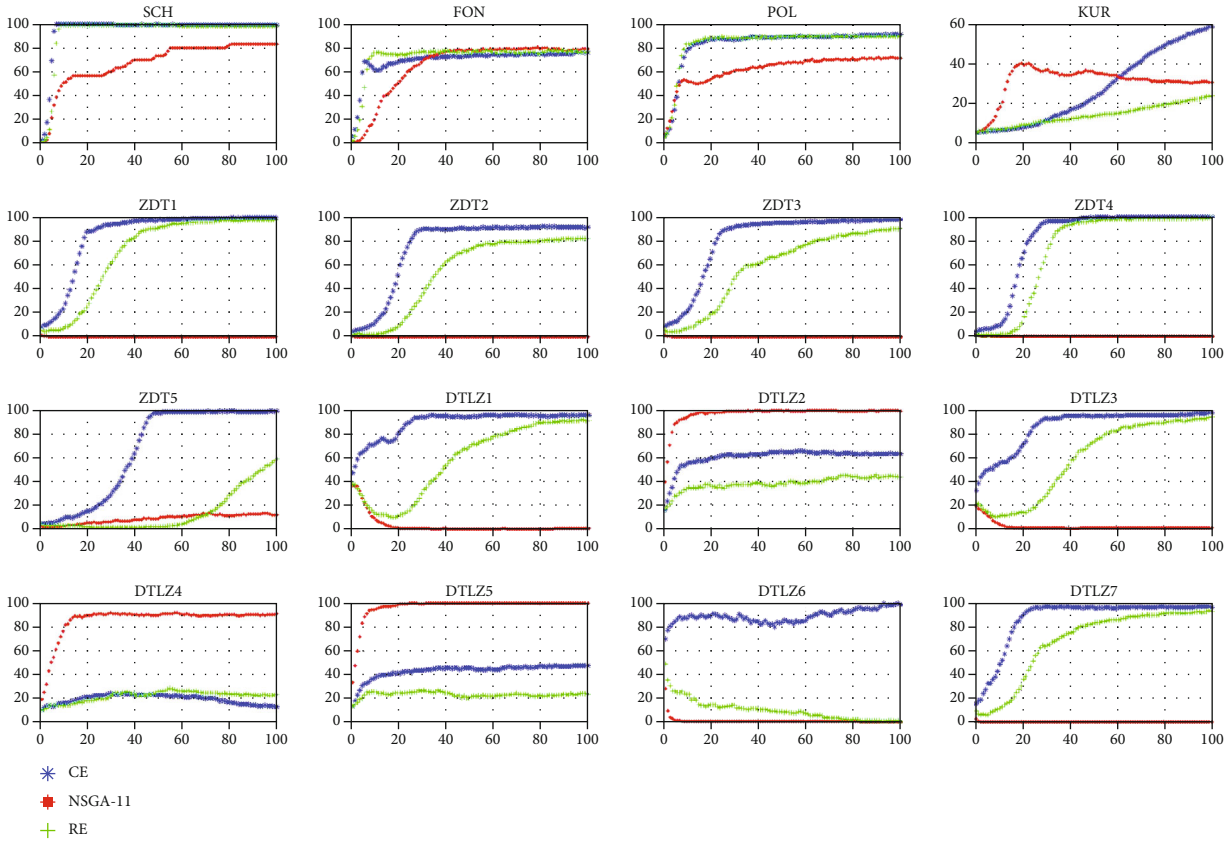

Figure 4: The average number of individuals in the nondominant frontier of the three algorithms. The blue star is CE, the green plus sign is RE, and the red dot is NSGA-II.

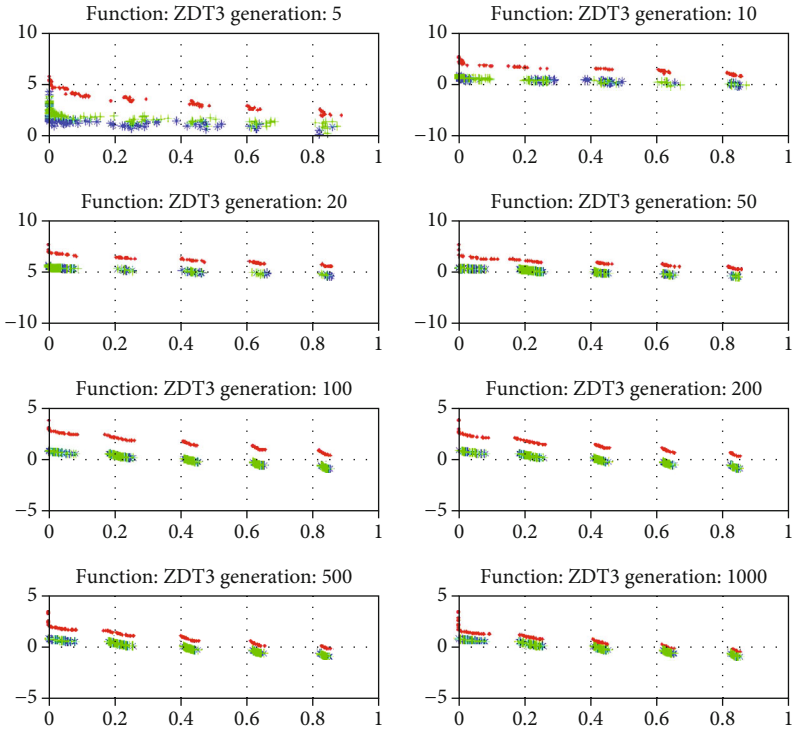

Figure 5: The Pareto frontier of the three algorithms in different generations (ZDT3). The blue star is CE, the green plus sign is RE, and the red dot is NSGA-II.

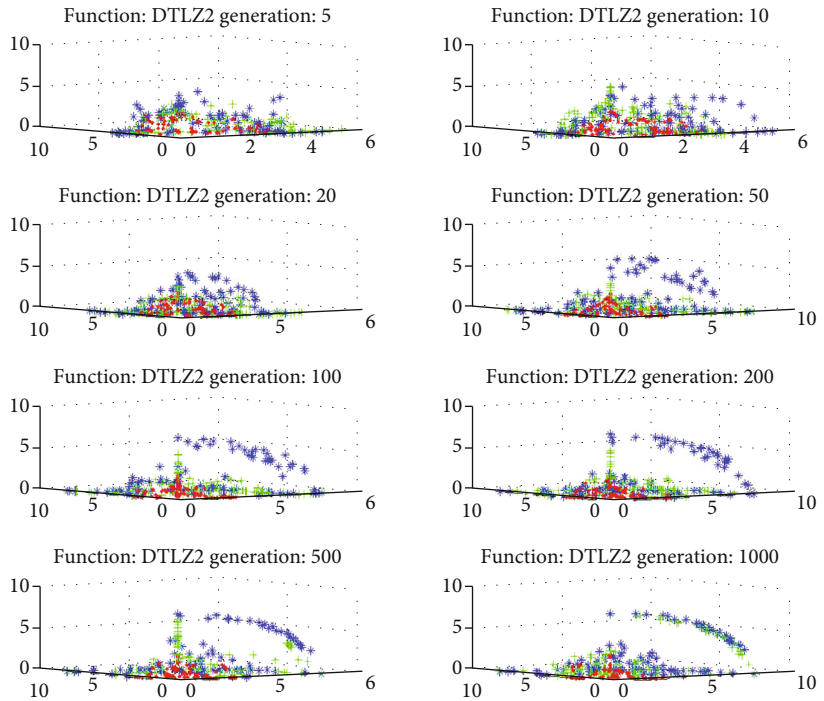

Figure 6: The Pareto frontier of the three algorithms in different generations (DTLZ2). The blue star is CE, the green plus sign is RE, and the red dot is NSGA-II.

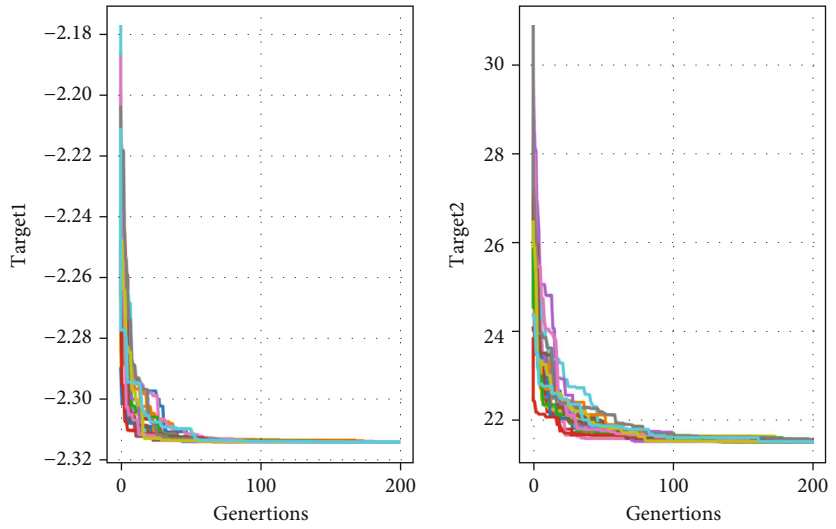

Figure 7: The evolutionary process of the two targets of 20 runs.

During the experiments, we use a small dataset provided by a research institute of aerodynamic. This dataset includes 20 known design solutions with 4 design parameters in each design solution and 4 kinds of performance under 28 working conditions. Since the 4-dimension design solutions cannot be shown in a figure, we process the 20 design solutions by Principal Component Analysis (PCA) and plot the first two components in a figure. From Figure 3, we can find that the data is sparely scattered in the design space. We regard the data with less training data and sparse distribution of the main parameter data points in the problem as sparse scattered data.

4.1. Experiment on the Surrogate Model. Based on the method illustrated in surrogate model construction of Section 3, a surrogate model based on sparse scattered data can be constructed. This section conducts several experiments to test the model and find the optimal metaparameters of the surrogate model, such as Q in (4) and P in (5).

Since the proposed method involves two stages, we test the two stages, respectively. For the first stage, we conduct experiment based on the 10-fold crossvalidation method. This method splits all 20 data into 10 groups and uses 9 groups as training data and uses the last group as test data in each experiment. In each experiment, we calculate an

Table 4: The design solutions in nondominated Pareto front of 4 runs.

Run #			Run 1				Run 2	
1	12	0	13	3.3	12	0	13	3.2
2	12	0	13	3.4	12	0	13	3.3
3	12	0.4	7	3.5	12	0	13	3.4
4	12.8	0.4	7	3.5	12	0	7	3.5
5	12.8	0.8	7	3.5	12	0.4	7	3.5
6	13.6	0.8	7	3.5	12.8	0.4	7	3.5
7	13	0.5	7.1	3.5	12.8	0.5	7	3.5
8	12.8	0.9	7.1	3.5	13	0.6	7	3.5
9	13	0.9	7.1	3.5	12.9	0.8	7	3.5
10	12.8	0.4	7.3	3.5	13	0.6	7.1	3.5
11	12.8	0.4	7.7	3.5	12	0	13	3.5
12	12	0	13	3.5	13.2	0.8	7	3.6
13	12.8	0.4	7	3.7	12.9	0.7	7	3.8
14	12.8	0.8	7	3.9	12.9	0.8	7	3.8
15	12	0.7	13	4.6	12.9	0.8	7.1	3.8
16	12	0.7	13	4.7	12	0.7	13.1	4.7
17	12	0.7	13	4.8	12	0.5	13.1	4.8
18	12	0.7	13.1	4.9	12	0.6	13.1	4.8
19	12	0.9	13.1	4.9	12	0.6	13.1	4.9
20	12	0.9	13.7	4.9	12	0.8	13.1	4.9
Run #			Run 3				Run 4	
1	12	0	13	3.2	12	0	13	3.3
2	12	0	13	3.3	12	0	13	3.4
3	12	0	13	3.4	12	0	7	3.5
4	12	0	7	3.5	12.8	0.4	7	3.5
5	12.8	0.4	7	3.5	12.8	0.5	7	3.5
6	13.2	0.5	7	3.5	12.8	0.9	7	3.5
7	12.8	0.8	7	3.5	12.8	1.1	7	3.5
8	13.2	1.2	7	3.5	12	0	13	3.5
9	12.8	0.6	7.2	3.5	12.8	0.6	7	3.6
10	13	0.6	7.4	3.5	12.8	0.7	7	3.6
11	12	0	13	3.5	13	0.7	7	3.6
12	12.8	0.9	7	3.6	12.9	1	7	3.7
13	12.8	0.4	7	3.8	12.8	0.5	7	3.8
14	12.8	0.6	7	4.1	12.8	0.5	7	4
15	12	0.4	13	4.8	12	0.4	13	4.5
16	12	0.7	13	4.8	12	0.4	13	4.6
17	12	0.7	13.1	4.8	12	1.4	13	4.7
18	12	0.9	13.1	4.8	12	0.2	13	4.8
19	12	0.4	13	4.9	12	0.4	13	4.8
20	12	1.2	13.1	4.9	12	0.4	13	4.9

averaged error by (9), and after 10 times of experiment, the error of the first stage is averaged again.

$$e = \frac{1}{2} \times \sum_{i=1}^{2} \sqrt{\frac{1}{28} \times \sum_{j=1}^{28} \left(\widehat{p}_{ij} - p_{ij}\right)^2}, \tag{9}$$

where P_{ij} is the value of the i^{th} predicted performance under the j^{th} working condition, while the P_{ij} is the real value of the i^{th} performance under the j^{th} working condition. Table 2 shows the results of different configurations of Q and P, and we can find that the surrogate model has the smallest error when $P = 1$ and $Q = 5$.

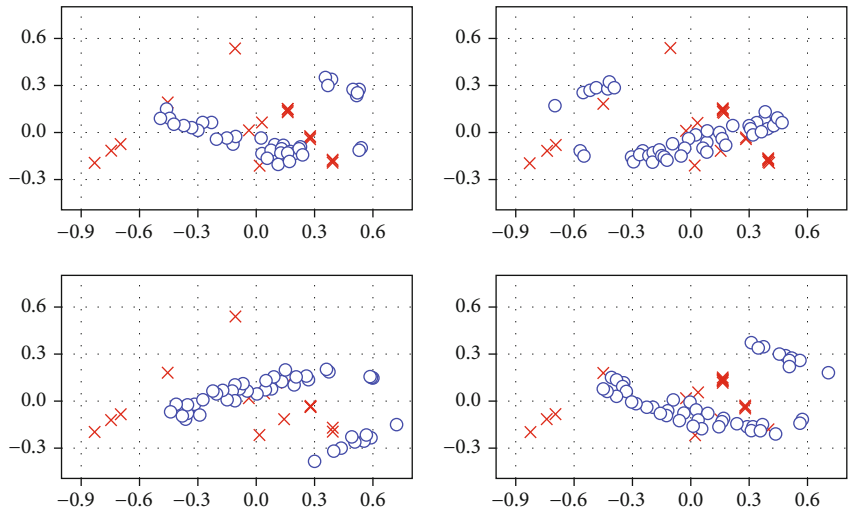

Figure 8: The 2-dimensional representation of the 20 known design solutions.

For the second stage, we also conduct experiment based on the 10-fold crossvalidation method. In each experiment, we calculate an averaged error by (9), and after 10 times of experiment, the error of the second stage is averaged again. Table 3 shows the results of different configurations of R and P, and we can find that the surrogate model has the smallest error when $P = 1$ and $R = 2$.

4.2. Experiment on Ergodic Evolution. In this section, we validate the ergodic evolution algorithm in terms of the capability of searching high-dimensional space. In this experiment, 16 benchmark functions are adopted as the test problems, including SCH, FON, POL, KUR, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, and DTLZ7. These benchmark functions are well known, and the detail can be found in [34].

We compare the performance of ergodic evolution with NSGA-II, which is a well-known evolutionary algorithm proposed in [35], and random evolution (RE), which is the same as Algorithm 1 except that the ergodic system is replaced by a general random generator. The crossover rate and mutation rate of NSGA-II are set to 0.9 and 0.1, respectively.

For each benchmark function, we run the three algorithms 30 times. For each run, the algorithms run 1000 generations with 100 individuals in each generation. After all experimental results are obtained, we conduct nondominant analysis. First, all individuals in the same generation of the three algorithms are combined, and then, count the number of individuals in the nondominant frontier after nondominant analysis. Figure 4 shows the averaged result (30 runs) of the top 100 generations. From the figure, we can clearly see that the proposed method is superior to NSGA-II and RE for most of the benchmark functions except DTLZ2, DTLZ4, and DTLZ5.

In addition, we plotted the Pareto frontiers of 5^{th}, 10^{th}, 20^{th}, 50^{th}, 100^{th}, 200^{th}, 500^{th}, and 1000^{th} generation of the three algorithms. Considering the length limitation, we only show the figures of ZDT3 (Figure 5) and DTLZ2 (Figure 6) in this paper.

4.3. Experiment on the Whole Method. In this section, we validate the proposed method. For the ergodic evolution, two targets are adopted. The first is a function of the 4 performances provided by the aerodynamic institute, while the second is calculated by equation (5). The surrogate model is constructed with $P = 1$, $Q = 5$, and $R = 2$.

We run the whole algorithm 20 times, and Figure 7 shows the evolutionary process of the two targets. For each run, the algorithm evolves 200 generations, and there are 100 individuals in each generation. The figures include 20 lines indicating the evolutionary process of the 20 runs, and each point in the line indicates the minimum value of targets in one generation. As we can see, the algorithm converges to minimum targets almost within 100 generations.

We randomly select 4 runs and list 20 design solutions in the nondominated Pareto front in Table 4. We plot the generated design solutions and known design solutions in a single figure, and both are proceeded by PCA. From Figure 8, we can find that most of the generated design solutions are near to known design solutions, which prove that the proposed method can achieve rapid exploration of high-dimensional space. This result is reasonable since the second target of the CE restricts the algorithm to exploit new design solutions in the space near to the known design solutions.

From the extensive experiment results above, we find that (1) the adoption of some basic assumptions or prior knowledge can relax the requirement of data for training the surrogate model. The prior knowledge here represents views of designers on design problems. In this work, we simply adopt two basic assumptions, and the whole model is divided into two parts, and some unimportant relationships are ignored by the surrogate model. Therefore, we think transferring basic assumption or prior knowledge into a computational manner and integrating it with the surrogate model are feasible ways to build a surrogate model with only

sparse and scattered data. (2) The simple IDW method can train the surrogate model. However, in different scenarios, IDW may not always be feasible, and some advanced methods like Kriging and RBF network should be adopted and validated. (3) One of the merits of the proposed method is that the searching space can be expanded with the addition of more known design solutions. This means that this method is fit for both sparse and scattered data and relatively bigger data.

5. Conclusion

To train a surrogate model with only sparse and scattered data and find new design solutions based on this surrogate model, this work proposed a two-stage interpolation-based method for the surrogate model and adopts CE to explore the high-dimensional space. This paper uses PCA to reduce the dimension of data and prove the characteristics of sparse and scattered data. Then, combining the two-stage interpolation and ergodic evolution method, the sparse scattered point data is generated into the design method. Three groups of experiments show that the surrogate model can predict performances, and the CE is efficient in terms of exploring high-dimensional space.

Although this work proposes a feasible method to deal with the problems, some remaining problems require extensive research works. In the future, we will enhance the method from the following aspects.

(i) This work only adopts basic assumptions to relax the data requirement, and the data requirement can be further relaxed by incorporating prior knowledge, such as experts' experience and physical law. The main problem will be the technique of embedding prior knowledge into surrogate models

(ii) In this work, the penalty factor is important and the simple Euclidean distance is used. In the future, some nonlinear function of the Euclidean distance can be used to measure the penalty factor. The difficulty will be how to predefine or learn a nonlinear function for the penalty factor

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] W. Yi, Y. Zhou, L. Gao, X. Li, and C. Zhang, "Engineering design optimization using an improved local search based epsilon differential evolution algorithm," *Journal of Intelligent Manufacturing*, vol. 29, no. 7, pp. 1559–1580, 2018.

[2] A. R. Yıldız, "A new design optimization framework based on immune algorithm and Taguchi's method," *Computers in Industry*, vol. 60, no. 8, pp. 613–620, 2009.

[3] P. Zhu, Y. Zhang, and G. Chen, "Metamodeling development for reliability-based design optimization of automotive body structure," *Computers in Industry*, vol. 62, no. 7, pp. 729–741, 2011.

[4] A. R. Conn and S. L. Digabel, "Use of quadratic models with mesh-adaptive direct search for constrained black box optimization," *Optimization Methods and Software*, vol. 28, no. 1, pp. 139–158, 2013.

[5] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219–241, 2010.

[6] I. Karen, N. E. C. M. E. T. T. İ. N. Kaya, and F. Öztürk, "Intelligent die design optimization using enhanced differential evolution and response surface methodology," *Journal of Intelligent Manufacturing*, vol. 26, no. 5, pp. 1027–1038, 2015.

[7] Wikipedia, "Response surface methodology," 2017, https://en.wikipedia.org/wiki/Response_surface_methodology/.

[8] R. Mallipeddi and M. Lee, "An evolving surrogate model-based differential evolution algorithm," *Applied Soft Computing*, vol. 34, pp. 770–787, 2015.

[9] W. Gong, A. Zhou, and Z. Cai, "A multioperator search strategy based on cheap surrogate models for evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 746–758, 2015.

[10] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, "Kriging models for global approximation in simulation-based multidisciplinary design optimization," *AIAA Journal*, vol. 39, no. 12, pp. 2233–2241, 2001.

[11] R. G. Regis and C. A. Shoemaker, "Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization," *Engineering Optimization*, vol. 45, no. 5, pp. 529–555, 2013.

[12] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, and C. F. J. Wu, "Building surrogate models based on detailed and approximate simulations," *Journal of Mechanical Design*, vol. 128, no. 4, pp. 668–677, 2006.

[13] C. Tang, K. Gee, and S. Lawrence, "Generation of aerodynamic data using a design of experiment and data fusion approach," in *In: Aiaa Aerospace Sciences Meeting and Exhibit*, Reno,Nevada, January 2005.

[14] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, *Computational Geometry: Introduction*, Springer, 2008.

[15] Q. Li, S. Chen, and X. Luo, "Using meshless local natural neighbor interpolation method to solve two-dimensional nonlinear problems," *International Journal of Applied Mechanics*, vol. 8, no. 5, p. 1650069, 2016.

[16] S. H. Hong, L. Wang, T. K. Truong, T. C. Lin, and L. J. Wang, "Novel approaches to the parametric cubic-spline interpolation," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1233–1241, 2013.

[17] L. M. Surhone, M. T. Tennoe, and S. F. Henssonow, *Nearest-Neighbor Interpolation*, Betascript Publishing, 2013.

[18] R. G. Zhou, W. W. Hu, G. F. Luo, X. A. Liu, and P. Fan, "Quantum realization of the nearest neighbor value interpolation method for INEQR," *Quantum Information Processing*, vol. 17, no. 7, pp. 1–37, 2018.

[19] J. Qian, J. Yi, Y. Cheng, J. Liu, and Q. Zhou, "A sequential constraints updating approach for Kriging surrogate model-assisted engineering optimization design problem," *Engineering with Computers*, vol. 36, no. 3, pp. 993–1009, 2020.

[20] R. G. Regis, "Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points," *Engineering Optimization*, vol. 46, no. 2, pp. 218–243, 2014.

[21] R. G. Regis and C. A. Shoemaker, "Constrained global optimization of expensive black box functions using radial basis functions," *Journal of Global Optimization*, vol. 31, no. 1, pp. 153–171, 2005.

[22] H. C. de Oliveira, A. P. Dal Poz, M. Galo, and A. F. Habib, "Surface gradient approach for occlusion detection based on triangulated irregular network for true orthophoto generation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 2, pp. 443–457, 2018.

[23] C. She, Z. Wang, F. Sun, P. Liu, and L. Zhang, "Battery aging assessment for real-world electric buses based on incremental capacity analysis and radial basis function neural network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3345–3354, 2020.

[24] X. Qiu, J. X. Xu, K. C. Tan, and H. A. Abbass, "Adaptive cross-generation differential evolution operators for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 232–244, 2016.

[25] K. Sethanan and R. Pitakaso, "Improved differential evolution algorithms for solving generalized assignment problem," *Expert Systems with Applications*, vol. 45, pp. 450–459, 2016.

[26] Z. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Applied Soft Computing*, vol. 8, no. 2, pp. 959–971, 2008.

[27] A. Rahimi-Vahed, S. Mirghorbani, and M. Rabbani, "A new particle swarm algorithm for a multi-objective mixed-model assembly line sequencing problem," *Soft Computing*, vol. 11, no. 10, pp. 997–1012, 2007.

[28] W. Dong and X. Yao, "Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms," *Information Sciences*, vol. 178, no. 15, pp. 3000–3023, 2008.

[29] R. Carvalho, R. R. Saldanha, B. Gomes, A. C. Lisboa, and A. Martins, "A multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi-Uda antennas," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 803–806, 2012.

[30] L. Rachmawati and D. Srinivasan, "Multiobjective evolutionary algorithm with controllable focus on the knees of the Pareto front," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 810–824, 2009.

[31] J. Bader and E. Zitzler, "Robustness in hypervolume-based multiobjective search," *Computer Engineering and Networks Laboratory*, vol. 317, 2010.

[32] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[33] G. Y. Lu and D. W. Wong, "An Adaptive Inverse-Distance Weighting Spatial Interpolation Technique," *Computers & Geosciences*, vol. 34, no. 9, pp. 1044–1055, 2008.

[34] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[35] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, *A Fast Elitist Nondominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II*, Springer, Berlin Heidelberg, 2000.