Applied Soft Computing 113 (2021) 107883

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Surrogate-adjoint refine based global optimization method combining with multi-stage fuzzy clustering space reduction strategy for expensive problems

Kai Wu [a], Faping Zhang [a,*], Yun He Zhang [a], Yan Yan [a], Shahid Ikramullah Butt [b]

[a] School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China
[b] School of Mechanical and Manufacturing Engineering, National University of Sciences & Technology, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 21 June 2020
Received in revised form 1 July 2021
Accepted 5 September 2021
Available online 15 September 2021

Keywords:
Global optimization
Surrogates Model
Design space reduction
Fuzzy clustering
Gauss process

ABSTRACT

In engineering optimization, surrogate model (SM) is widely used to replace the involved time expensive model, due to the expensive model is complex and high precise requirement caused a long calculation cycle. In traditional process of engineering optimization, the separation of the surrogate model static construction stage and dynamic optimization stage depresses the optimization accuracy and efficiency. Moreover, in order to ensure the accuracy of the surrogate model, expensive model had to be intensively invoked to get enough representative samples in the design space for the SM training. In this paper, a surrogate model adjoint refine based global optimization method combining with the multi-stage fuzzy clustering space reduction strategy (MFCPR-SGO) is proposed to improve the optimization accuracy and efficiency. Firstly, the optimal Latin hypercube design method (OLHD) is used to sample in design space to assure the initial sample set with strong space filling property. Then, the design space is subdivided into three tiered subspaces by using the space reduction strategy of multi-stage fuzzy clustering, which has the ability of space focusing, space reduction and jumping out of local optimum. On this basis, the hierarchical optimization method with ADAM gradient descent is proposed to quickly and accurately search the local minimum value of the objective function in each subspaces. At the same time, combined with the extremum sampling and the gaussian process sampling, a dynamic sampling algorithm is given to realize the synchronization of optimization and surrogate model update. Finally, the benchmark test problems in 12 different dimensions are used to verify the proposed method. The results show that the optimization accuracy can be improved by 21.3% and expensive model invoking times are reduced by 31.5% compared with other three heuristic optimization methods and the three recent surrogate-based optimization (SGO) algorithms. It indicated that the optimization precision and efficiency can be greatly improved by synchronizing the dynamic updating of the surrogate model with the engineering optimization search.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.6727417.v1

1. Introduction

With rapid development of computer techniques, numerical simulation and design optimization of complex product have entered new era which supported by finite element technology and multi-body dynamics technology, so that the designed product has better performance [1,2]. However, most of the high performance simulation models used for design optimization are computationally expensive models, which is of very time resource consumption at a time. There are the instances such as the design optimization of finite element model of vehicle collision [3], vehicle stationary model with vertical vibration [4,5], the airframe lift model of civil aircraft [6] With the increasing complexity of the high performance required product, it becomes popular that thousands times of expensive model invoking are needed to explore the design space and get the global optimal solution. Consequently, this approach will lead to unacceptable computing burdens, even with today's vastly improved computer performance [7,8]. How to improve the efficiency of design optimization has become an important research focus.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

[*] Corresponding author.
E-mail addresses: wukai_bit@qq.com (K. Wu), zfpnew@163.com (F. Zhang).

https://doi.org/10.1016/j.asoc.2021.107883
1568-4946/© 2021 Elsevier B.V. All rights reserved.

In recent years, known as metamodel or response surface model, surrogate model (SM) [9,10] could find the implicit nonlinear relationship of the engineering question in a data-driven way and express it in the form of black box model, and has been widely used in the products optimization design in the aviation, aerospace, navigation and automobile industries, which has greatly improved the efficiency of optimization design [11].

Compared with the traditional global optimization method, global optimization with surrogate model (SGO) [12,13] can significantly reduce the use of expensive models by searching the design space under the guidance of SM and plays an important role in engineering design optimization. SGO search process includes three main steps generally: (1) Select sample points by experiment design (DOE) method [14,15] such as fractional factor design (FFD), central composite design (CCD), Latin hypercube design (LHD) [16,17], etc. (2) Construct the SM. Typical surrogate models include kriging (KRG) [18], radial basis function (RBF) [19], polynomial regression (PR) [20], artificial neural network (ANN) [21], etc. (3) Global optimization methods are used to solve engineering optimization problems, such as genetic algorithm (GA) and simulated annealing (SA).

The traditional SGO methods separate the construction of the SM and the optimization iterative as two independent processes. The one process is to build an accurate SM with a large number of sample points in the whole design space sampling by using the DOE method, and the other is to conduct optimization search by the well-constructed SM. In this way it is difficult to improve the optimization efficiency while ensuring the optimization accuracy. Therefore, how to further improve the efficiency and accuracy of SGO method simultaneously has become the focus of optimization research.

2. Related works

In recent years, using the SGO methods to solve the problem of long optimization cycle caused by expensive models have been widely concerned by scholars. In the process of engineering optimization, the traditional SGO methods usually build the surrogate model statically and carry out the dynamic optimization process separately. The SM was constructed by sampling with DOE, and then the optimal solution was obtained by iterative optimization on the SM. The detail process of SGO method was introduced in review articles [22] which compared the characteristics and robustness of different SGO methods. Through the neural network model, Zhang Jian [23] constructed the relationship between the running comfort index of vertical vibration of high-speed train and suspension stiffness, damping, and searched on the neural network model through the improved differential evolution algorithm to realize the optimal design of high-speed train. Ma Shilei [24] introduced the sparse grid method into the establishment of multidisciplinary approximate model, applied the approximate model to the optimal design of NASA reducer, and made a lightweight design of a drive axle housing based on it. The traditional SGO method used in current researches needs a large number of samples at the beginning of operation to control the global error of the SM in the whole design space, so as to ensure the optimization accuracy. However, the number of samples required will increase explosively with the increase of the dimension and size of the design space. So it is difficult for traditional SGO methods to make breakthroughs in the improvement of optimization accuracy and efficiency.

In order to solve the problem of low computational efficiency caused by large number of sampling in traditional SGO methods, the researchers proposed an adaptive SM updating framework [25]. Under this framework, the initial SM is constructed with fewer initial sample points, and the further update process of SM is carried out in each optimization iteration . By constantly sampling and updating the SM in the region of interest, the accuracy of searching and locating the region where the global optimal solution may exist by SM in the global design space are improved, hence the construction of the SM economically with few sample points is possible. The update strategy of self-adaptive SM can be divided into three directions. The advantages and disadvantages of each direction are as follows:

(1) Sampling strategy by adding an approximate optimal solution

Sampling strategy of adding an approximate optimal solution is the simplest adaptive approximate optimization strategy. Acar E [26] optimized on the SM by using the global search algorithm and obtained the approximate optimal solution. In each iteration, the SM was updated with the approximate optimal solution until getting the global optimization solution. However, when the sample points are too sparse or the uncertainty of the SM is large, it is difficult to ensure the accuracy of the global optimal solution by using this method.

(2) Sequential sampling strategy by design space filling

The surrogate model management and update strategy which uses sequential sampling strategy by design space filling is carried out to sample directly according to the error information of the SM. Chaudhuri [27] improved the efficient global optimization method (EGO) by introducing an Expected Improvement (EI) as the sampling criterion for determining the space filling sequence, moreover continuously searching for the maximum position of EI and sampling here to update the SM. EGO has high efficiency and global convergence because its criterion formula considers the error of the new points. In addition to EI criterion, other criteria has been used to determine the sampling points of space filling sequence either, such as Probability of improvement (PI) [28]. TANG [29] used the hybrid surrogate model to approximate the expensive model, determined the new sample points according to the fitness function value in the process of particle swarm evolution, and proposed a surrogate model-based PSO algorithm (SBPSO).

In the process of update SM, the sequential sampling strategy by design space filling only used the error information of SM and neglected the distribution of the global optimal solution in the design space. The SM update process is not closely combined with the optimization iteration process in this strategy. Therefore, when the dimension of each design variable is high, the convergence rate would be greatly reduced [30].

(3) Sequential sampling strategy by design space reduction

The sequential sampling strategy by design space reduction finds out several regions which may contain the global optimal solution through searching the initial SM, and uses the DOE method to sample in these regions to update the SM until obtaining the global optimal solution. The sampling methods with space reduction include trust region method (TR), segmentation plane method (SP) and key design space method (KDS).

The trust region method adjusts the size of the interest space (ROI) according to the predictive ability of the target function of the SM. The standard trust region method relies on gradient information and is widely used in the multi-precision approximate optimization strategy based on the SM. For example, PEREZ [31] proposed the self-adaptive design method with the trust region to update the PRSM model by gradually increasing the sample points. GANO [32] used the trust region method to update the KRG model in multi-precision optimization. In addition, rather than relying on some gradient information, Long [33] proposed a trust region based optimization strategy for dynamic SM. The SM management and update strategy with trust region is efficient and can obtain the global optimal solution in most cases, but it may miss the global optimal solution in the multi-peak problem.

WANG [34] proposed a new self-adaptive response surface optimization strategy. They deleted the area which have a large response value of the SM in design space through the segmentation plane method, and then sampled in the reduced region of interest (ROI) to update the PRSM model. The segmentation plane method could effectively reduce the space and fast located to the optimal solution. However, it usually missed the global optimal solution due to it cannot increase ROI.

Another surrogate model management and updating strategy with design space reduction is the spatial design method. Long [35] constructed ROI directly according to optimality and approximate accuracy of SM by using the key design space method. The method can improve the approximate accuracy of the SM in the vicinity of the optimal solution more efficiently and the parameters of the method are less than that of the trust region method. In addition, Xu [36] reduced the design space by using self-organizing Maps (SOM) method, and further reduced the design space and constructed SM by using the variable precision modeling method with Kriging addition scale, which performance was proved by numerical cases and engineering cases. Principal component analysis (PCA) has been used to reduce the dimension of the design space and combined with BP neural network to model and predict the thermal drift error of the spindle [37].

By means of reducing the design space, sequential sampling strategy could make the sample center focusing on ROI, hence ensuring the fast convergence of the optimization searching and improving the ability to explore the global optimal solution. Now, for complex multi maximum problems, how to ensure the sequential sampling method with space reduction converging to the global optimal solution is still a problem deserving study .

In summary, the dynamic update methods of the SM can be categorized as global update mode and local update mode. Global update mode guarantees the accuracy of all locations in the design space, but usually a large sample set is required. While local update mode focuses limited computational power on the ROI in the whole design space, thus enabling us to use fewer sample points with the same accuracy, further improving the modeling efficiency.

In order to improve the efficiency and accuracy of the engineering design optimization of the SM, based on the sequential sampling strategy of space reduction, this paper proposes a surrogate-adjoint refine based global optimization method combining with multi-stage fuzzy clustering space reduction strategy (MFCPR-SGO) by synchronizing the SM dynamic updating process with the optimization iteration process. Compared with the existing methods, the optimization precision is further improved and the sample usage is reduced.

The paper is organized as follows: in Sect.3, ESEA-OLHD is constructed to improve the spatial filling of the initial sample set. On this basis, the RBF method is further used to construct the SM of the objective function and design variables, aiming to reduce the sensitivity of SM to the number of samples and noise. In Sect.4, a multi-stage fuzzy clustering space reduction strategy is proposed to divide design space into three subspaces, and corresponding searching methods in these subspaces are given. The optimization method based on ADAM gradient descent for the three subspaces is proposed in Sect.5. In Sect.6, the benchmark test problems in 12 different dimensions (2∼16 dimensions) is used to verify the significance of the proposed model.

3. Construct initial surrogate model with optimal Latin cube experiment design

The accuracy of the SM depends on the spatial uniformity of the initial sample set, the sensitivity of the SM construction method to noise and sample size. In this section, ESEA-OLHD algorithm is used to sample in the design space to improve the spatial filling of the initial sample set. On this basis, RBF method is used to construct the SM of the objective function, so as to reduce the sensitivity of SM to sample size and noise.

Table 1
Table of commonly used radial basis functions.

Name	Radial function, $r = \|x - x_i\|_2$
Linear	$\phi(r) = cr$
Cubic	$\phi(r) = (r + c)^3$
Thin plate spline	$\phi(r) = r^2 \log(cr^2)$
Gaussian	$\phi(r) = \exp(-cr^2)$
Multi quadratic	$\phi(r) = (r^2 + c^2)^{1/2}$

3.1. Construct the RBF surrogate model

The radial basis function (RBF) model has significant fitting ability and fast training speed in different scenarios, less precision loss under the influence of noise, and performs well in the case of few samples. So the extended RBF model is selected as a tool to fit the relationship between design variables and response indicators in this paper.

The radial basis function (RBF) model is a method of constructing approximate model by linear weighting the radial function, consists of input layer, hidden layer and output layer, which can be regarded as a simple single hidden layer neural network [38]. The basic formula for constructing RBF model is shown as:

$$y(x) = \sum_{i=1}^{N} \omega_i \phi(\|x - x_i\|) \tag{1}$$

where, N is number of sample points, ω_i is the coefficient obtained by least square method, x_i is the coordinate vector of the ith sample point, $\|*\|$ is the Euclidean distance, ϕ is the radial basis function. It is important to select the radial basis functions to train the RBF model, because the RBF model can be represented as a linear combination of Euclidean distance and weight value as shown in Eq. (1). The commonly used radial basis functions ϕ are shown in Table 1, where c is a constant that is normally valued at 1.

The extended RBF model [39] adds a linear polynomial on the basis of Eq. (1) to enhance its fitting ability of linear problems. The mathematical expression is:

$$y(x) = \sum_{i=1}^{N} \omega_i \phi(\|x - x_i\|) + \sum_{j=1}^{m} b_j p_j(x) \tag{2}$$

where, $p_j(x)$ is the polynomial which usually expressed in the form of first-order polynomial. m is the number of polynomials which usually value at $m = N + 1$, and N is the number of sample points. Since the number of unknown coefficients is more than the number of samples after introducing the linear polynomials, so orthogonality conditions need to be added:

$$\sum_{i=1}^{N} \omega_i p_j(x_i) = 0, j = 1, 2, \ldots, m \tag{3}$$

Coefficient matrices w and b of RBF model can be obtained by taking the sample set into the joint Eq. (2)∼(3).

3.2. Optimal Latin cube design for training sample set

Whether the initial SM can effectively characterize the response surface in the entire design space depends on the spatial uniformity of the initial training sample set. The sample points that are gotten by traditional DOE methods such as the center

Table 2
Computational complexity and adaptability of different criteria in ESEA-OLHD method.

	ϕ_p	CL2	Entropy
Computational complexity	$O(n) + O(n\log 2(p))$	$O(n)$	$O(n^2) + O(n) \sim O(n^3) + O(n)$
Adaptation problem	6∼10	>10	2∼6

complex method, the whole factorial design, uniform design (UD) [40] and Latin hypercube design (LHD) have poor uniformity in multidimensional space. In this section, the Optimal Latin hypercube design with enhanced stochastic evolutionary algorithm (ESEA-OLHD) [41] is proposed to sample in the initial design space, so as to obtain high spatial uniformity sample points and reduce the number of sample points as much as possible in different dimensions.

The ESEA-OLHD method is suitable for different spatial dimensions, different spatial dimensions, and different optimality criteria. After the initial design space is given, the process of obtaining the sample set with high spatial uniformity is divided into four parts:

(1) Generate the initial sample set. The conventional LHD method is used to generate an initial sample set randomly, which may have poor spatial uniformity.

(2) Get a new sample matrix through the element exchange method. The new sample matrix is obtained by replacing several pairs of elements in a column of the original sample set matrix.

(3) Evaluate index of sample matrix. According to the dimensional characteristics of the experimental design problem, ϕ_p, entropy and CL2 are used to evaluate the index of new sample matrix, and the exploration coefficient n_{acp} and improvement coefficient n_{imp} are updated according to whether the evaluation index is improved or not.

(4) Construct optimization framework with inner loop and outer loop. The inner loop constructs a new design based on the initial random sample set through the element exchange strategy and decides whether to accept the design result according to evaluation criteria. The outer loop forms a complex combination strategy by improving coefficient n_{imp} and exploring coefficient n_{acp}, and dynamically adjusts the convergence threshold Th of the algorithm to control the whole optimization process. Through the optimization framework with inner loop and outer loop, the final sample is guaranteed to be uniform in space and adaptable to design problems.

In addition, the evaluation criteria is repeatedly invoked in the iteration of sampling. Therefore, in order to improve the overall efficiency of ESEA-OLHD method, different optimal evaluation criteria are used to evaluate problems of different dimensions. Table 2 shows the calculation complexity of ESEA-OLHD algorithm in different dimensions by combining three indexes.

The detailed running procedure of ESEA-OLHD algorithm is shown in Appendix C.

4. Multi-stage design space reduction

Sampling a large number of points to ensure the accuracy of the SM at every point is obviously not desirable in engineering optimization problems with expensive model when the feasible regions of design variables are large. The multi-stage space reduction strategy is proposed in this section by reducing the sampling space to three tiered dynamically changing subspaces during optimization iteration to effectively reduce the amount of samples used to update the SM and the number of expensive model computations invoked during optimization. For nonlinear multi-peak problems, this method still has good effectiveness.

4.1. Division of three kinds of subspaces

Multi-stage design space reduction strategy reduces the design space step by step in the iterative process to reduce the negative impact of the large initial feasible region. Firstly, the initial sample set is obtained by ESEA-OLHD method to train the initial SM. Subsequently, three tiered subspaces can be obtained through the space reduction strategy in each iteration. The scope of the subspace that is finally determined in each optimization iteration will be used as the benchmark space for the next iteration.

Multi-stage space reduction divides the entire design space into the following three tiered subspaces, and the relationship of each subspace is shown in Fig. 1:

(1) Original design area (OA): OA is the benchmark space for the first iteration which is equivalent to the given initial design space, and remains unchanged in the following iteration process. In order to make the algorithm jump out of the local optimal solution, the combination sampling strategy with extremum method and gaussian error method is implemented in the OA subspace to improve the overall accuracy of the SM. On that basis, the hierarchical optimization method with ADAM gradient descent and the relaxation discriminant formula are used to find the possible missing minimum value.

(2) Middle transition area (MA): MA is a transition region between OA and IA. Each MA subspace is a preliminary reduction of the design space where may locate one or more local minimum values of the optimization problem. The hierarchical optimization method with ADAM gradient descent is used to find the local minimum in MA subspaces, and the sample points are obtained by the extremum method to update the SM. By comparing the local minimum values, the optimal prediction solution of this iteration is determined. By means of parallel optimization and sampling of multiple reduced MA subspaces, it is easily to find most of the local optimal regions and avoiding the loss of the global optimal solution.

(3) Important area (IA): The region most likely containing the global optimal solution is IA subspace, which will be the benchmark space in the next iteration. In the same way, the hierarchical optimization method with ADAM gradient descent is used to carry out optimization search in IA subspace. The combination sampling strategy is adopted to focus on sampling in IA subspace to further explore and continuously improve the SM.

By dividing the design space in this way, the algorithm is conducive to saving a lot of computing resources and preserving the global exploration ability by focusing the optimizing and updating of the SM to a smaller region that will continue to be reduced in the iterative process. In addition, the new sample points for updating the SM will be obtained from the MA and IA subspaces in turn during the iteration. When the algorithm falls into local optimization, the new sample points will be obtained by sampling in the OA subspace.

4.2. The determination of MA by quadratic clustering algorithm

In the process of the surrogate-based optimization method with the space reduction strategy, MA is several neighborhood regions formed by each local minimum point, and the global optimal solution is most likely to be in one of MA subspaces. In this part, the quadratic clustering method is used to determine MA subspaces according to the response distribution characteristics of the sample points. The determination of MA subspace can be divided into the following steps as shown in Fig. 2 and Algorithm.1.

(1) population verification by first clustering (FCM). The clustering quantity interval C is determined when all the sample

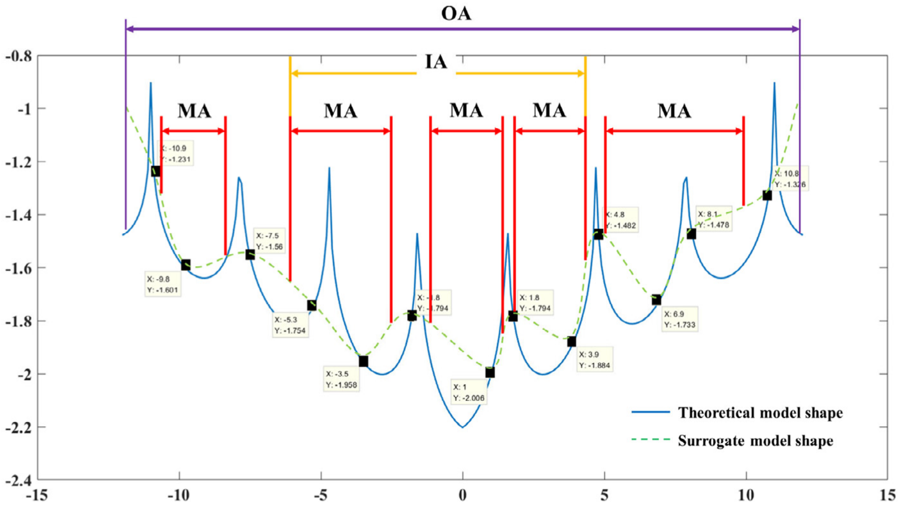


Fig. 1. The diagram of three subspaces in one iteration.

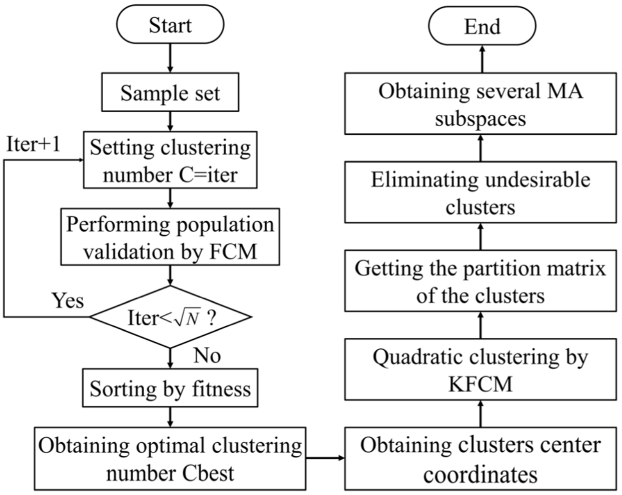

Start
Sample set
Iter+1
Setting clustering number C=iter
Performing population validation by FCM
Yes
Iter<√N ?
No
Sorting by fitness
Obtaining optimal clustering number Cbest
Obtaining clusters center coordinates
Quadratic clustering by KFCM
Getting the partition matrix of the clusters
Eliminating undesirable clusters
Obtaining several MA subspaces
End

Fig. 2. The process of determining the MA subspace with quadratic clustering.

points in the benchmark space of this iteration are given. FCM clustering is conducted with each value in the quantity interval C, and the validity of clustering results in each group is evaluated in evaluation indexes. The optimal clustering quantity c_{best} is obtained according to evaluation results, and the corresponding clustering center coordinates are retained.

(2) population division by secondary clustering (KFCM). KFCM clustering is carried out on the sample set to obtain c_{best} clusters that contain local minimum with the priori conditions obtained in the above step. In order to eliminate some undesirable clusters, the top 50% (empirical value) clusters with low average response are selected as MA subspaces, and the subspaces ranges are determined according to the maximum and minimum values of each dimension coordinates of the samples contained in each cluster.

4.2.1. Initiatory clustering and verification based on FCM

The fuzzy c-means clustering algorithm (FCM) is efficient and robust, but the number of clustering c is needed as the prior information. Meanwhile in the automatic process of optimization iteration, c cannot be set manually for each time of clustering. Therefore, cluster validation is proposed in this paper to obtain the optimal clustering quantity adaptively in each iteration process [39].

First, we set the feasible space of population number as $2 \leq C \leq [\sqrt{N}]$. Where, N is the number of samples, and $[*]$ is the integer of real number *. Within this value range, $[\sqrt{N}]-1$ groups of clustering results can be obtained by FCM clustering algorithm. With the number of clusters $c \in C$ and $c \in Z^+$, the correlation between each data point and each cluster center can be measured by Mahalanobis distance norm, and the cost function is formed by weighting the correlation value by Eq. (4). The goal of the FCM algorithm is to minimize this equation:

$$J(X,U,V)=\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^m\left\|x_j-v_i\right\|_A^2 \tag{4}$$

where, U is the fuzzy c partition matrix and V is the clustering center, and they are the outputs of the cost function. Moreover, c is the number of clusters, N is the number of sample points. The index m which usually values at 2 is a weight factor that determines the degree of population fuzziness. The formula of Mahalanobis distance norm is as follows:

$$D_{ij}^2=(x_j-v_i)^T A_i(x_j-v_i) \tag{5}$$

$$A_i=[\rho_i \det(F_i)]^{1/N}F_i^{-1} \tag{6}$$

$$F_i=\frac{\sum_{j=1}^{N}(\mu_{ij})^m(x_j-v_i)(x_j-v_i)^T}{\sum_{j=1}^{N}(\mu_{ij})^m} \tag{7}$$

The Mahalanobis distance norm can be obtained by the joint Eqs. (5)∼(7), that is, the matrix D, where m is usually equal to 2. Each u represents the membership degree of the data point and the jth clustering center v. The calculation method of membership degree and the constraint conditions that the variable μ should satisfy with in the calculation process are as follows:

$$\mu_{ij}=\frac{1}{\sum_{k=1}^{c}\left(\left\|x_j-v_i\right\|/\left\|x_j-v_i\right\|\right)^{\frac{2}{m-1}}}$$
$$s.t.\sum_{i=1}^{c}\mu_{ik}=1,\ \mu_{ik}\in[0,1] \tag{8}$$

On this basis, in the case of formulating the convergence rule, the optimal solution (U,V) can be obtained by Eq. (9).

$$\min\{J(U,V)\}$$
$$\max_{ij}\left\{\left|\mu_{ij}^{(T+1)}-\mu_{ij}^{(T)}\right|\right\}<\varepsilon \tag{9}$$

After solving the minimization problem of Eq. (9), the results of dividing samples by the first clustering can be obtained when the number of clusters is set to c, which includes the center matrix V of each cluster and the sample point matrix U of each cluster. In order to evaluate the validity of clustering results, the multi-index weighting method is introduced in this paper.

According to the characteristics and optimization trend of each discriminant index in Table 3, the three indexes (2), (3) and (4) are selected to form a joint discriminant formula. After internally standardizing the three discriminant results in the base unified scale [0,1], the cluster effectiveness evaluation results are obtained by weighting the three indexes as Eq. (10).

$$\begin{cases} g = \omega_1 * \left\| CE_{[\sqrt{N}]x1} \right\|_{0\sim 1} \\ \qquad + \omega_2 * \left\| PI_{[\sqrt{N}]x1} \right\|_{0\sim 1} + \omega_3 * \left\| SI_{[\sqrt{N}]x1} \right\|_{0\sim 1} \\ \omega_1 + \omega_2 + \omega_3 = 1 \end{cases} \tag{10}$$

where, g denotes the joint cluster evaluation index, ω is the weight of each precision, N is the number of samples, $[*]$ represents the integer of real number *, $\| * \|_{0\sim 1}$ represents the normalization of vector * to [0,1]. CE, PI and SI are three discriminant indexes respectively.

Sorting the elements of the joint cluster evaluation index set $G = [g_1, g_2, \ldots, g_{[\sqrt{N}]}]$ from small to large, the optimal number c_{best} of clustering and the corresponding clustering center matrix V_{best} required by the quadratic clustering are obtained.

4.2.2. Quadratic clustering by KFCM

The ordinary FCM algorithm has high clustering efficiency, but lacks the accuracy of population division. In order to divide the samples accurately into several regions that may contain the local minimum, an improved fuzzy c-means clustering algorithm based on kernel function (KFCM) is proposed in this part, which also had better robustness to noise and isolated points.

Compared with FCM, KFCM has higher cluster partitioning accuracy due to it adopts kernel function as Eq. (11) to map the sample points from the original space to a feature space when the points in the original space cannot be divided by linear function. Suppose $X \in R^D$, defines the mapping from X to the feature space H: $\Phi : X \to H : \Phi(x) = y$.

$$K(x, \tilde{x}) = (y, \tilde{y}) = \| \Phi(x), \Phi(\tilde{x}) \| \tag{11}$$

where, x is the D-dimension vector. The kernel function is used to replace the distance function in the FCM algorithm. The core formula of the KFCM is shown as Eq. (12).

$$\begin{aligned} J_{kfcm}(U, V) &= \sum_{i=1}^{c} \sum_{j=1}^{N} (\mu_{ij})^m \left\| \Phi(x_j) - \Phi(v_i) \right\|_A^2 \\ &= \sum_{i=1}^{c} \sum_{j=1}^{N} (\mu_{ij})^m (2 - 2K(x_j, v_i)) \end{aligned} \tag{12}$$

$K(x_j, v_i)$ is the gaussian radial basis function, in the form as follows:

$$K(x_j, v_i) = e^{-\frac{\|x_j - v_i\|^2}{2\sigma^2}} \tag{13}$$

The iterative formula of U and V is derived by combining the extreme value necessary conditions of Lagrange as follows:

$$u_{ij} = \frac{(1 - K(x_j, v_i))^{\frac{-1}{m-1}}}{\sum_{k=1}^{c} (1 - K(x_j, v_k))^{\frac{-1}{m-1}}}, \; v_i = \frac{\sum_{j=1}^{n} u_{ij} K(x_j, v_j) x_j}{\sum_{j=1}^{n} u_{ij} K(x_j, v_j)} \tag{14}$$

where, $c = c_{best}$ and $V = V_{best}$ are the priori conditions of the cluster quantity and the cluster center. m is the constant coefficient which is usually valued at 2. The precise partition of points in the sample set is completed by minimizing Eq. (12). According to the clustering partition matrix U, multiple clusters are formed. After sorting the sample response mean of each cluster, the top 50% (empirical value) clusters with smaller response mean values are selected as MA subspaces.

In this part, the optimal cluster number and the optimal cluster center coordinates are first determined through cluster verification based on FCM. Then, KFCM method is introduced to divide the samples into multiple clusters, each cluster will contain a few local optimal solutions, from which the MA subspaces and its' boundaries are obtained to provide key regions for subsequent iterative optimization and updating of the SM.

4.3. The determination of IA

Focusing optimization search and sampling on IA subspace that may contain global optimal solutions can improve optimization efficiency. In the initial stage of optimization, the SM may have low precision in some local positions due to the small number of sample points, so the IA determination is to be used every 3 iterations to avoid falling into local optimization as much as possible. The specific operation steps are as follows:

(1) Determine several MA subspaces by quadratic clustering;

(2) Determine the center of the IA subspace according to optimal predictive value which is obtained by sorting the local minimum values in all MA subspaces from small to large;

(3) Calculate the boundaries of IA subspace based on the boundaries of MA subspace where the IA subspace is located and the optimal predicted values. In the case that the MA quantity is greater than 1 or equal to 1, the calculation method of IA boundary $IArange$ is as follows:

$$\begin{cases} IArange = \begin{cases} prebest + 0.7 * (MArange\_up - prebest) \\ prebest - 0.7 * (prebest - MArange\_low) \end{cases} \\ \quad = \begin{cases} 0.7 * MArange\_up + 0.3 * prebest \\ 0.7 * MArange\_low + 0.3 * prebest \end{cases}, if \; size(MA) > 1 \\ IArange = \begin{cases} 0.4 * MArange\_up + 0.6 * prebest \\ 0.4 * MArange\_low + 0.6 * prebest \end{cases}, if \; size(MA) = 1 \end{cases} \tag{15}$$

where, $MArange\_up$ and $MArange\_low$ denote the upper and lower bounds of each dimension of MA subspaces respectively, and $prebest$ is the coordinate vectors of each dimension at the point where the optimal predicted value is located. More samples are collected in the IA subspace to improve the accuracy of the SM in this local region, thus facilitating the capture of the global optimal solution. Moreover, the range of IA subspace is further reduced compared with that of MA subspace. Algorithm1 shows the variable flow process of the IA subspace determination process.

In this part, the design space is divided into three tiered subspaces to provide focus regions for optimizing and updating SM in the following process. The OA subspace provides the ability for avoiding falling into local optimal solution. The code of multi-stage space reduction strategy is shown in Algorithm.1. The Begin section illustrates the parameter initialization of the multi-stage space reduction and the training of initial SM. The Space Reduction section introduces the flow of variable of the MA and IA subspaces determination process respectively.

5. Global optimization based on the surrogate model with the multi-stage space reduction strategy

The three tiered subspaces are determined by multi-stage space reduction strategy to focus the optimization and updating on important region. In order to continuously improve the

Table 3
Fuzzy clustering discriminant index and its characteristics.

Serial	Indicators	Features	Make better
1	Partition coefficient (PC)	The degree of overlap between populations	Close to 1
2	Classification entropy (CE)	Similar to (1)	Close to 0
3	Partition index (PI)	The ratio of similarity to segregation	Low
4	Separation index (SI)	The effectiveness of cluster partitioning	Low
5	Xie and Beni's index (BI)	Similar to (3)	Low

Algorithm 1. Multi-stage space reduction strategy

```
Begin Section
    Initialization: initializes the number of samples m and dimension D, in which the number of samples m recommended is: when the number of dimensions $n=2$, the recommended sampling quantity is $m\in[20,40]$; when $2<n<10$, $m\in[6n,8n]$; when $2<n<10$, $m\in[70,100]$. The number of iterations required for each round of training RBF model is 80, and the initial feasible design space of each dimension is denoted as LHSrange.
    Get the expensive initial sample set:
        Xolh←The sample point matrix with size of $m\times D$ is obtained in LHSrange by using optimized Latin hypercube sampling
        Yexp←The response value of the initial sample set Xolh is calculated by the expensive model, and the matrix size is $m\times 1$
        Sexp←The coordinates of the initial sample points and their corresponding response values are combined to obtain the matrix of the expensive initial sample set with the size of $m\times(D+1)$
    Divide training set and test set:
        Sexp_train and Sexp_test←The initial expensive sample set was divided into training set and validation set by using the left one cross validation method to train the SM with higher accuracy
    Train surrogtae model:
        RBF(Sexp_train, Sexp_test)←Training sets and validation sets are used to train the RBF model, and the accuracy of the SM is improved by cross validation
        save NET_RBF_iter1.mat -mat net←Save the trained proxy model during this iteration
End
Space Reduction Section
    Input: Sexp, NET_RBF_iter1.mat, iter
    Initialization: Set the maximum number of iterations of FCM and KFCM to 100, and the error threshold to 1e-5. The gaussian kernel with the parameter sigma of 150 is used to KFCM method.
    The determination of MA: After using quadratic clustering method to divide the population, s MA are obtained through screening
            Two-stage clustering:
                Cbest←Optimal clustering quantity
                Para_miu_best←The center of each cluster obtained by the first clustering with the optimal clustering quantity
                Label_KFCM←After the cluster partition matrix is obtained, the cluster identity, data coordinates and response are stored in the matrix
        MA: The average response value of the clusters obtained in the previous stage of clustering is calculated, and all the populations were ranked according to the order of average response value from small to large, finally the top 50% populations are taken as $s$ MA subspaces. If the population number is less than 3, only one population is selected as MA.
                MA_plt←Sample points of each MA subspace
                MA_range←The interval boundary range of each MA subspace
    The determination of IA: On the basis of determining $s$ MA subspaces, the hierarchical optimization method is used to determine which space is IA subspace
        MA: After the MA subspaces are determined, the MA_plt and MA_range matrices are obtained
        hierarchical optimization method: The local minimum of each MA subspace are found by hierarchical optimization method
                Local_min_MA←The matrix of local minimum in each transition interval
        IA: Determine the range of the IA subspace by Eq. (15)
                IA_plt←The matrix of sample points located in IA
                IA_best←The predictive optimal solution in IA
                ImpAre_range←The interval boundary range of IA
    Output: MA_plt and MA_range / IA_plt, IA_best and ImpAre_range
End
```

accuracy of the SM by sampling on the proper location in the subspace, this section decoupled optimization search process and surrogate model update process through optimizing efficiently with the ADAM gradient descent in each subspace. Further, the combination sample strategy of extremum method and gaussian process method extremum points are proposed to dynamically update the SM. Finally, the measures to prevent the search process from falling into local optimality are proposed, and the whole operation process of MFCPR-SGO algorithm is demonstrated intuitively.

5.1. Multi-level optimization method with the ADAM gradient descent

In the process of optimization iteration, it is necessary to approach the real global optimal solution by continuously exploring the local minimum in the MA and IA subspaces. Considering that the optimization algorithm needs to adapt to multimoding functions, and be low computational complexity, it must meet the following criteria:

(1) Parameters of the algorithm cannot be set manually during the operation, for ensuring the self-adaptive of the method;

(2) Searching range of the algorithm need be set for restricting the search to a specified interval of MA and IA subspace

(3) Most of the local minimum points in the search space need to be captured by the algorithm.

Compared with the traditional optimization algorithm, the hierarchical optimization method with ADAM gradient descent algorithm has the advantages of simple, accelerated convergence and adaptive adjustment of step size. Moreover, starting from multiple starting points, the algorithm can capture multiple representative local minimum in the search space, the process of which is shown in Algorithm 2.

The MA and IA subspaces and their upper and lower bounds have been obtained by the space reduction strategy. In a certain optimization iteration process, the sample point matrix contained in the MA subspace MA_i is MA_plt_i, the size of the matrix is $K \times (D+2)$, and it contains K observation sample points. The significance of each column in MA_plt_i is: the identification of MA subspace (column 1), D-dimensional coordinates of the sample points (column $2 \sim (D+1)$), and RBF prediction response (column D+2). The observation sample points are taken as the starting points, and the ADAM gradient descent algorithm is used to search along the starting points. Three local minimum points are selected and the matrix *Local_min* is formed after getting rid of redundant results.

In addition, it is necessary to limit the optimal search within the boundary of MA or IA subspaces. The range matrix of a group of MA subspaces is as follows:

$$MA\_range = \left\{ \begin{array}{l} \qquad\quad \begin{array}{cccc} x_1 & x_2 & \dots & x_D \end{array} \\ MA_1 = \begin{bmatrix} 2 & 3 & \dots & 5 \\ -2 & -3 & \dots & -5 \end{bmatrix} \begin{array}{l} upper\ bound \\ lower\ bound \end{array} \\ MA_2 \\ \vdots \\ MA_k \end{array} \right\}_{2\times D\times k} \tag{16}$$

The above equation is a tensor matrix. The three dimensions of the matrix represent the upper and lower bounds, the values of the boundary of each dimension, and the serial numbers of the MA subspaces respectively. Taking the subspace MA_i as an example, in the process of gradient descent, the dimension close to the upper and lower boundary will stay at the boundary, while the other dimension will still search in the direction of negative gradient until it reaches the preset maximum number *step_num* of iterations or satisfies the following convergence conditions as Eq. (17). The detailed process of the ADAM gradient descent will be explained in Appendix D.

$$\|deltaX\|_2 < 1e-8 \tag{17}$$

5.2. Dynamic updating of the surrogate model with multi-layer spatial sampling strategy

The new sample points are the important factor for updating the SM in the optimization iteration process. In this section, a multi-layer spatial sampling strategy based on the combination of extremum method and gaussian error method is proposed to continuously improve the accuracy of the SM on the location of the global optimal solution and the neighborhood where the error is large.

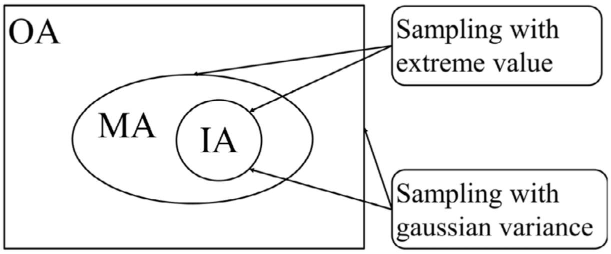


Fig. 3. Which subspaces are affected by the multi-layer spatial sampling method.

5.2.1. Sampling strategy based on the combination of extremum method and gaussian process method

Some of the new sample points obtained by the combination sampling strategy represent the candidate global optimum location, the other part is the location of large prediction errors in MA or IA subspaces. The sampling methods for these three subspaces MA, IA and OA are decided according to their functions respectively as shown in Fig. 3.

(1) Sampling based on extreme value

The candidate points in the overall design space that may be the global optimum are obtained by sampling based on extreme value. As shown in Fig. 3, this method can be used in OA, MA, and IA subspaces. The process of sampling method based on extreme value is as follows:

(a) Determine of the *MA_plt* and *MA_range* of observation sample points contained in the target subspaces (such as MA).

(b) Invoke the hierarchical optimization method to search for all local minimum points in each subspace. The local minimum are sorted from small to large according to the predicted response to obtain the top p local minimum of each subspace. The value of parameter p in this paper is as follows:

$$p = \begin{cases} 3, s = 1 \\ 2, 1 < s < 4 \\ 1, 4 < s \end{cases} \tag{18}$$

where, s is the number of subspaces generated in this iteration.

(c) Remove redundant sample points to obtain the new sample set

Searching from multiple starting points to obtain the local minimum by using the hierarchical optimization method may eventually converge to the same location, which is the first kind of redundancy of the new sample point set. In addition, the new sample points obtained may coincide with some points in the original sample set, which is the second kind of redundancy. So the following means are taken to eliminate the redundancy: (1) Calculate Euclidean distance of each point in the new sample points set, and eliminate the new sample points with small distances; (2) Calculate the Euclidean distance between the new sample set and the original observed sample set one by one, and eliminate the redundant points. The evaluation formula of Euclidean distance is as follows:

$$\|x_1 - x_2\|_2 < \|LHSrange(1, :) - LHSrange(2, :)\|_2 * (1e-5) \tag{19}$$

where, $\|*\|_2$ is L2 norm of the vector *, x_1, x_2 are two sample points, $LHSrange(1, :)$ and $LHSrange(2, :)$ are the upper and lower bounds of each dimension of the initial design space respectively.

Algorithm 2. Multi-stage optimization method using ADAM gradient descent

```
The hierarchical optimization method with ADAM
Initialization: learning rate $\eta$, maximum iteration step size, initial parameters, decay rate
    Input:
        MA_plt and MA_range / IA_plt and ImpAre_range ← the sample points contained in the MA or IA subspace and the corresponding subspace range, assuming that a subspace contains $s$ sample points
        NET_RBF_iter.mat ← RBF model for this iteration training
    While    iter_opt < step_num ← set the maximum number of iteration to step_num
        grad ← calculates the numerical gradient
        Square gradient ← cumulative Square gradient
        Update parameter
        deltaX = -grad * $\eta$ ← calculate the increment of each dimensional coordinate of the sample point in this iteration
        $X_{iter\_opt+1}$ = $X_{iter\_opt}$+deltaX ← update the coordinates of the points
        If    $\|deltaX\|_2 < 1e-8$ ← if all dimension increments are less than 1-e8, terminate the loop
            Break;
        End if
    End while
    Output: Local_min ← get the local minimum matrix of size $S \times (D+1)$
End
```

(2)Sampling based on gaussian process error

The sampling method based on gaussian process error can capture the sample points where the SM has large error during the current iteration. As shown in Fig. 3, this method is used in IA and OA subspaces.

The conventional error evaluation method requires a large number of samples due to the needed error calculation for a certain position by comparing the difference between the predicted value and the observed one, and is not fitted for the question in this paper. So the Gaussian process error method is introduced to solve this problem, which consists of two parts: (1) calculate Gaussian process error; (2) obtain the deviation between gaussian process model (GPM) and RBF model.

Calculate the gaussian process errors of each point within the spatial range after determining the *IA_plt* and *IA_range* of the observation sample points contained in the target subspace (such as IA). The calculation of gaussian process error consists of four steps:

(a) Train the Gaussian process regression model *IA_grf* in the subspace by using the sample set *IA_plt*, and obtain the predicted values *IA_grf _pre* and variance σ of gaussian process everywhere in the IA subspace;

(b) Calculate the predicted response deviations $\Delta pre = |IA\_GRF\_pre - IA\_RBF\_pre|$ of GRF and RBF everywhere in the IA subspace;

(c) Normalize the variance of the gaussian process and the deviation of predicted response according to Eq. (20), and make the data to the same scale;

$$\begin{aligned}\Delta pre_{nor} &= \Delta pre \odot \frac{1}{\max(\Delta pre) - \min(\Delta pre)}\\ \sigma_{nor} &= \sigma_{nor} \odot \frac{1}{\max(\sigma_{nor}) - \min(\sigma_{nor})}\end{aligned} \tag{20}$$

(d) Weight the standardized variance and the deviation of predicted response to obtain the total error value $SE = 0.2\Delta pre_{nor} + 0.8\sigma_{nor}$.

After the total error value is calculated, the points are sorted from large to small according to the total error, and the top three points (with the three largest total errors) are selected as the new sample points. The relevant contents of gaussian process variance and gaussian process regression is explained in detail in Appendix B.

5.2.2. The process of multi-layer spatial sampling strategy

The alternatively using of the subspaces in the optimization iterative process is as Algorithm 3. And how to control the use of subspaces in iterative process to ensure the fast and efficient convergence of the algorithm is a problem worthy of elaboration.

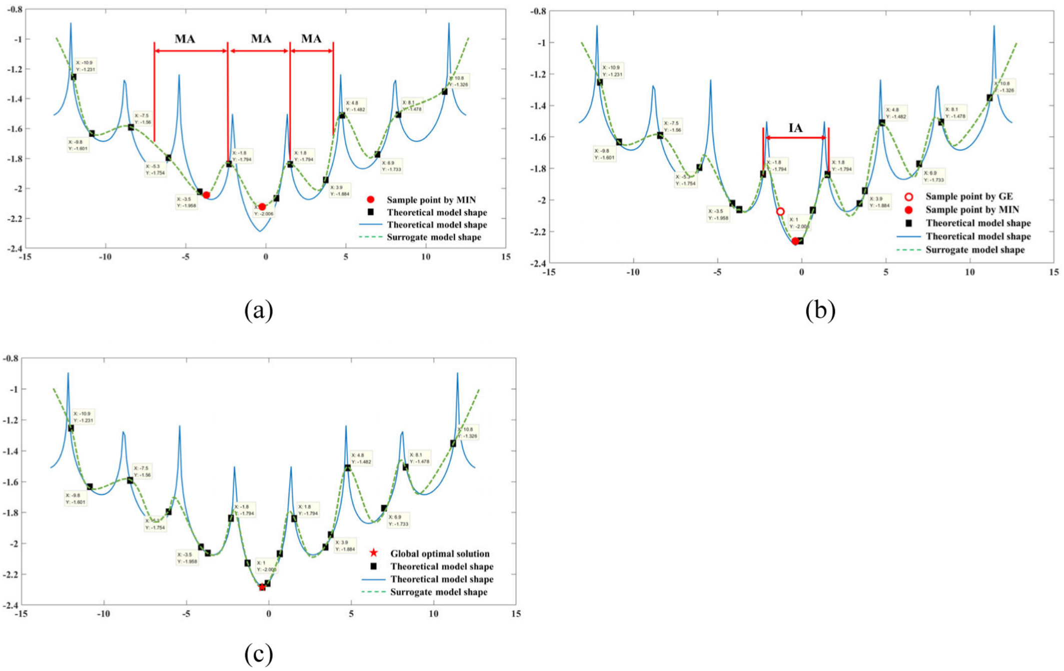
(1) The function of MA subspace is to reduce the design space initially and facilities the optimization focusing the region on all local minimum neighborhoods. If the number of optimization iterations is not an integer multiple of three, Quadratic clustering as mentioned above is used to generate multiple MA subspaces. And optimization search is performed in those MA subspaces to obtain new sample points and update the SM, and further take the boundary of those MA subspaces as the boundary of benchmark space *BS_range* for next iteration;

(2) The function of IA subspace is to significantly reduce the design space and focus on the global optimal location. In order to prevent excessive convergence, IA subspace is generated once every three iterations. To prevent IA shrinking to a point, the benchmark space *BS_range* of the next iteration will be expanded according to the boundary of the global design space and IA subspace, as follows:

$$BSrange(k) = \begin{cases} IArange\_up(k) * (1 + \dfrac{1}{LHSrange\_up(k)}) \\ IArange\_low(k) * (1 + \dfrac{1}{LHSrange\_low(k)}) \end{cases}, i = 1, 2, \ldots, D \tag{21}$$

where, *LHSrange* is the initial design space, *D* is the design variable dimension, *up* and *low* represents the upper and lower bounds of each dimension of the space respectively.

Taking a simple two-dimensional optimization problem in Fig. 4 as an example, the initial SM in Fig. 4a is not very accurate, but it can roughly reflect the global trend of the curve. By using quadratic clustering method to determine the MA subspaces and updating the SM with new sample points (red dots) which are obtained by extremum method in the MA subspaces, the accuracy of SM is improved effectively. Fig. 4b shows the schematic diagram of the SM and IA subspace in a certain iteration. In this iteration, two new sample points are obtained by using the multi-layer spatial sampling strategy, which respectively represent the predicted global optimal solution and the position with large error. After updating the SM, as shown in Fig. 4c, the SM curve



(a)

(b)

(c)

Fig. 4. Elaboration of the SM by multi-layer spatial sampling strategy during iterative optimization.

nearby the global optimal position is basically coincides with the real curve.

5.2.3. The strategy of jumping out of the local optimal solution

In order to avoid the algorithm falling into local optimal solution, this paper combines sampling strategy with extremum and gaussian error into OA subspace to update the SM and adjust the benchmark space. The process is divided into following four steps:

(1) Judging whether the current iteration process is trapped in local optimization. The judgment is made based on the two rules as shown in Algorithm 3: (a) whether the number of new sample points is zero, if so, it is difficult for the current iteration to produce more diversified new sample points; (b) whether the observed optimal value does not change in 5 consecutive iterations, if so, the algorithm cannot get better convergence. Then next iteration should enter the OA subspace and take the sparse sampling procedure.

(2) Sampling by gaussian error method: the sparse sample area is explored through gaussian error, and the new sample points with large predicted response errors are obtained;

(3) Sampling by extremum value: hierarchical optimization is applied to search in OA space from multiple starting points. Further, comparing with the current predicted optimal solution to determine whether to extend the benchmark space. The detailed implementation steps are as Algorithm 3: the initial sample set obtained by ESEA-OLHD method is added to the new sample set to obtain the local minimum in P OA space by using the hierarchical optimization algorithm to update the SM.

(4) Updating the benchmark space: update the predicted optimal solution if the following relaxation discrimination Eq. (22) is satisfied, and merge the positions of *BS_range* and *OA_best* to update the next iteration benchmark space.

$$OA\_best - pre\_best(iter) < \frac{pre\_best(iter)}{pre\_best(1)} \tag{22}$$

where, *OA_best* is the local minimum and *pre_best*(*iter*) is the current predicted optimal solution.

5.3. Overall optimization process

The overall flow of surrogate-adjoint refine based global optimization method combining with multi-stage fuzzy clustering space reduction strategy is shown in Fig. 5, which consists of four parts: (1) initialize the method before entering the loop iteration; (2) determine the three subspaces by after entering the loop iteration, and carry out the optimization search from the three subspaces to obtain new sample points; (3) judge and explore the sparse space to jump out of the local optimal solution and determine the new sample set; (4) update the benchmark space and surrogate model and determine whether the termination condition is met.

The part of initiation is as follows:

(1) Determine the design variables included in the optimization problem, the initial design space and the expensive simulation model are needed to calculate the observation response, and initialize the parameters of each method;

(2) Generate initial sample points in the given original design space by ESEA-OLHD method;

(3) Calculate the target value Y_{exp} of each point in the initial DOE sample set X_{olh} by using the expensive model and the objective function, and combine the dimensional coordinates and observed values of the sample points to form the initial observation sample point set S_{exp};

The part of multi-stage space reduction is as follows:

(4) Enter the cyclic iteration process, and record the iteration number iter. When iter>1, calculate the target values of each point in the last iteration of the new sample set, add the new sample set *Sample_new* to the total sample set S_{exp}^{iter-1}, and update the RBF model with the total sample set S_{exp}^{iter-1}. Meanwhile, the benchmark space *BS_range* of this iteration is determined.

(5) Cluster the sample points in the sample set S_{exp} for the first time, and the optimal clustering number C_{best} and the coordinates of each clustering center *Para_miu_best* are determined by means of population verification.

(6) Divide all sample points in the expensive sample set into C_{best} populations by the quadratic clustering algorithm KFCM after the optimal cluster number and the corresponding central

Algorithm 3. Optimization flow and escape from local optima

```
Overall iterative flow
While   iter<itermax←enter the loop, itermax is the maximum number of iterations
    Iter=Iter++
    If    mod (iter, 3)==0←whether the current iteration iter is an integer multiple of 3
        IA_plt, IA_best, BS_range←perform the multi-stage space reduction algorithm to obtain the IA subspace,
        the next iteration benchmark space and the included sample points
        Sample_best←performs a sampling strategy based on the extremum method to obtain additional sample
        points that may contain global optima
        Sample_gua←implements the gaussian error sampling strategy to obtain new sample points for the larger
        error region in the IA subspace
        Sample_new←after combining Sample_best and Sample_gua, performs the sample point filtering method,
        remove the samples that are too close to each other inside and outside, and get the final new sample set
    Else
        MA_plt, BS_range←perform the multi-layer space reduction algorithm to obtain the MA
        Sample_best←performs the sampling method based on extreme values
        Sample_new←performs sample point filtering
    Identify and execute the unknown area exploration section
    If    iter>5←after 5 iterations, calculate the difference CRT between the current optimal predictive value and the
    previous 5 optimal predictive values
        CRT=Pre_best(iter)-Pre_best(iter-5)
    Else
        CRT=1e10
    End if
    If    size(Sample_new)=0 or CRT<1e-6←whether the number of new sample points is 0 or the CRT is less than
    the threshold
        Perform the sparse sampling method to sample in OA subspace
       Sample_OA←performs gaussian error sampling in the OA subspace to generate new sample points
       OA_best←with multi-point as the starting point, the hierarchical optimization method is used to determine the
       optimal predictive value in the OA subspace
        Sample_new=[Sample_new; Sample_OA]←merge the new sample points
        If    OA_best-pre_best(iter)<pre_best(iter)/pre_best(1)
            Pre_best(iter)=OA_best←update the optimal predictive value
            BS_range←expand and update the benchmark space for the next iteration
            Sample_new=[Sample_new; Sample_OA; OA_best]←merge the new sample points
        End if
    End if
    Sample_new_rdc←performs the de-redundant operation on the newly added set of sample points to generate a
    non-redundant sample matrix
    Ysample_new←calculates the observed value of the new sample points through the expensive model
    Sexp←[Sexp;[Sample_new_rdc,Ysample_new]] combines the new sample coordinates and responses into the
    overall sample set
    NET_RBF_iter+1.mat←update the RBF model
End While
Out put: pre_best, RBF←outputs the global optimal solution and the final RBF model
```

coordinates of each cluster are obtained in the previous step, each population containing an uncertain number of sample points;

(7) Obtain the minimum observation samples contained in each population, rank them from small to large according to the observed values, and select the top 20% populations as s MA subspaces. Correspondingly. The observation sample matrix subordinated to the s MA subspaces is *MA_plt*.

(8) Apply the hierarchical optimization method to s MA subspaces, optimize the RBF model of the current iteration, find the local optimal solution in each MA subspace, and generate the matrix *Local_min*.

(9) Check if the condition $Mod\,(iter, 3) = 0$ is satisfied, if so go to the step 11. Otherwise, go to the next step. Where, mod (*) represents the formula for calculating the remainder.

(10) Take the local minimum *Local_min* in each MA subspace as the new sample points, implement the sample points screening method, eliminate the sample points that are close to each other, and generate the sample point set *Sample_new*, go to step 13;

(11) Order the local minimum values in all MA subspaces to obtain the optimal predicted value Pre_best^{iter}. If the MA quantity is greater than 1, the corresponding MA subspace of *Pre_best* is used as the IA subspace. If the quantity of MA is equal to 1, then the boundary of IA is determined by the Eq. (15);

(12) Use gaussian error sampling method within the range *IA_range*. The gaussian variance and gaussian process regression deviation are weighted to obtain the errors everywhere in the subspace. The top three points with largest errors are selected as the new sample points and merged with the best prediction point *Pre_best*. Perform the sample point screening method, remove the sample points that are close to each other, and obtain the new sample point set *Sample_new*;

The part of exploration of sparse region is as follows:

(13) Check if the condition $size\left(Sample\_new^{iter}\right) = 0orCRT < 1e{-}6$ is satisfied, if so go to next step. Otherwise, go to step 15

(14) Execute the sparse region exploration program, update the predicted optimal solution *Pre_best*, the next iteration of

Table 4
Description of benchmark numerical optimization problem.

Category	Function	Dimensions	Design space	The analytical solutions
Low-dimensional problems (n = 2-8)	BR	2	[−5, 10]×[0, 15]	0.3979
	SC	2	[−2, 2]2	−1.0316
	CT	2	[−2, 2]2	−2.0626
	GF	2	[−2, 2]2	0.5233
	HM	2	[−6, 6]2	0.0000
	Leon	2	[−10, 10]2	0.0000
	Shekel	4	[0, 10]4	−10.1532
	Levy	4	[−10, 10]4	0.0000
	Trid6	6	[−36, 36]6	−50.0000
High-dimensional problems (n > = 10)	Sphere	10	[−5.12, 5.12]10	0.0000
	Sums	15	[−5, 5]15	0.0000
	F16	16	[−1, 1]16	25.8750

the benchmark space *BS_range*, and the new sample point set *Sample_new*;

The part of termination is as follows:

(14) Calculate the observed value of the newly added sample points by using the expensive model to obtain *Sexp*, and update the RBF model by using the new sample set;

(15) Check if the termination condition is met, if so stop the whole loop, and output the final RBF model and the global optimal solution. Otherwise, repeat steps 3–14 until the termination condition is met.

The termination conditions used in this method are as follows:

$$\begin{cases} \dfrac{\left|y_{best}-y_{opt}\right|}{\left|y_{opt}\right|}\leq 0.01, \quad if\ \ y_{opt}\neq 0 \\ y_{best}<0.001, \quad if\ \ y_{opt}=0 \\ NFE>300 \\ iter>iter\max=200 \end{cases} \tag{23}$$

where, $y_{opt} = pre\_best$ is the optimal predicted value obtained during this iteration, and y_{opt} is the theoretical optimal solution of the numerical problem. The NFE is the number of calculations using the expensive model, and iter is the number of iterations.

6. Numerical verification

The benchmark test problems in 12 different dimensions are used to verify the significance of the proposed method. Among them, the low-dimensional function (2∼8 dimensions) [42] includes: BR, SC, CT, GF, HM, Leon, Shekel, Levy, Trid6; higher-dimensional functions include (10∼16 dimensions) [43,44]: Sphere, Sums, F16. Table 4 shows the details of these test functions, including design variable dimensions, initial design space, theoretical optimal solution, etc. Detailed formulas for all test functions are explained in Appendix E. The parameter values of each sub-method in MFCPR-SGO are set as shown in Table 5.

6.1. Preliminary comparison and testing

The MFCPR-SGO algorithm in this paper is compared with other well-known traditional global optimization algorithms (HS, DE, DIRECT). it is proved that MFCPR-SGO algorithm has obvious advantages in reducing the number of expensive function calls.

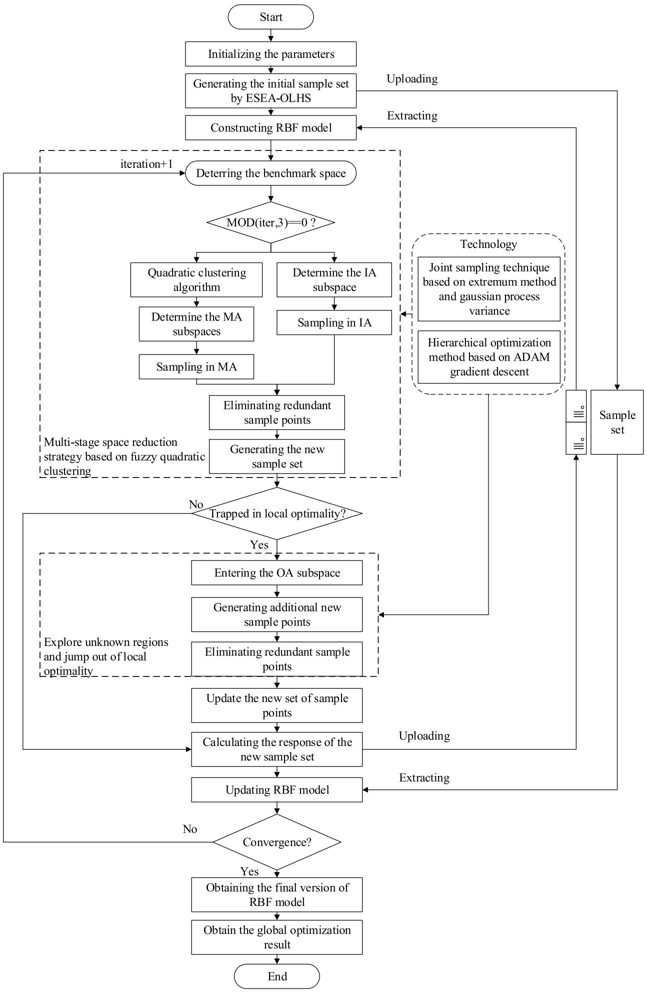

Start
Initializing the parameters
Generating the initial sample set by ESEA-OLHS
Uploading
Extracting
Constructing RBF model
iteration+1
Deterring the benchmark space
MOD(iter,3)==0 ?
Quadratic clustering algorithm
Determine the IA subspace
Determine the MA subspaces
Sampling in IA
Sampling in MA
Eliminating redundant sample points
Generating the new sample set
Multi-stage space reduction strategy based on fuzzy quadratic clustering
Technology
Joint sampling technique based on extremum method and gaussian process variance
Hierarchical optimization method based on ADAM gradient descent
Sample set
No
Trapped in local optimality?
Yes
Entering the OA subspace
Generating additional new sample points
Eliminating redundant sample points
Explore unknown regions and jump out of local optimality
Update the new set of sample points
Calculating the response of the new sample set
Uploading
Updating RBF model
Extracting
No
Convergence?
Yes
Obtaining the final version of RBF model
Obtain the global optimization result
End

Fig. 5. The process of surrogate model construction and optimization.

Then, the algorithm is compared with another recently proposed global optimization algorithm based on space reduction (HAM ESGO SEUMRE).

Table 5
The initial values of each sub-method parameter in MFCPR-SGO.

Iter_max	*NFE*	*iter_RBF*	*neurons*	*expRBF*	*iter_fcm*	*sigma*	*alpha*	*epsKFCM*	*iter_opt*	*epsCRT*
100	300	50	80	0	100	150	2	1e−5	100	1e−6

Table 6
The average value of the global optimal solution and the NFE of 10 optimization results.

Fun.	HS		DE		DIRECT		MFCPR-SGO	
	Min	*NFE*	*Min*	*NFE*	*Min*	*NFE*	*Min*	*NFE*
BR	0.4021	9122	0.3992	1390	0.3984	603	**0.3979**	**15**
SC	−1.0276	310	−1.0299	450	−1.0248	117	−**1.1064**	**20.7**
CT	−2.0532	512	−1.9881	830	−2.0382	101	−**2.0531**	**36.3**
GF	0.6247	450	0.5461	3070	**0.5238**	2883	**0.5238**	**43.1**
Shekel	−2.6829	10000	−10.0930	3730	−**10.0934**	340	−9.7992	**230**
Trid6	−45.5191	698	−47.5219	3660	−49.2154	213	−**49.9734**	**129**
F16	26.1207	915	26.1022	3690	26.0884	6439	**25.9964**	**156**

Considering that the uncertainty in the operation of the algorithm may make the results fluctuate, the seven optimization methods are tested for 10 times for each benchmark numerical case. At the same time, in order to compare the algorithms in efficiency and accuracy, this article uses the following two indexes: (1) The number of function evaluations (NFE) [45], which refers to the invoked times of expensive function. It takes up most of the time consumed in optimization algorithm. In this problem, the smaller NFE, the less time it would take, and the higher the convergence efficiency of the algorithm is. (2) the accuracy of the global optimal solution *prebest*, which reflects the accuracy of the optimization results.

Firstly, this algorithm is compared with other three well-known global optimization algorithms, in which, harmonious search (HS) [46] and differential evolution (DE) [47] algorithms are natural heuristic global optimization methods. DIRECT [48] is an effective space reduction algorithm. The maximum allowable NFE for all problems is 300. The parameters of each method in the algorithm are set in Table 5. The ranges and averages of the global optimal solutions f_{opt} and NFE obtained are shown in Table 6.

In the table, the NFE values with symbols > indicate that in the 10 times of optimization solution, at least one test cannot satisfy the convergence condition within 300 times of NFE, and the number of failed searches is shown in the brackets. In this paper, the smaller average values in fopt and NFE are marked in bold.

Fig. 6 shows the convergence curve and NFE growth curve of MFCPR-SGO algorithm in four typical test functions of Trid6, Sphere, Sums and F16 dimensions. It can be seen from the subgraph of Fig. 6(a), (b) and (c) that the algorithm takes three iterations as a cycle, and the target value of the predicted optimal solution obtained every three iterations will drop significantly, finally it can converge to the global optimal solution stably. This shows that the multi-stage space reduction strategy of alternate optimization in OA, MA and IA subspaces can make the algorithm converge to the global optimal solution quickly. In Fig. 6(d), the 4-10th iteration and the 14-24th iteration curves both fall into the local optimal solution, but after several iterations, the algorithm successfully jumps out of the local optimal solution and finally converges to the global optimal solution. This shows that the combined sampling strategy based on gaussian error method and extremum method in OA subspace can effectively help the algorithm to get rid of local optimization.

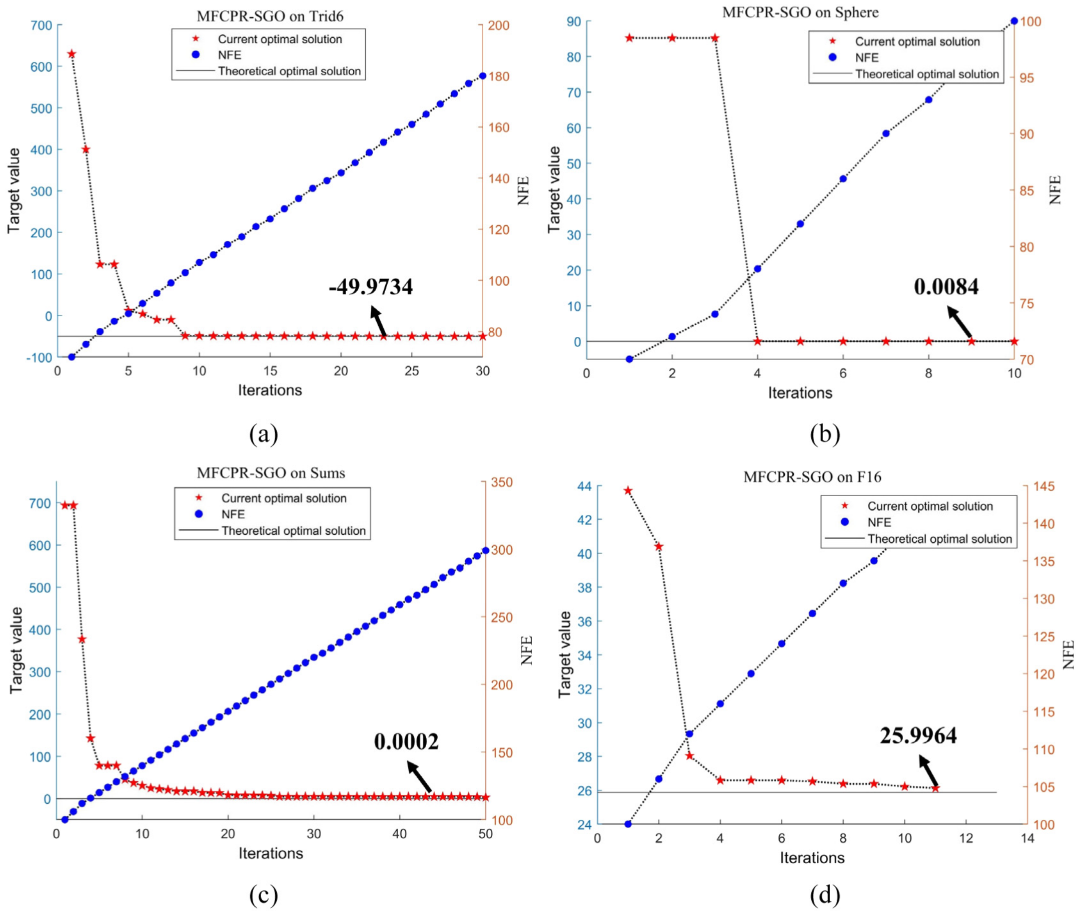
Table 6 shows the results of SM construction and global optimization of HS, DE, DIRECT and MFCPR-SGO algorithms for 7 benchmark numerical cases. Each numerical case is optimized for 10 times, and the lower bound, upper bound and average value of the global optimal value solved for 10 times are calculated. According to the results in Table 6, the following conclusions can be drawn: (1) the HS and DE algorithms require a high NFE in most problems; (2) DIRECT algorithm performs well in 2–4 dimensional test functions, but this method is not applicable to high-dimensional functions such as F16; (3) the proposed MFCPR-SGO algorithm shows good performance in most test cases except Shekel, and it uses less NFE than other methods, which indicates that the multi-stage space reduction strategy can indeed reduce the frequency of the algorithm to call expensive functions.

In summary, since non-surrogate optimization methods directly invoke the expensive model to calculate the target value during the search, large NFE are often required. The surrogate-based optimization method uses the surrogate model to guide the exploration of design space, effectively reducing the NFE, and the multi-stage space reduction strategy in this paper further expands this advantage. Compared with the nature-inspired global optimization method, the new MFCPR-SGO method has advantages in the optimization problem with a large amount of computation.

6.2. Further testing and comparison

In order to further demonstrate the advantages of the MFCPR-SGO algorithm over other similar methods, comparison with another three effective spatial reduction algorithms HAM, SEUMRE and ESGO [49] is conducted.

HAM uses three types of surrogate models to fit the design space: Kriging, RBF, and PRS. This strategy effectively combines the characteristics of multiple SM. However, it is likely to converge to local optimality once these samples are concentrated in the same region.SEUMRE uses grid sampling to divide the design space into multiple unimodal regions. After obtaining the key unimodal region where optimization and reconstruction are mainly carried out, the KRG model was established by the sample points. However, the algorithm may fall into the local optimality of multimodal functions or high-dimensional problems due to the initial processing of experimental data may miss the true global optimal position. Although SEUMRE algorithm has high



(a)

(b)

(c)

(d)

Fig. 6. The optimization curve of MFCPR-SGO in 6–16-dimensional test cases.

Table 7
The minimum value, maximum value and average value of optimal solution 10 results.

Fun.	HAM		ESGO		SEUMRE		MFCPR-SGO	
	Min and max	*Median*	*Min and max*	*Median*	*Min and max*	*Median*	*Min and max*	*Median*
BR	[0.3979, 0.4014]	0.3993	[0.3979, 0.4019]	0.3990	[0.3982, 0.4009]	0.3991	[0.3979,0.3990]	**0.3979**
SC	[−1.0316, −1.0221]	−1.0279	[−1.0310, −1.0222]	−1.0257	[−1.0314, −1.0213]	−1.0268	[−1.4270,−1.0253]	−**1.1064**
CT	[−2.0625, −2.0458]	−**2.0547**	[3.0053, 3.0284]	3.0174	[−2.0576, −2.0429]	−2.0503	[−2.0623,−2.0421]	−2.0531
GF	[0.5236, 0.5278]	0.5260	[3.0053, 3.0284]	3.0174	[0.5248, 0.5296]	0.5263	[0.5234,0.5241]	**0.5238**
HM	[0.0001, 0.0087]	0.0035	[0.0002, 0.0094]	0.0032	[0.0011, 0.0754]	0.0134	[0.0000,0.0008]	**0.0003**
Leon	[1.2102, 9.5885]	5.2184	[1.3180, 2.8752]	2.2528	[1.12e−4, 0.3207]	0.2831	[4.5e−3,1.3e−2]	**9.5e−3**
Shekel	[−10.1472, −2.6166]	−5.1562	[−7.8920, −3.3575]	−7.5345	[−10.0546, −2.6303]	−5.1572	[−9.8051, −9.6158]	**-9.7992**
Levy	[2.96, 20.26]	8.7925	[0.0092, 0.2517]	0.0158	[6.63e−4, 0.1103]	0.0529	[4.21e−4,5.52e−3]	**0.0013**
Trid6	[−49.8777,−42.8814]	−48.9573	[−49.1002, −32.5320]	−45.5183	[−47.5255, −7.9626]	−16.3528	[−49.9550, −49.9910]	**-49.9734**
Sphere	[4.20e−3, 0.1847]	0.1102	[0.1002, 2.1391]	0.8750	[1.8147, 17.2568]	10.2375	[1.20e−4, 3.13e−4]	**2.23e−4**
Sums	[1.45e−3, 3.191]	1.0145	[0.1258, 2.8424]	0.8975	[2.1584, 15.1285]	4.1285	[0.0842,4.1094]	2.2151
F16	[26.0264, 6.1879]	26.1190	[25.9754, 26.1350]	26.0485	[28.5612, 34.2704]	30.9123	[25.9782, 26.1284]	**25.9964**

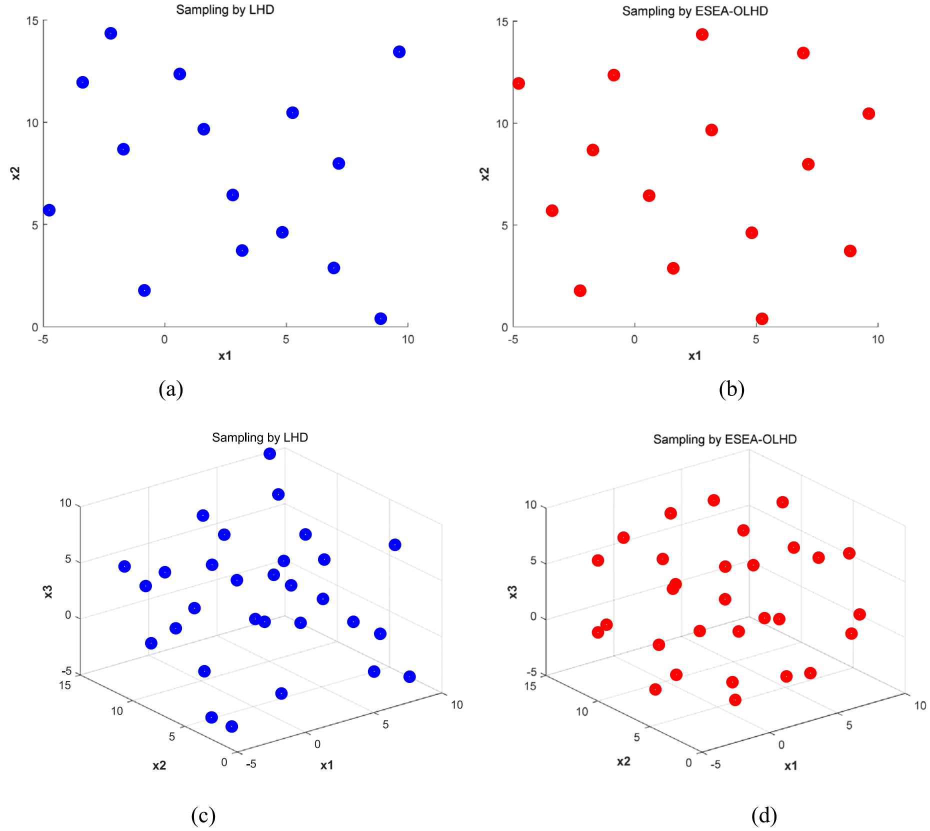
search convergence efficiency, it is unstable for complex global optimization problems.

Table 7 shows the results of HAM, ESOG, SEUMRE and MFCPR-SGO algorithm in the surrogate model construction and optimization of 12 benchmark numerical cases. For each numerical case, the lower limit, upper limit and average value of 10 optimization results are obtained. Table 8 gives the minimum, maximum and average NFE for 10 calculations, respectively.

It can be seen that HAM, ESGO and SEUMRE algorithms perform well in SC, BR, GP and GF functions, and can accurately find the global optimal solution in multi-mode low-dimensional problems such as CT. However, they are hard to solve multi-mode

Table 8
The minimum value, maximum value and average value of NFE in 10 results.

Fun.	HAM		ESGO		SEUMRE		MFCPR-SGO	
	Min and max	*Median*	*Min and max*	*Median*	*Min and max*	*Median*	*Min and max*	*Median*
BR	[21, 66]	34.1	[25, 52]	40.1	[44, 79]	54.5	[15,43]	**27**
SC	[26, 52]	35.8	[25, 50]	37.5	[41, 58]	49.3	[16,29]	**20.7**
CT	[15, 102]	35.8	[16, 206]	67.1	[15, 45]	**22.3**	[16,72]	36.3
GF	[57, 258]	123.5	[34, 91]	57.2	[65, >300]	>103.5(1)	[29,55]	**43.1**
HM	[20, 89]	62.4	[27, 130]	75.1	[58, >300]	>150.9(3)	[38,59]	**42.4**
Leon	[102, >300]	>233(6)	[91, 260]	**192**	[142, 300]	>194(2)	[150, 243]	203
Shekel	[269, >300]	>300(8)	[1905, >300]	>300(4)	[217, >300]	>300(9)	[200, 261]	**230**
Levy	[115, >300]	>300(7)	[170, >300]	>300(2)	[119, >300]	>300(6)	[105, 230]	**201**
Trid6	[67, >300]	>181.8(4)	[52, 80]	**66.8**	[125, >300]	>259.7(5)	[50,109]	79
Sphere	[106, 144]	130	[95, 270]	195	[>300, >300]	>300(10)	[118,133]	**129**
Sums	[>300, >300]	>300(10)	[180, >300]	>218(4)	[>300, >300]	>300(10)	[85, 140]	**115**
F16	[>300, >300]	>300(10)	[75, >300]	>275.8(9)	[>300, >300]	>300(10)	[170, >300]	>**227(4)**



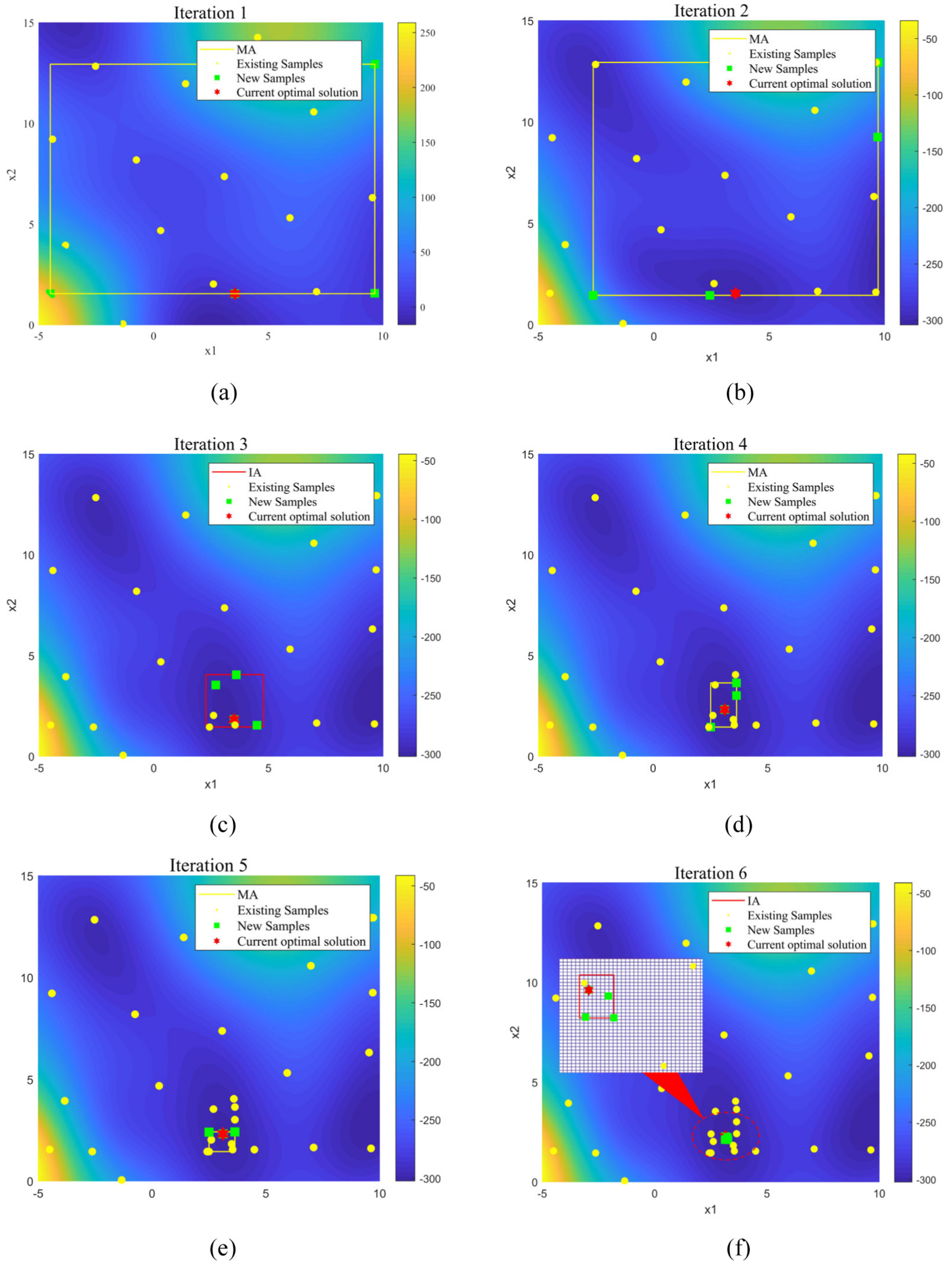
(a) (b)

(c) (d)

Fig. 7. The sample sets were obtained by using ordinary LHS and ESEA-OLHD respectively.

high-dimensional problems. This is reflected in that SEUMRE has a 50% probability of successfully finding the optimal solution in 300 NFE in the test of Shekel and Levy functions, and has almost no chance to find the optimal solution in medium and high dimensional multi-mode functions such as Trid6, Sphere, Trid10 and F16. HAM performs better in higher dimensions problems than SEUMRE, but worse in multi-mode functions such as Shekel and Levy. ESGO is an effective algorithm, because when dealing with the high-dimensional problems, EGO consumes much less NFE than HAM and SEUMRE, and has a greater probability to find the global optimal within 300 NFE. However, as shown in Table 7, ESGO algorithm is weaker than MFCPR-SGO algorithm in finding the global optimal solution for most problems.

Generally, HAM sometimes falls into local optimality. And SEUMRE method is not effective in multi-mode and high-dimensional optimization. While ESGO method has poor performance in multi-mode problems and generally high NFE. In



(a) (b)

(c) (d)

(e) (f)

Fig. 8. Space reduction during iteration in BR problem.

contrast, the MFCPR-SGO algorithm handles most 2–6 dimensional functions and all 10–16 dimensional functions well, but more NFE is used when deal with CT and Leon. In most cases, MFCPR-SGO still has a higher convergence accuracy and a lower NFE on average in multiple calculations than the other three algorithms, which means that the algorithm is more efficient in dealing with complex computational models.

6.3. Demonstration of MFCPR-SGO operation process

Operation process of the algorithm in solving the BR numerical optimization problem is used to demonstrate the convergence process and advantages of MFCPR-SGO method intuitively. Fig. 7 shows the sample sets obtained in 2-D and 3-D design space respectively by using ordinary LHS method and ESEA-OLHS optimization method. Fig. 7 shows the changes of MA and IA subspaces, the predicted global optimal solution for each iteration. Fig. 8 shows the process of searching the local optimal solution in MA subspaces using ADAM gradient descent method.

As shown in Fig. 7, two groups of samples were collected: (1) in the two-dimensional initial design space, the number of samples is 15; (2) in the three-dimensional initial design space, the number of samples is 30. Fig. 7 a and Fig. 7 c respectively represent the sample point sets obtained by sampling on (1) and

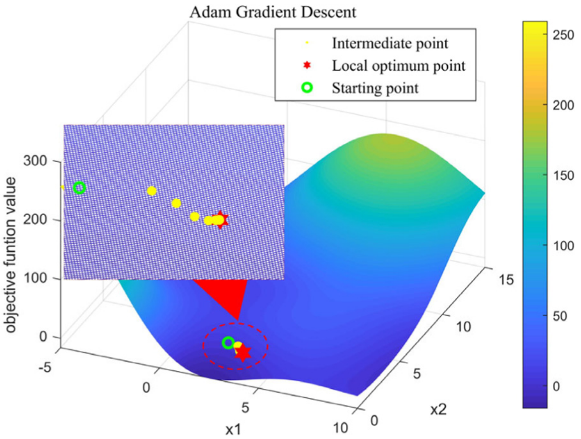


Fig. 9. Multi-layer optimization method is used in iterative process.

(2) using the ordinary LHS test design method, while Fig. 7 c and Fig. 7d represent the sample point sets obtained by sampling on (1) and (2) using the ESEA-OLHD optimized test design method. In the two experimental designs, the spatial filling index of the sample set decreased from 0.82 to 0.37 and from 0.89 to 0.41 respectively. Fig. 7 indicates that to achieve the same level of space filling, ESEA-OLHD method uses less sample size. The surrogate model trained with high space filling sample can reflect the characteristics of the problem more accurately and comprehensively.

As shown in Fig. 8, the algorithm takes every 3 iterations as a cycle, and generates the MA subspaces in the previous 2 iterations to achieve the initial reduction of space. After sampling in MA subspaces and updating the surrogate model, IA subspace is used to further shrink the space to the position where the global optimal solution may exist in the third iteration. As shown in the enlarged area in Fig. 8(f), green dots are newly added sample points, representing the optimal prediction and the position with large error in the red box (IA subspace). Compared with HAM which only uses one sampling strategy, this method focuses the search process and surrogate model update process on the MA and IA subspaces (yellow and red wireframe in Fig. 8), it not only accelerates the convergence, but also saves more expensive sample points compared with the global sampling algorithm such as HS. In this example, only 12 iterations and 42 expensive sampling points are needed to find the global optimum.

Fig. 9 shows the process of rapidly finding local minimum in MA subspace in the fifth iteration of BR problem by using the multi-layer optimization method based on ADAM gradient descent. In the fifth iteration, the design space is effectively contracted, so only one local minimum exists in the subspace. On the contrary, in the first two iterations, 3∼4 local minimum can be found in the subspaces. According to the data, compared with the MATLAB GA algorithm, the time spent in a single operation with the help of multi-layer optimization method based on ADAM gradient descent is reduced from 0.021 s to 0.003 s.

7. Conclusions

In this paper, an surrogate-based global optimization method with space reduction strategy is proposed to build a fusion mechanism of the optimization iterative process and the dynamic updating process of the surrogate model, which greatly reduces the demand of sample size and provides a better solution to the complex black box global optimization problem. The main contributions of MFCPR-SGO algorithm proposed in this paper are as follows:

(1) an optimized Latin cube sampling method based on enhanced random evolution random algorithm (ESEA-LOHD) was proposed. The algorithm is divided into inner and outer loops. Firstly, a random experimental design matrix is generated by using the column element exchange method in the inner loop. After that, the uniformity of experimental design samples is measured by three decision criteria. In the outer loop, the two strategies of improvement and exploration are used to control the algorithm switch between jumping out of the local optimal region and exploring the global optimal solution. This strategy provides an initial sample set with higher spatial uniformity and better reflection of the overall characteristics for the training and updating of the surrogate model.

(2) a multi-stage space reduction strategy with fuzzy clustering is introduced. The strategy divides the initial design space into three tier subspaces: OA, MA and IA. OA represents the initial global design space, which remains unchanged in the iterative search process. MA and IA subspaces are contracted by corresponding strategies in the iterative process, and the boundary and coordinate of these two subspaces change with the capture of local optimal solution and the update of surrogate model. Fuzzy clustering algorithm is used to determine the neighborhood where the surrogate model may have local optimal solutions in the design space defined as MA subspace, so the search range is narrowed from OA subspace to MA subspace. Further,IA subspace acceleration algorithm is used to search the more important local design space. Through this strategy, the search scope is gradually narrowed from the given large design space and finally focused on the location of the global optimal solution.

(3) a multi-level optimization method based on gradient descent algorithm is proposed. Several starting points in MA or IA subspaces are obtained by ESEA-OLHD experimental design method, and all local minimum points that may be global optimal solution are searched by ADAM gradient descent algorithm quickly, which not only facility the algorithm to get the global optimal solution, but also improve the accuracy of the surrogate model especially at the local extremum, and finally the accurate global optimal solution is obtained.

(4) The benchmark test problems in 12 different dimensions are used to verify the significance of the proposed method. When all the optimization results are counted, the average precision of the global optimal solution is improved by 21.3%, and the average number of expensive model calls is decreased by 31.5%. The results show that MFCPR-SGO method has a strong ability to deal with black box global optimization problem with expensive model.

By analyzing the results of numerical cases, this paper finds that the algorithm needs continuous improvement in the following three aspects: (1) more samples are needed when the shape of the global optimal solution is sharp or the overall mapping is multi-mode; (2) there is an optimization space for the selection of algorithm super parameters; (3) it is necessary to measure between the accuracy of the surrogate model and the regularization ability. In the next phase of the research, we will continue to improve the algorithm in terms of the above three aspects and apply it to the structural optimization and dynamic performance optimization of the vehicle for further testing and improvement.

CRediT authorship contribution statement

Kai Wu: Methodology, Validation, Formal analysis, Writing – original draft. **Faping Zhang:** Conceptualization, Writing – review & editing, Data curation, Project administration. **Yun He**

Zhang: Visualization, Investigation. **Yan Yan:** Visualization, review. **Shahid Ikramullah Butt:** Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the State Department project of China (Grant No. JCKY2017208A001, 2019-JCJQ-ZD-133-01, 51805502). The authors gratefully acknowledge the facilities provided by the Industrial and Intelligent System Engineering Laboratory (IISEL) at the Beijing Institute of Technology, China.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.17632/4nphrp8bxx.2.

Appendix B. Gaussian process variance

The variance of the gaussian process is determined by the distance between the sample points. It thinks that the farther the distance from the sample points is, the greater the variance is, and the higher the probability of error is. The variance of the gaussian process is completely determined by the distance between the coordinates of the sample points and has no relation with the observed response, so it is applicable to this problem. For an input set $X = x_i \in R^d | i = 1, \dots, n$ you can compute the set $y = y_i | i = 1, \dots, n$ of observations. The input set and the observation set constitute the training set D. Gaussian process regression can obtain the mapping relation $(f(\cdot)\colon R^d| \rightarrow R$ between input X and output y, not only that, but also the distribution parameters of each unknown point x^* on the mapping relation. Where the mean $\mu(x^*)$ represents the predicted value $f(x^*)$ of this point and the variance $\sigma(x^*)$ represents the error of this point. According to the gaussian process regression method, the joint prior distribution of observed value y and predicted value f_* is subject to:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \tag{B.1}$$

where, K, K_* and K_{**} are defined as follows:

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \cdots & \cdots & \cdots & \cdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix} \tag{B.2}$$

$$K_* = \begin{bmatrix} k(x_*, x_1) & k(x_*, x_2) & \cdots & k(x_*, x_n) \end{bmatrix} \tag{B.3}$$

$$K_{**} = k(x_*, x_*) \tag{B.4}$$

where, $k(x_1, x_1)$ is the covariance function. Generally speaking, the isotropic mean-square exponential kernel function is the form of the covariance function often used. Its function form is as follows:

$$k(x_i, x_j) = \exp(-\frac{\lambda \left\| x_i - x_j \right\|^2}{2}) \tag{B.5}$$

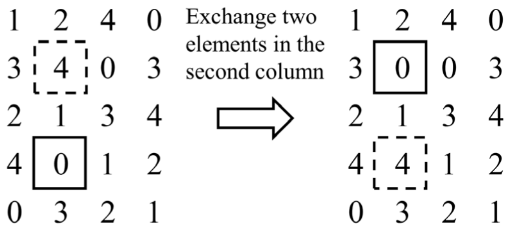


Fig. C.1. The new sample matrix is generated by element substitution.

Where, $\|\cdot\|^2$ represents the sum of the squares of the elements in the vector. Therefore, given a test set $X^* = x_i^* \in R^d | i = 1, \dots, n$, according to the Bayesian regression method, the posterior probability can be calculated as follows:

$$p(f_* | X^*, X, f) = N(u_*, \sigma_*) \tag{B.6}$$

Where, $u^* = K_*^T K^{-1} f$, $\sigma_* = K_{**} - K_*^T K^{-1} K_*$ represents the predicted value and variance of this point respectively.

Appendix C. Improved Latin cube test design method

In this paper, the Latin hypercube experiment design method optimized by enhanced random evolutionary algorithm (ESEA-OLHD) is used. Compared with traditional Latin hypercube design (LHD), ESEA-OLHD algorithm is more effective in calculating time, the number of exchanges needed to generate new design and the optimality criterion. More importantly, the algorithm can distribute the samples more evenly in the design space, preserving the structural characteristics as much as possible. The introduction of the algorithm is divided into three parts: (1) generate new sample matrix by element exchange method; (2) evaluate the advantages and disadvantages indicators of sample matrix; (3) optimize the sample matrix.

C.1. Generate new sample matrix by element exchange method

Firstly, the traditional LHS method is used to generate the initial sample matrix. After that, the existing design is updated to construct a new design. Finally, calculate the criterion value of the new design and decide whether to replace the existing design with the new design. Updating an existing design is done primarily through column operations because column operations tend to preserve structural properties of the design associated with the column, such as balance and orthogonality. In this method, we focus on a special type of column operation called element swapping, in which two different elements in a column are exchanged to ensure that the balance attribute is maintained. Take a LHD matrix (5 × 4) as an example:

Obviously, after the element exchange, the balance attribute in column 2 is retained, and the design is still LHD (see Fig. C.1).

C.2. Indicators to evaluate the advantages and disadvantages of sample matrix:

A design is called a maximum distance design if it maximizes the distance between adjacent points. Morris and Mitchell (1995) proposed an extension of the maximum distance design criterion. Firstly, the distance matrix $D = [d_{ij}]_{n \times n}$ calculated according to the Eq. (C.1) is the symmetric matrix, and its element is the spacing of sample points in the current design.

$$D = [d_{ij}]_{n \times n}$$
$$d(x_i, x_j) = d_{ij} = \left[\sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|^2 \right]^{1/2} \tag{C.1}$$

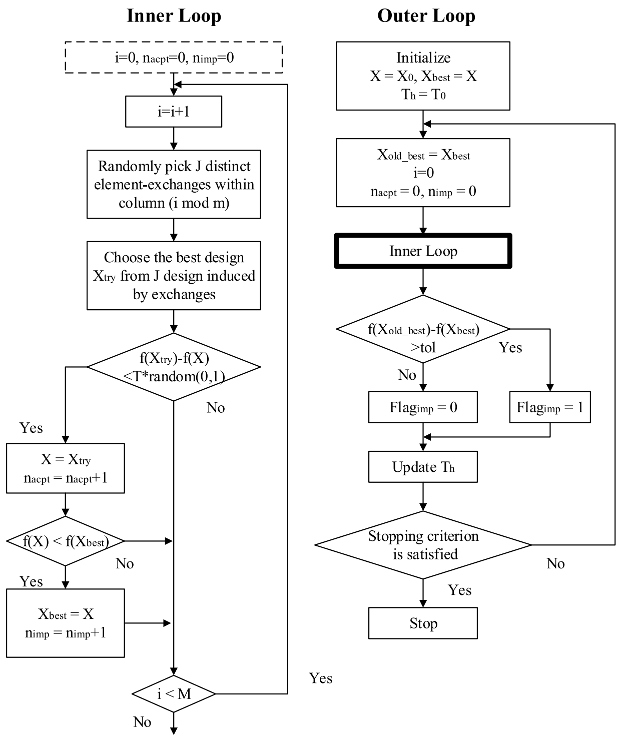


Fig. C.2. Algorithm process of ESEA-OLHS.

For a given design, minimize Eq. (C.2).

$$\phi_p = \left[\sum_{1 \le i < j \le n} (1/d_{ij})^p \right]^{1/p} = \left[\sum_{1 \le i < j \le n} d_{ij}^{-p} \right]^{1/p} \tag{C.2}$$

The above formula is in the form of p-norm, where the value of p is usually 2. Where, m is the dimension of sample points, and n is the number of sample points.

C.3. Sample matrix optimization process:

The enhanced stochastic evolutionary (ESE) algorithm is divided into two parts: (1) Inner loop; (2) Outer loop. The overall flow is shown in Fig. C.2.

(1) Inner loop:

The inner loop constructs new designs by exchanging column elements and decides whether to accept them based on an acceptance criterion. In the figure, each loop swaps elements in the mod(m, i) column of matrix X. Where, n is the number of samples, ne is the number of sample combinations. We set J to be $ne/5$ but no larger than 50, and M to be $2ne * m/J2$ but no larger than 100.

(2) Outer loop:

The outer loop controls the entire optimization process by adjusting the thresholds Th in the acceptance criteria. Set the initial value Th_0 to 0.005 times of the initial design evaluation value. If X is improved after an inner loop, the search process is converted to an improved process.

During the improvement process, adjust Th to quickly find a locally optimal design. If there is no improvement after a loop, then the search process will become an exploration process, and the adjustment of Th will help the algorithm get rid of the local optimal design. The adjustment strategy of Th during the improvement process and the exploration process is shown in Algorithm C.2.

Appendix D. ADAM gradient descent algorithm

The gradient is the direction in which a function rises most rapidly (it can be understood as the tangent direction of a two-dimensional space curve), and the negative gradient is the direction in which the multivariate function falls most rapidly. So we search in the negative gradient direction. The gradient $gradf = (gradf_1, gradf_2, \dots, gradf_D)$ of a multivariate function $p = f(x)$ is the vector of the partial derivatives of each of the dimensional variables. If the coordinate of the point is $x = (x_1, x_2, \dots, x_D)$, then the gradient formula can be used to obtain the gradient of the previous point $\nabla g = gradf(x) = (gradf_1(x_1), gradf_2(x_2), \dots, gradf_D(x_D))$. In the process of implementing the algorithm by programming, it is more convenient to use numerical gradient instead of analytic gradient when the problem dimension is not very high and the computational efficiency will not be significantly reduced. In advance, a small number $grad\_h = 1e{-}04$ is defined and the object function is known $f(x)$. Then the negative gradient of a value x_i of one dimension can be obtained by Eq. (D.1):

$$gradf(x_i) = -\frac{f(x_i + grad\_h) - f(-grad\_h)}{2h}, i = 1, 2, \dots, D \tag{D.1}$$

The conventional gradient descent method is updated by the following equation:

$$\begin{aligned} x^{t+1} &= x^t - \eta * \frac{\partial f}{\partial x}(x^t, y^t) \\ &= x^t - \eta * \nabla g \end{aligned} \tag{D.2}$$

where, η is the step size of the search forward. Considering the search efficiency and adaptability, the ADAM gradient descent method is used. In this method, the idea of momentum is introduced to exert influence on the current direction through the inertia given by the historical search direction, which can accelerate the convergence of the algorithm. On this basis, exponential decay sliding average is used to discard the history of the distant past and prevent the difficulty of training the algorithm near the local minimum. The update process of the method is as follows:

$$\begin{aligned} v_t &= \rho_1 * v_{t-1} + (1 - \rho_1) * \nabla g \\ \hat{v}_t &= \frac{v_t}{1 - \rho_1^t} \\ r_t &= \rho_2 * r_{t-1} + (1 - \rho_2) * \nabla g \odot \nabla g \\ \hat{r}_t &= \frac{r_t}{1 - \rho_2^t} \\ x_t &= x_{t-1} - \eta * \frac{\hat{v}_t}{\varepsilon + \sqrt{\hat{r}_t}} \end{aligned} \tag{D.3}$$

For most problems, the default parameter is: $\rho_1 = 0.9, \rho_2 = 0.99, \eta = 0.1, \varepsilon = 1\text{e} - 5$.

Appendix E. List of benchmark optimization

(1) Branin function (BR) with n = 2

$$f(x) = \left[x_2 - 5.1(\frac{x_1}{2\pi})^2 + (\frac{5}{\pi})x_1 - 6 \right]^2 \\ +10(1 - \frac{1}{8\pi}) \cos x_1 + 10$$

(2) Six-hump Camel-Back function (SC) with n = 2

$$\begin{aligned} f(x) =& \left[\ 1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 \right. \\ & \left. +6x_1x_2 + 3x_2^2)\ \right] \\ & \times \left[\ 30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 \right. \\ & \left. -36x_1x_2 + 27x_2^2)\ \right] \end{aligned}$$

$$-2 \le x_i \le 2, i = 1, 2$$

Algorithm C.1 Control strategy for parameter Th in the outer loop

```
    R_acp=n_acp/M;    R_imp=n_imp/M  ←  calculate the improvement coefficient and exploration coefficient
Improve process
    If    $R_{acp} > 0.1$
        If    $R_{imp} < R_{acp}$
            $T_h = 0.8 * T_h$;
        Else if $R_{imp} = R_{acp}$
            $T_h = T_h$;
        Else
            $T_h = T_h / 0.8$;
        End if
    Else
        $T_h = T_h / 0.8$;
    End if

Acception process
    If    $R_{acp} < 0.8$
        $T_h = T_h / 0.7$;
    Else
        $T_h = 0.9 * T_h$;
    End if
```

(3) Cross-IN-TRAY Function (CT) with n = 2

$$f(x) = -0.0001\left(\left|\sin(x_1)\sin(x_2)\exp\left(\left|100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right|\right)\right| + 1\right)^{0.1}$$

(4) Generalized polynomial function (GF) with n = 2

$$f(x) = (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2^2))^2$$
$$+(2.625 - x_1(1 - x_2^3))^2$$

(5) Himmelblau function (HM) with n = 2

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

(6) Leon function, n=2:

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$
$$-6 \le x_i \le 6, i = 1, 2$$

(7) Shekel function, n=4:

$$f(x) = -\sum_{i=1}^{5}(c_i + \sum_{j=1}^{4}(x_j - a_{ji})^2)^{-1}$$

$$a = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 \\ 4 & 1 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 3 \\ 4 & 1 & 8 & 6 & 7 \end{bmatrix}; c = \begin{bmatrix} 0.1 & 0.2 & 0.2 & 0.4 & 0.4 \end{bmatrix}$$

$$0 \le x_i \le 10, i = 1, 2, 3, 4$$

(8) Levy function, n=4:

$$f(x) = \sin 2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2(1 + 10\sin 2(\pi y_i + 1))$$
$$+(y_n - 1)^2(1 + 10\sin 2(2\pi y_n))$$
$$y_i = 1 + \frac{x_i - 1}{4}; -10 \le x_i \le 10; i = 1, 2, 3, 4;$$

(9) Trid function, n=6, 10:

$$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$$

(10) Sphere function, n=10:

$$f(x) = \sum_{i=1}^{10} x_i^2$$
$$-5.12 \le x_i \le 5.12, i = 1, \ldots, 10$$

(11) Sum squares function (Sums)

$$f(x) = \sum_{i=1}^{15} i x_i^2$$

(12) A function of 16 variables (F16) with n = 16

$$f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1), i, j = 1, 2, \ldots, 16$$

$$\alpha_{ij(row1-8)} = \begin{bmatrix} 1001001100000001 \\ 0110001001000000 \\ 0010001011000100 \\ 0001001000100010 \\ 0000110001010001 \\ 0000010100000010 \\ 0000001000101000 \\ 0000000101000010 \end{bmatrix};$$

$$\alpha_{ij(row9-16)} = \begin{bmatrix} 0000000010010001 \\ 0000000001000100 \\ 0000000000101000 \\ 0000000000010100 \\ 0000000000001100 \\ 0000000000000100 \\ 0000000000000010 \\ 0000000000000001 \end{bmatrix}$$

References

[1] T.W. Simpson, J.D. Peplinski, P.N. Koch, J.K. Allen, Metamodels for computer-based engineerin design: survey and recommendations, Eng. Comput. 17 (2001) 129–150, http://dx.doi.org/10.1007/pl00007198.

[2] R. Yondo, E. Andres, E. Valero, A review on design of experiments and surrogate models in aircraf real-time and many-query aerodynamic analyses, Prog. Aerosp. Sci. 96 (2018) 23–61, http://dx.doi.org/10.1016/j.paerosci.2017.11.003.

[3] B. Liu, Q. Zhang, G.G.E. Gielen, A Gaussian process surrogate model assisted evolutionar algorithm for medium scale expensive optimization problems, Ieee Trans. Evol. Comput. 18 (2014) 180–192, http://dx.doi.org/10.1109/tevc.2013.2248012.

[4] Y. Nie, Z. Tang, F. Liu, J. Chang, J. Zhang, A data-driven dynamics simulation framework fo railway vehicles, Veh. Syst. Dyn. 56 (2018) 406–427, http://dx.doi.org/10.1080/00423114.2017.1381981.

[5] M. Taheri, M. Ahmadian, Machine learning from computer simulations with applications in Rai vehicle dynamics, Veh. Syst. Dyn. 54 (2016) 653–666, http://dx.doi.org/10.1080/00423114.2016.1150497.

[6] S. Koziel, L. Leifsson, Surrogate-based aerodynamic shape optimization by variable-resolutio models, Aiaa J. 51 (2013) 94–106, http://dx.doi.org/10.2514/1.J051583.

[7] J. Park, K.-Y. Kim, Meta-modeling using generalized regression neural network and particle swarm optimization, Appl. Soft Comput. 51 (2017) 354–369, http://dx.doi.org/10.1016/j.asoc.2016.11.029.

[8] G.G. Wang, S. Shan, Review of metamodeling techniques in support of engineering desig optimization, J. Mech. Des. 129 (2007) 370–380, http://dx.doi.org/10.1115/1.2429697.

[9] Q. Zeng, H. Huang, A stable and optimized neural network model for crash injury severit prediction, Accid. Anal. Prev. 73 (2014) 351–358, http://dx.doi.org/10.1016/j.aap.2014.09.006.

[10] Z. Li, P. Liu, W. Wang, C. Xu, Using support vector machine models for crash injury severity analysis, Accid. Anal. Prev. 45 (2012) 478–486, http://dx.doi.org/10.1016/j.aap.2011.08.016.

[11] G.H. Teichert, K. Garikipati, Machine learning materials physics: Surrogate optimization and multi-fidelity algorithms predict precipitate morphology in an alternative to phase field dynamics, Comput. Methods Appl. Mech. Engrg. 344 (2019) 666–693, http://dx.doi.org/10.1016/j.cma.2018.10.025.

[12] A.J. Booker, J.E. Dennis, P.D. Frank, D.B. Serafini, V. Torczon, M.W. Trosset, A rigorous framework for optimization of expensive functions by surrogates, Struct. Optim. 17 (1999) 1–13, http://dx.doi.org/10.1007/bf01197708.

[13] A. Younis, Z. Dong, Trends, features, and tests of common and recently introduced global optimization methods, Eng. Optim. 42 (2010) 691–718, http://dx.doi.org/10.1080/03052150903386674.

[14] D.C. Montgomery, Design and Analysis of Experiment, Vol. xii, John Wiley, Chichester, 2001, http://dx.doi.org/10.2113/gseegeosci.xxviii.3.303.

[15] W. Shen, D.K.J. Lin, C.-J. Chang, Design and analysis of computer experiment via dimensional analysis, Qual. Eng. 30 (2018) 311–328, http://dx.doi.org/10.1080/08982112.2017.1320726.

[16] F.A.C. Viana, G. Venter, V. Balabanov, An algorithm for fast optimal latin hypercube design of experiments, Internat. J. Numer. Methods Engrg. 82 (2010) 135–156, http://dx.doi.org/10.1002/nme.2750.

[17] M.E. Johnson, L.M. Moore, D. Ylvisaker, Minimax and maximin distance designs, J. Statist. Plann. Inference 26 (1990) 131–148, http://dx.doi.org/10.1016/0378-3758(90)90122-b.

[18] X. Kang, X. Deng, Design and analysis of computer experiments with quantitative and qualitative inputs: A selective review, Wiley Interdiscip. Rev.-Data Min. Knowl. Discov. 10 (2020) http://dx.doi.org/10.1002/widm.1358.

[19] R.L. Hardy, Multiquadric equations of topography and other irregular surfaces, J. Geophys. Res. 76 (1971) 1905, http://dx.doi.org/10.1029/JB076i008p01905.

[20] W.A. Jensen, Response surface methodology: Process and product optimization using designed experiments 4th edition, J. Qual. Technol. 49 (2017) 186–187, http://dx.doi.org/10.1080/00224065.2017.11917988.

[21] P. Langley, H.A. Simon, Applications of machine learning and rule induction, Commun. Acm 38 (1995) 55–64, http://dx.doi.org/10.1145/219717.219768.

[22] A. Agarwal, L.T. Biegler, A trust-region framework for constrained optimization using reduced order modeling, Opt. Eng. 14 (2013) 3–35, http://dx.doi.org/10.1007/s11081-011-9164-0.

[23] M. Da Lio, D. Bortoluzzi, G.P.R. Papini, Modelling longitudinal vehicle dynamics with neural networks, Veh. Syst. Dyn. (2019) http://dx.doi.org/10.1080/00423114.2019.1638947.

[24] S. Bruni, J. Vinolas, M. Berg, O. Polach, S. Stichel, Modelling of suspension components in a rail vehicle dynamics context, Veh. Syst. Dyn. 49 (2011) 1021–1072, http://dx.doi.org/10.1080/00423114.2011.586430.

[25] T. Bartz-Beielstein, M. Zaefferer, Model-based methods for continuous and discrete global optimization, Appl. Soft Comput. 55 (2017) 154–167, http://dx.doi.org/10.1016/j.asoc.2017.01.039.

[26] E. Acar, Various approaches for constructing an ensemble of metamodels using local measures, Struct. Multidiscip. Optim. 42 (2010) 879–896, http://dx.doi.org/10.1007/s00158-010-0520-z.

[27] A. Chaudhuri, R.T. Haftka, Efficient global optimization with adaptive target setting, Aiaa J. 52 (2014) 1573–1577, http://dx.doi.org/10.2514/1.J052930.

[28] D.R. Jones, A taxonomy of global optimization methods based on response surfaces, J. Global Optim. 21 (2001) 345–383, http://dx.doi.org/10.1023/a:1012771025575.

[29] Y. Tang, J. Chen, J. Wei, A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions, Eng. Optim. 45 (2013) 557–576, http://dx.doi.org/10.1080/0305215x.2012.690759.

[30] A. Bhosekar, M. Ierapetritou, Advances in surrogate based modeling, feasibility analysis, and optimization: A review, Comput. Chem. Eng. 108 (2018) 250–267, http://dx.doi.org/10.1016/j.compchemeng.2017.09.017.

[31] V.M. Perez, J.E. Renaud, L.T. Watson, Adaptive experimental design for construction of response surface approximations, Aiaa J. 40 (2002) 2495–2503, http://dx.doi.org/10.2514/2.1593.

[32] S.E. Gano, J.E. Renaud, J.D. Martin, T.W. Simpson, Update strategies for kriging models used in variable fidelity optimization, Struct. Multidiscip. Optim. 32 (2006) 287–298, http://dx.doi.org/10.1007/s00158-006-0025-y.

[33] S. Wild, C. Shoemaker, Global convergence of radial basis function trust-region algorithms for derivative-free optimization, SIAM Rev. 55 (2013) http://dx.doi.org/10.1137/120902434.

[34] G.G. Wang, Z.M. Dong, P. Aitchison, Adaptive response surface method - A global optimization scheme for approximation-based design problems, Eng. Optim. 33 (2001) 707–733, http://dx.doi.org/10.1080/03052150108940940.

[35] H. Akbari, A. Kazerooni, KASRA: A Kriging-based Adaptive Space Reduction Algorithm for global optimization of computationally expensive black-box constrained problems, Appl. Soft Comput. 90 (2020) 22, http://dx.doi.org/10.1016/j.asoc.2020.106154.

[36] L. Shu, X. Meng, Q. Zhou, H. Jiexiang, J. Xu, A variable-fidelity modeling method based on self-organizing maps spatial reduction, 2016, http://dx.doi.org/10.1109/IEEM.2016.7798172.

[37] Y. Yang, X.D. Yao, J.G. Yang, Y.S. Zhang, F. Yuan, Thermo-drifting error modeling of spindle based on combination of principal component analysis and BP neural network, Shanghai Jiaotong Daxue Xuebao/J. Shanghai Jiaotong Univ. 47 (2013) 750-753+759.

[38] M. Buhmann, Radial basis functions: Theory and implementations, Radial Basis Funct. 12 (2003) http://dx.doi.org/10.1017/CBO9780511543241.

[39] V.M. Perez, J.E. Renaud, L.T. Watson, Reduced sampling for construction of quadratic response surface approximations using adaptive experimental design, Eng. Comput. 25 (2008) 764–782, http://dx.doi.org/10.1108/02644400810909607.

[40] K.-T. Fang, D.K.J. Lin, Ch, Uniform experimental designs and their applications in industry, in: Handbook of Statistics, Elsevier, 2003, pp. 131–170, http://dx.doi.org/10.1016/S0169-7161(03)22006-X (Ch. 4).

[41] R.C. Jin, W. Chen, A. Sudjianto, An efficient algorithm for constructing optimal design of computer experiments, J. Statist. Plann. Inference 134 (2005) 268–287, http://dx.doi.org/10.1016/j.jspi.2004.02.014.

[42] A. Younis, Z. Dong, Metamodelling and search using space exploration and unimodal region elimination for design optimization, Eng. Optim. 42 (2010) 517–533, http://dx.doi.org/10.1080/03052150903325540.

[43] P. Ye, G. Pan, Z. Dong, Ensemble of surrogate based global optimization methods using hierarchical design space reduction, Struct. Multidiscip. Optim. 58 (2018) 537–554, http://dx.doi.org/10.1007/s00158-018-1906-6.

[44] L.Q. Wang, S.Q. Shan, G.G. Wang, Mode-pursuing sampling method for global optimization on expensive black-box functions, Eng. Optim. 36 (2004) 419–438, http://dx.doi.org/10.1080/03052150410001686486.

[45] H. Dong, B. Song, Z. Dong, P. Wang, SCGOSR: Surrogate-based constrained global optimization using space reduction, Appl. Soft Comput. 65 (2018) 462–477, http://dx.doi.org/10.1016/j.asoc.2018.01.041.

[46] X.-S. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications, 2010, http://dx.doi.org/10.1002/9780470640425.

[47] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359, http://dx.doi.org/10.1023/a:1008202821328.

[48] M. Björkman, K. Holmström, Global optimization using the DIRECT algorithm in Matlab, 1999.

[49] J. Gu, G.Y. Li, Z. Dong, Hybrid and adaptive meta-model-based global optimization, Eng. Optim. 44 (2012) 87–104, http://dx.doi.org/10.1080/0305215x.2011.564768.