ICAS 2021 SHANG HAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021 Pudong Shangri-La, Shanghai, China

Efficient adaptive surrogate-based optimization using fuzzy clustering for complex system optimizations

Chunna Li[1], Wen Ji[1], Chunlin Gong[1], Hai Fang[2]

[1]Shaanxi Aerospace Flight Vehicle Design Key Laboratory, School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China
[2]Shanghai Electro-Mechanical Engineering Institute, Shanghai 201109, China

Abstract

Adaptive infilling is a key element for Surrogate-based optimization (SBO) in its efficiency and convergence. For complex system optimization problems with expensive black-box functions, the SBO with common adaptive infilling shows imperfection in local exploitation, especially for problems with large design space and many design variables. An efficient adaptive SBO using fuzzy clustering algorithm for the infilling procedure is proposed in the paper. In each refinement cycle during SBO process, a Kriging model is constructed using the high-fidelity samples in current design space; then the current design space is divided into several subspaces using the fuzzy clustering algorithm with respect to the features of the objective function; hence new infilling samples are selected within each subspace and parallelly solved; thereafter, the current design space is updated by merging the subspaces. The proposed method is validated and assessed by several benchmark tests with bound constraints, further it is used for solving a beam optimization problem and an interstage-section optimization of a rocket. The results indicate the proposed method performs well both in local exploitation and global exploration, and is efficient in solving complex system optimizations.

Keywords: surrogate-based optimization, complex system, fuzzy clustering, adaptive infilling

1. Introduction

In modern complex system optimizations, especially in aircraft design, the problems always have large design space and show strong nonlinearity due to complex couplings. Thus, it provides a prohibitive difficulty in employing complicated and time-consuming high-fidelity analysis techniques during optimization [1]. Surrogate model has been popularly used in lieu of expensive high-fidelity analyses in optimization after the fruitful paper [2], to improve optimization efficiency. However, the adaptability of surrogate-based optimization (SBO) in handling complex system optimizations are greatly influenced by the adaptive infilling [3].

Common adaptive infilling strategy includes maximizing surrogate error (MSE), minimizing surrogate prediction (MSP), the infilling based on the expected improvement (EI), the probability of improvement (PI), and the lower confidence bound (LCB) [4]. The EI-based infilling is widely used in SBO due to its balance in local exploitation and global exploration. However, abovementioned infilling strategies choose only one infill sample in each refinement cycle to update the surrogate model during SBO process.

Currently more efficient parallel adaptive infilling strategies are of our concern [5], which shows good balance in local exploitation and global exploration. Ginsbourger selects q infill samples simultaneously using the EI-based infilling, of which the Kriging Believer and Constant Liar are proposed to update the surrogate q times in one refinement cycle [6]. It is suitable for general nonlinear problems. The clustered Multiple Generalized Expected Improvement is proposed to adaptively balance the exploitation and exploration of the EI-based infilling, and choose multiple local extrema of the generalized EI function in parallel by the NM-simplex [7]. The method needs more local optimizations based on surrogate model. Li combines different common adaptive infilling strategies to generate infill samples to balance local exploitation and global exploration [8]. However, it is greatly influenced by the initial set of samples, especially in case that the initial samples lack too much information of the design space. Those strategies focus on adding infill samples balancing

local exploitation and global exploration.

A successive response surface methodology is proposed to reduce design space around an optimum during SBO process [9]. But it will find local optimum when initial samples loose design-space information. An intuitive methodology is proposed by Wang and Simpson to systematically reduce the design space during refinement [10], which depends more on the initial sampling, and might miss global optimum. Guo et al first generate a huge number of samples through the surrogate, then use the fuzzy c-means clustering to cluster those samples in order to identify interesting spaces for selecting infills [11]. Three metamodels are combined with the optimized weight factors, then the optimum and the promising samples in the sparse region are chosen as new infills within the reduced space [12]. The above methods also show deficiency in global exploration for problems with strong nonlinearity.

This paper proposes an efficient adaptive SBO based on Kriging surrogate model, by using the fuzzy clustering algorithm to identify several subspaces for select new infill samples, after which the design space is reformed. Hence, during the sequential infilling procedure, the design space is adaptively modified which can improve the adaptability and effectiveness of SBO for solving complex system optimization problems. Finally, the proposed method is validated, and applied to solve the beam and the rocket interstage-section optimization problems.

2. Methods

2.1 Procedure of the proposed surrogate-based optimization with fuzzy clustering

A general nonlinear optimization problem under constraints can be expressed as,

$$\begin{aligned} &\min\ f(\boldsymbol{x}) \\ &\text{s. t. } g_j(\boldsymbol{x}) \le 0 \quad j = 1,2,...,J \\ &\quad x_{\mathrm{L},i} \le x_i \le x_{\mathrm{U},i} \quad i = 1,2,...,m \end{aligned} \tag{1}$$

where, $g_j(\boldsymbol{x})$ is the jth constraint; $x_{\mathrm{L},i}$ and $x_{\mathrm{U},i}$ are respectively the lower and upper bound of the ith design variable. It is assumed that the objective function or the constraint functions are expensive to evaluate. Thus, SBO is taken to improve optimization efficiency. The procedure of the proposed SBO method is shown in Figure 1. An initial set of samples are first generated by a certain design of experiment (DoE) method, then analyses are performed to obtain high-fidelity data, based on which the Kriging model of the objective is built. Thereafter, the objective function is optimized based on the Kriging model, and high-fidelity analysis is performed of the optimum. If the surrogate model is accurate, the optimum could be the final solution; or else, the adaptive infilling is executed.

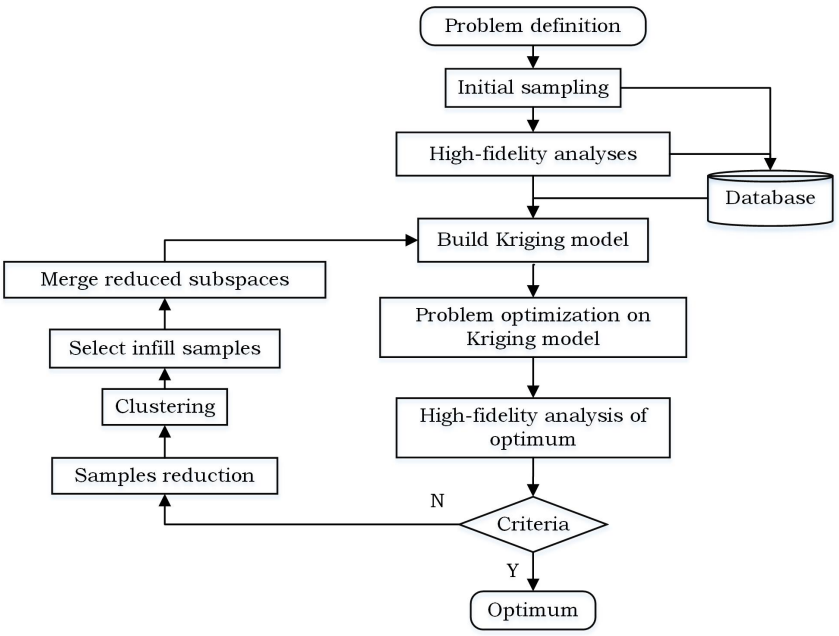


Figure 1 – Procedure of the proposed SBO combined with fuzzy clustering algorithm.

In comparison with the basic SBO process, the adaptive infilling procedure consists of the following four steps: 1) Reduction of the pseudo samples generated in current design space; 2) clustering of the trimmed pseudo samples; 3) Select of infill samples in subspaces; 4) merging of the reduced subspaces. The Kriging modeling, the adaptive infilling procedure, and the constraints handling are detained in the following subsections.

2.2 Kriging model

The South Africa geologist Daniel G. Krige first proposed the Kriging model in 1951 [13], to predict the unobserved value in a random field using the observations nearby. The French mathematician Georges Matheron developed the regional-variable theory and coined it Kriging [14].

Suppose an objective or a constraint function with m design variables, which is expensive to obtain, then a Kriging model needs to be built for the unknown function y: $\Re^m \rightarrow \Re$ with respect to $\boldsymbol{x}$. First, a certain DoE method, e.g. Latin hypercube sampling (LHS), is used to generate an initial sampling plan $\boldsymbol{S}=[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\ldots, \boldsymbol{x}^{(n)}]^T$. Then high-fidelity analyses, such as CFD, are performed of the n samples to obtain the responses $\boldsymbol{y}=[y^{(1)}, y^{(2)}\ldots, y^{(n)}]^T$. Hence, the sampled data set as well as the corresponding responses could be used to build the Kriging model. A common Kriging model is an interpolation model based on statistics. It is expressed as,

$$\hat{y}(\boldsymbol{x}) = \mu + Z(\boldsymbol{x}) \tag{2}$$

where $\boldsymbol{x}$ is the design vector containing m design variables, and $\hat{y}(\boldsymbol{x})$ is the predicted function value at a certain $\boldsymbol{x}$; μ is the regression parameter, which is the mean of the observations; $Z(\boldsymbol{x})$ is a stochastic process with zero mean. $Z(\boldsymbol{x})$ can be calculated through the correlation between the local position and its nearby observations, which can be formed by the correlation matrix

$$\mathrm{Cov}\left[Z(\boldsymbol{x}^i), Z(\boldsymbol{x}^j)\right] = \sigma_{\mathrm{C}}^2 \boldsymbol{R}\left[R(\boldsymbol{x}^i, \boldsymbol{x}^j)\right] \tag{3}$$

in which $\boldsymbol{R}$ is the correlation matrix, and σ_{C} is its variance. R is the Gaussian correlation function which can be computed by,

$$R(\boldsymbol{x}^i, \boldsymbol{x}^j) = \exp\left[-\sum_{l=1}^{m} \theta_l \left| x_l^i - x_l^j \right|^2\right] \quad i, j = 1, 2, \ldots, n \tag{4}$$

where, n is the number of the samples; θ_l is the lth unknown correlation parameter for tuning Kriging model. The correlation function represents the influence of the neighbor samples on the local model accuracy and the smoothness of the model. Thus, the Kriging model can be expressed as,

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}) \tag{5}$$

where, $\boldsymbol{r}$ represents the correlation between the unobserved sample $\boldsymbol{x}$ and all the observed samples; and $\boldsymbol{y}$ denotes the vector containing the function values of all the samples. Hence, with a specified $\boldsymbol{\theta}$, the estimated regression parameter $\hat{\mu}$ and the estimated variance $\hat{\sigma}_{\mathrm{C}}$ can be computed by

$$\hat{\mu} = \frac{\mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{y}}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}} \quad , \quad \hat{\sigma}_{\mathrm{C}}{}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\hat{\mu})^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \mathbf{1}\hat{\mu})}{n} \tag{6}$$

In Eq. (6), $\boldsymbol{\theta}$ can be obtained by maximizing the concentrated logarithm likelihood function, as follows

$$Ln(\boldsymbol{\theta}) = -\left[n \ln\left(\hat{\sigma}_{\mathrm{C}}{}^2\right) + \ln|\boldsymbol{R}|\right] / 2 \tag{7}$$

Simultaneously, the mean square error of the Kriging model can also be computed by,

$$s^2(\boldsymbol{x}) = \hat{\sigma}_{\mathrm{C}}{}^2 \left[1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r} + \frac{\left(1 - \mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{r}\right)^2}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}}\right] \tag{8}$$

2.3 Adaptive infilling

To explain the adaptive infilling procedure, following expressions are first introduced. $\boldsymbol{D}_{\mathrm{init}}$ is the initial design space, where $\boldsymbol{x}_{\mathrm{L,init}}$ and $\boldsymbol{x}_{\mathrm{U,init}}$ are the lower and upper bound of the initial design space,

respectively. Then, c_1 is specified to restrict the minimum design space to $\boldsymbol{\Delta}_{\text{limit}}= c_1(\boldsymbol{x}_{\text{U,init}} - \boldsymbol{x}_{\text{L,init}})$ during refinement, in which c_1 is recommended as 0.1. Thus, in the first refinement cycle, the current design space is set to $\boldsymbol{D}_{\text{init}}$.

2.3.1 Pseudo samples reduction

Before clustering, a set of samples needs to be generated using surrogate model, named pseudo samples, because the real samples through high fidelity analyses are too few to be clustered. In the paper, the random LHS (RLHS) is used to generate N_{p} pseudo samples, as N_{p} is usually large than 100. Thus, the DoE method has small influence on the uniformity of the sampling plan.

In the jth refinement cycle, $\boldsymbol{x}_k$ (k=1, 2, …, N_{p}) pseudo samples are generated in the current design space $\boldsymbol{D}_j$=[$\boldsymbol{x}_{\text{L,c}}$, $\boldsymbol{x}_{\text{U,c}}$]. The objective are accordingly f_k (k=1, 2, …, N_{p}). Thus, the reduction of the pseudo samples is carried out by the following steps. a) Compute mean objective f_{m} of the pseudo samples; b) Calculate objective threshold by f_{t}=t_{r}($f_{\max}$-$f_{\min}$), in which t_{r} (0<t_{r}≤1) restricts the objective threshold, fmax and $f_{\min}$ are the maximum and minimum predicted objective respectively; c) Identify the value for deleting pseudo samples by comparing the mean and threshold objective f_{d}=max(f_{m}, f_{t}); d) Delete the pseudo sample, when f_{k}>f_{d} (k=1, 2, …, N_{p}), then N_{r} pseudo samples are remained for clustering. Thus, it makes a balance between the mean and threshold objective, which can keep enough samples for clustering, as well as to prevent fast reduction of the design space.

On the one hand, the reduction of the design space can amplify the local features of the design space, thus to improve optimization quality; on the other hand, it can reduce the samples for building the surrogate, accelerating modeling procedure.

2.3.2 Clustering of the trimmed pseudo samples

Clustering algorithm is a statistical analysis technique that can classify samples or indexes. Since a clustering algorithm can capture the features of each cluster, it is used to generate subspaces for selecting infill samples. In this step, the remained pseudo samples are clustered by the fuzzy cluster means (FCM) algorithm after pseudo samples reduction. The FCM algorithm is a kind of soft clustering algorithm first proposed by Bezdek [15], which determines the membership of the samples. If the number of the clusters N_{c} is given, the samples can be automatically partitioned into N_{c} clusteres by the FCM. Suppose there are N_{r} pseudo samples $\boldsymbol{x}_k$ (k=1, 2, …, N_{r}) after reduction, the discrepancy between the cluster centers and other samples is

$$J(\boldsymbol{U},\boldsymbol{V}) = \sum_{k=1}^{N_{\text{r}}}\sum_{i=1}^{N_{\text{c}}}(u_{ik})^d \left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|^2 \tag{9}$$

in which $\boldsymbol{U}$ is the fuzzy partition of the pseudo samples; u_{ik} is the weight of the kth pseudo sample in the ith cluster; $\boldsymbol{v}_i$ is the ith cluster center; d is a constant larger than 1, and is set to 2 here. $\boldsymbol{U}$ and $\boldsymbol{V}$ can be obtained by minimizing the discrepancy, and the weight u_{ik} of each sample to every cluster center could also be obtained, which satisfies

$$\sum_{i=1}^{N_{\text{c}}} u_{ik} = 1 \tag{10}$$

The abovementioned optimization problem can be solved by the Lagrange multiplier method, subject to the equality constraint in Eq. (10). Hence the cluster centers are obtained by employing the necessary condition of the Lagrange function, as is shown in Eq. (11), and the weights are solved by evaluating the membership of each sample shown in Eq. (12).

$$\boldsymbol{v}_i = \sum_{k=1}^{N_{\text{r}}} u_{ik}^d \boldsymbol{x}_k \Big/ \sum_{k=1}^{N_{\text{r}}} u_{ik}^d \quad , \quad i = 1,2,\ldots,N_{\text{c}} \tag{11}$$

$$u_{ik} = 1 \Big/ \sum_{j=1}^{N_{\text{c}}} \left(\frac{\left\|\boldsymbol{x}_k - \boldsymbol{v}_i\right\|}{\left\|\boldsymbol{x}_k - \boldsymbol{v}_j\right\|} \right)^{\frac{2}{d-1}} \quad , \quad 0 \le u_{ik} \le 1 \tag{12}$$

Given a set of initial cluster centers (iteration step k=1), the membership matrix $\boldsymbol{U}_k$ needs to be first computed, then the cluster center matrix $\boldsymbol{V}_k$ could be updated. If the cluster centers are stable, final

cluster center matrix is obtained; if not, update the iteration step by $k = k + 1$.

2.3.3 Infill samples selection

By means of clustering, new samples are generated within each partitioned subspace to enhance both global exploitation and local exploration. The hybrid refinement strategy combining the EI-based infilling and the MSP is taken to select new samples. Suppose in the ith subspace $[\boldsymbol{x}_{\mathrm{L},i}, \boldsymbol{x}_{\mathrm{U},i}]$, N_c samples $\boldsymbol{S}_1=\{\boldsymbol{x}_j | j=1, 2, \ldots, N_\mathrm{c}\}$ are generated by maximizing EI function, and N_c samples $\boldsymbol{S}_2=\{\boldsymbol{x}_j | j=1, 2, \ldots, N_\mathrm{c}\}$ are from minimizing the surrogate prediction. Hence the new samples selected by the infilling criterion are expressed as $\boldsymbol{S}_\mathrm{new}=\{\min\{\boldsymbol{S}_2\}, \boldsymbol{S}_1\}$. As the subspaces might overlap, new samples selected in each subspaces might quite similar, thus only one of those new samples is kept, and other similar samples need to be removed from the sample set $\boldsymbol{S}_\mathrm{new}$. The new samples finally selected are expressed as $\{\boldsymbol{x}_{\mathrm{new},i}\} \subset \boldsymbol{S}_\mathrm{new}$ ($i=1, 2, \ldots, p_j$). Infill samples selection simultaneously considers the global and local features of the objective function, which is more adaptive.

2.3.4 Subspaces merging

After selecting infill samples, the subspaces need to be merged into a new design space, thus to reduce the design space during optimization. The merged design space $\boldsymbol{D}_m = \cup [\boldsymbol{x}_{\mathrm{L},i}, \boldsymbol{x}_{\mathrm{U},i}]$($i=1, 2, \ldots, N_\mathrm{c}$), is the union of N_c subspaces, and the design space is updated by

$$\boldsymbol{D}_\mathrm{u} = \max\left(\min\left(\boldsymbol{D}_\mathrm{m}, \boldsymbol{D}_\mathrm{init}\right), \boldsymbol{\Delta}_\mathrm{limit}\right) \tag{13}$$

where, $\boldsymbol{\Delta}_\mathrm{limit}$ is the restricted minimum design space, and $\min(\boldsymbol{D}_m, \boldsymbol{D}_\mathrm{init})$ restricts the updated design space within the initial design space. Then the design space can be updated by $\boldsymbol{D}_\mathrm{u}$. After updating the design space, high-fidelity analyses are performed for those new infill samples in parallel to improve optimization efficiency.

2.4 Constraints handling

Different from penalizing the objective function, the constraints are handled here during refinement, by constraining the EI function and penalizing the surrogate prediction as well. During SBO, surrogate models are needed to be built for constraints when they are expensive to evaluate. To reduce computational cost, only one surrogate model is fitted by using the maximum value of all constraint functions at each sample. Hence, the constraint function might be discontinuous. If $g(\boldsymbol{x})<0$ is defined to satisfy the constraint, the probability of satisfying all the constraints of the EI function is

$$P_\mathrm{EI}\left(g_\mathrm{max}\left(\boldsymbol{x}\right) < 0\right) = 1 - \Phi\left(\hat{g}_\mathrm{max}\left(\boldsymbol{x}\right) \big/ s_\mathrm{c}\left(\boldsymbol{x}\right)\right) \tag{14}$$

in which, $g_\mathrm{max}(\boldsymbol{x})$ is the maximum value of all constraint functions at $\boldsymbol{x}$, computed by $g_\mathrm{max}(\boldsymbol{x})=\max g_i(\boldsymbol{x})$; $\hat{g}_\mathrm{max}(\boldsymbol{x})$ is the built surrogate model for the constraints; $s_\mathrm{c}(\boldsymbol{x})$ is the error of the constraint model. Thus, the constrained EI function is expressed in Eq. (15), and it is used to replace the EI function during refinement procedure to select infill samples.

$$E'[I(\boldsymbol{x})] = E[I(\boldsymbol{x})] \cdot P_\mathrm{EI}\left(g_\mathrm{max}\left(\boldsymbol{x}\right) < 0\right) \tag{15}$$

For the MSP infilling criterion, the predicted objectives need to be penalized during refinement to prevent infills against constraints being added into the design space. The penalty function for the predicted objective function is $P_\mathrm{MSP}(\boldsymbol{x})=a \cdot \max(0, g_\mathrm{max}(\boldsymbol{x}))$, where a is the factor to tune the penalty. The value of a can be specified as the inverse of the optimum in current iteration, it should be restrained to a small positive value to prevent the penalty becoming infinity. Hence, during refinement the surrogate of the objective function is replaced by the penalized surrogate for the objective, expressed as

$$\hat{y}'\left(\boldsymbol{x}\right) = \hat{y}\left(\boldsymbol{x}\right) + P_\mathrm{MSP}\left(\boldsymbol{x}\right) \tag{16}$$

3. Test cases and applications

3.1 Analytical tests

The proposed method is validated by several analytical benchmark test cases. The proposed method is named SBO-FCM for short. Details of the benchmark test cases are shown in Table 1.

Table 1 – Illustration of the benchmark test cases.

Tests	N_m	N_f	Formular	$\boldsymbol{x}^*$	f^*	$[\boldsymbol{x}_L, \boldsymbol{x}_U]$
Branin(BR)	2	3	$f(\boldsymbol{x})=[x_2-5.1(0.5x_1/\pi)^2+(5/\pi)x_1-6]^2$ $+10[1-0.125(1/\pi)]\cos x_2+10$	(-π,12.275) (π, 2.275) (9.4248,2.475)		$x_1\in[-5,10]$ $x_2\in[0,15]$
Rosenbrock(RB)	2	1	$f(\boldsymbol{x})=100(x_2-x_1^2)^2+(x_1-1)^2$	(1,1)	0	$x_1,x_2\in[-5,10]$
Six-hump camel back(SC)	2	2	$f(\boldsymbol{x})=4x_1^2-2.1x_1^4+x_1^6/3+x_1x_2-4x_2^2+4x_1^4$	(0.0898, -0.7126) (-0.0898, 0.7126)	-1.0316	$x_1,x_2\in[-2,2]$
Rastrigin(RS)	2	1	$f(\boldsymbol{x})=18+\sum_{i=1}^{2}\left[x_i^2-10\cos(2\pi x_i)\right]$	(0, 0)	0	$x_1,x_2\in[-1,1]$
Griewank(GN)	2	1	$f(\boldsymbol{x})=\sum_{i=1}^{2}x_i^2/200-\prod_{i=1}^{2}\cos(x_i/\sqrt{i})+1$	(0, 0)	0	$x_1,x_2\in[-100,100]$
Generalized polynomial Function(GF)	2	1	$f=t_1^2+t_2^2+t_3^2$ where $t_j=a_j-x_1\left(1-x_2^j\right)\quad j=1,2,3$ and $a_1=1.5, a_2=2.25, a_3=2.625$	(0, 0)	0	$x_1,x_2\in[-5,5]$
Hartmann(HN6)	6	1	$f(\boldsymbol{x})=-\sum_{k=1}^{4}b_k\exp\left[-\sum_{i=1}^{6}a_{ki}\left(x_i-p_{ki}\right)^2\right]$	(0.2017, 0.1500, 0.4769, 0.2753, 0.3117, 0.6573)	-3.322	$x_i\in[0,1]$ $i=1,2,\cdots,6$
F16 function(F16)	16	1	$f=\sum_{i=1}^{16}\sum_{j=1}^{16}a_{ij}\left(x_i^2+x_i+1\right)\left(x_j^2+x_j+1\right)$		25.878	$x_i\in[-1,0]$ $i=1,2,\cdots,16$

Parameters in test case HN6 and F16 can be related in reference [8].

The SBOs with EI-based infilling (SBO-EI) and with MSP (SBO-MSP) are taken to compare with the proposed method to evaluate the performance of the adaptive infilling. Setting of the parameters N_c, N_p and t_r for adaptive infilling can be found in Table 2. Initial samples are determined following the policy: $(m+1)(m+2)/2$ samples for problems with $m<6$; $2m$ samples for problems with $m\geq 6$. For test cases in two dimension, 6 initial samples are generated to avoid missing promising regions due to too few samples.

Results of 30 repeated runs by each method are also compared in Table 2. It shows the SBO-FCM has better performance than the SBO-EI and SBO-MSP in optimization quality and robustness for test cases of SC, BR, RB, GN, GF, HN6 and F16. Since the RS function has many local optima, the SBO-EI is inefficient in exploring more local optima. In comparison, the SBO-FCM finds more local optima due to adaptive infilling. Besides, in view of the variances, the SBO-FCM has smallest value in comparison with the SBO-EI and SBO-MSP, which indicates that the SBO-FCM has best robustness.

Table 2 – Results comparison by different SBOs.

Test cases	DoE	SBO-FCM					SBO-EI		SBO-MSP	
		N_c	N_p	t_r	mean opt.	variance	mean opt.	variance	mean opt.	variance
SC	6	2	100	0.75	-1.0312	3.5125e-6	-1.0300	5.5469e-5	-1.0261	2.6159e-4
BR	6	2	100	0.7	0.3978	2.8206e-9	0.3984	1.4624e-6	0.6622	3.9860e-1
RB	6	2	100	0.75	0.0036	1.9748e-4	0.0217	9.0286e-3	0.7028	2.4285
GN	6	4	100	0.1	0.0439	6.4191e-3	0.0952	1.7599e-2	0.3804	3.1022e-1
RS	6	8	200	0.9	-1.9960	4.8886e-4	-1.5616	4.7992e-1	-1.4600	5.8407e-1
GF	6	2	200	0.8	0.0136	8.0778e-4	0.0128	7.7404e-4	0.4924	4.7491e-1
HN6	28	2	100	0.7	-3.2690	6.5762e-3	-3.1687	5.3283e-2	-3.1656	6.3367e-2
F16	32	2	100	0.7	25.8765	5.8149e-7	25.8765	4.1221e-6	25.8788	2.4042e-5

The number of the total samples, the iteration cycles, and the optimum range of the 30 repeated

runs are compared in Table 3. The optimum range by the SBO-FCM is the narrowest for each test case. It indicates that the SBO-FCM has best robustness. In comparison, the SBO-FCM performs fewest iteration cycles, although it always has most samples evaluated.

The box plots in Figure 2 indicates that the SBO-FCM has best robustness in comparison with SBO-EI and SBO-MSP. The median optimum by the SBO-FCM is close to the lower bound of the optimum range for all the test cases, which indicates it has good robustness. The SBO-EI also shows good robustness for simple problem; however, its robustness decreases for complex problems. In comparison, the SBO-MSP shows worst robustness.

Table 3 – Comparison of total samples, iteration cycles and optimum range by different SBOs.

Test cases	SBO-FCM			SBO-EI			SBO-MSP		
	Opt. range	samples	cycles	Opt. range	samples	cycles	Opt. range	samples	cycles
SC	[-1.032, -1.021]	88.43	25.80	[-1.032, -0.991]	46.50	37.17	[-1.032, -0.953]	53.00	39.60
BR	[0.398, 0.398]	87.77	25.77	[0.398, 0.402]	48.00	36.70	[0.398, 1.943]	83.10	59.03
RB	[1.919e-6, 0.076]	247.60	73.00	[2.436e-4, 0.519]	125.60	75.37	[1.076e-6, 7.591]	351.17	226.17
GN	[2.092e-9, 0.150]	215.63	40.00	[1.085e-4, 0.397]	596.27	309.23	[1.010e-3, 1.768]	376.03	244.70
RS	[-2.000, -1.879]	168.60	17.97	[-2.000, -0.539]	47.80	30.00	[-2.000, -0.539]	78.40	50.60
GF	[2.736e-5, 0.113]	672.72	136.08	[4.162e-6, 0.124]	551.83	286.63	[4.550e-4, 1.619]	617.50	364.97
HN6	[-3.322, -3.188]	112.73	26.27	[-3.322, -2.629]	71.73	34.37	[-3.322, -2.629]	68.40	148.70
F16	[25.875, 25.877]	106.23	22.57	[25.875, 25.882]	84.50	47.13	[25.876, 25.887]	96.23	26.88

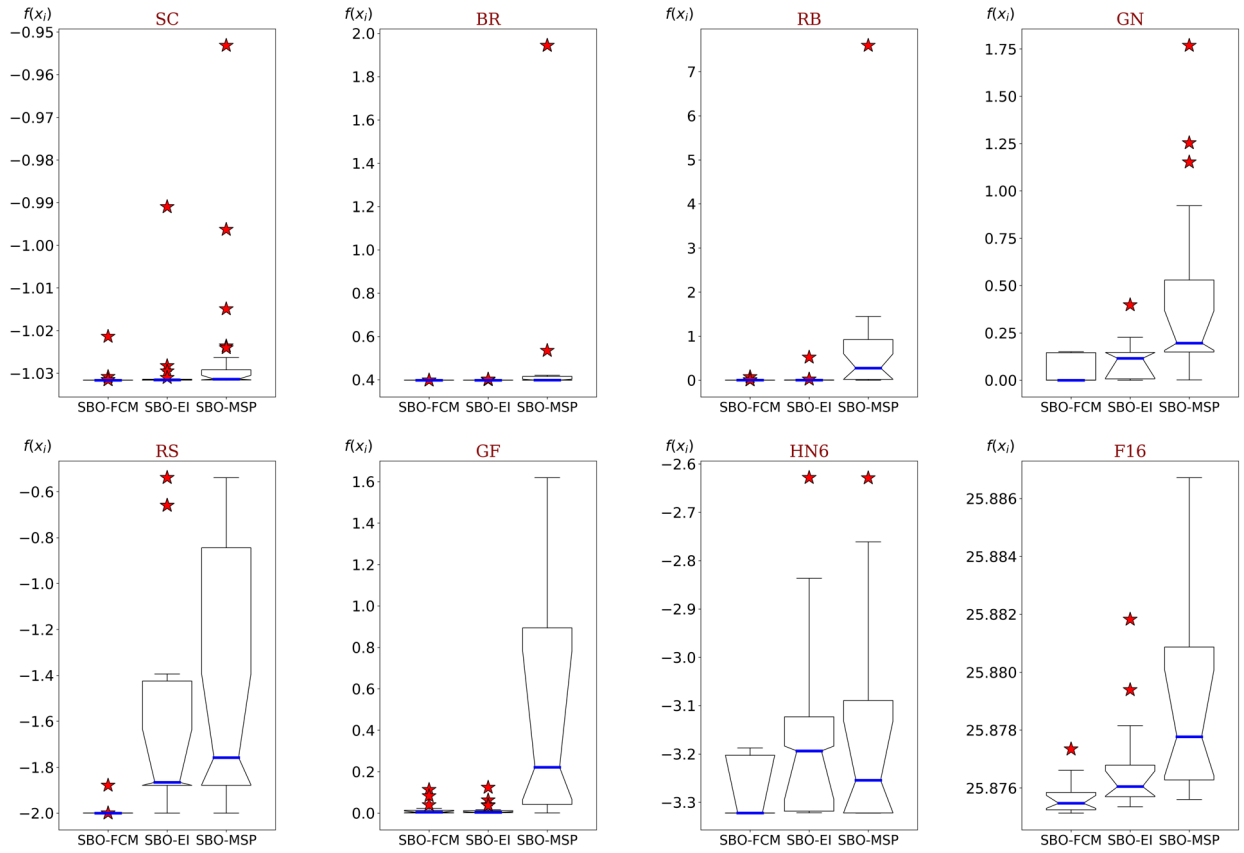
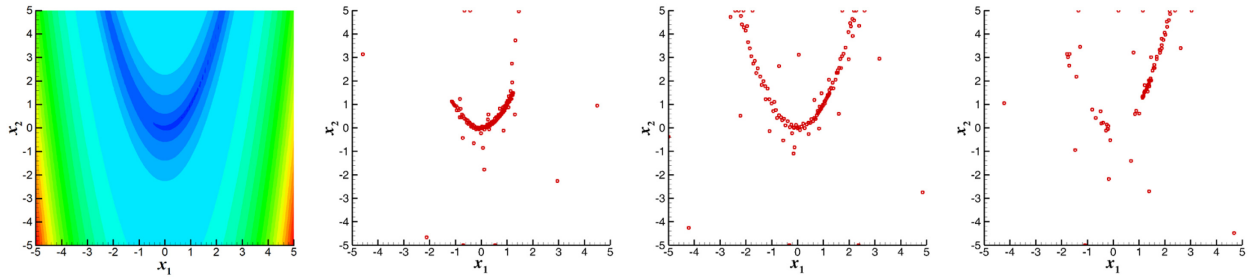


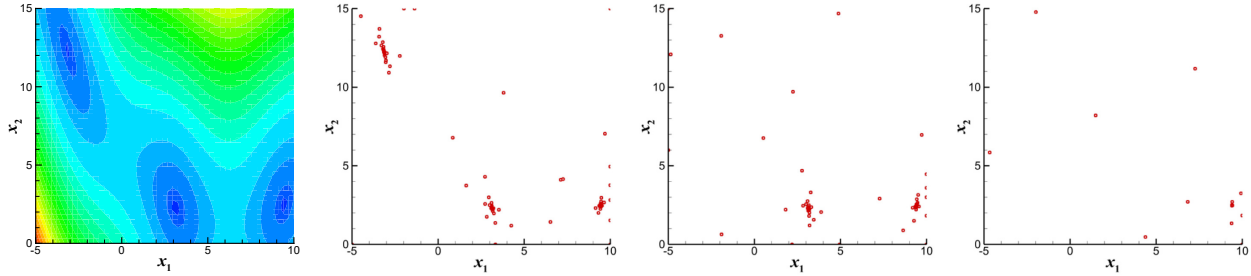
Figure 2 – Box plots of 30 optima by different SBOs.

The adaptivity of the infilling is compared in Figure 3. The samples for the best optimization of the 30 runs by different SBOs are compared. It indicates that the SBO-MSP just focuses on local exploitation, resulting fast direct convergence to a local optimum. The SBO-EI behaves much better, because the EI function can balance local exploitation and global exploration as well. However, for highly nonlinear problems, the SBO-EI also shows insufficiency in global exploration, proven by the

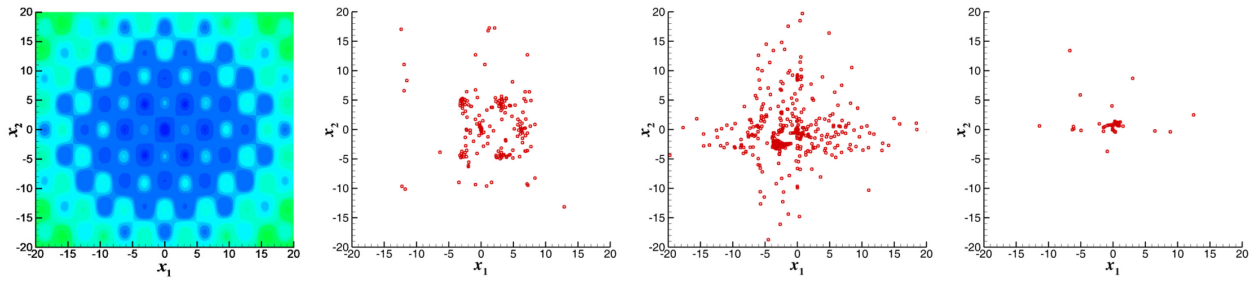
samples in the plots of the figure. The samples by the SBO-FCM are concentrated in regions with optima or ridges, which indicates the SBO-FCM has good adaptivity considering exploitation and exploration. Especially for optimizing the GN and GF functions, the SBO-FCM can find many valleys and the two long ridges of the objective function.



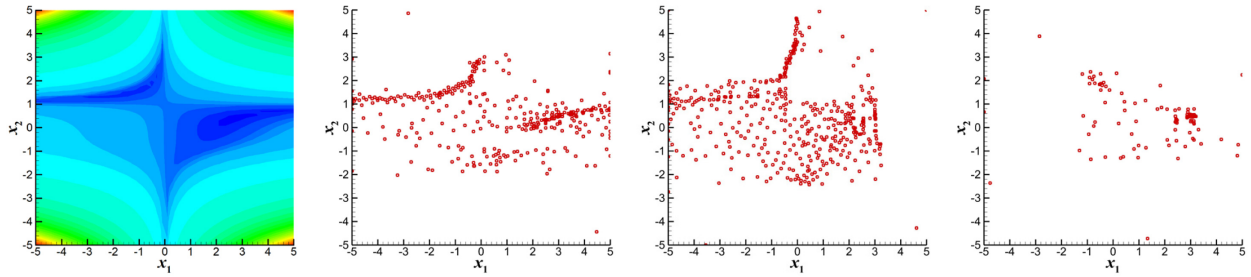
(a) Optimization of RB function (1: contour; 2: SBO-FCM; 3: SBO-EI; 4: SBO-MSP)



(b) Optimization of BR function (1: contour; 2: SBO-FCM; 3: SBO-EI; 4: SBO-MSP)



(c) Optimization of GN function (1: contour; 2: SBO-FCM; 3: SBO-EI; 4: SBO-MSP)



(d) Optimization of GF function (1: contour; 2: SBO-FCM; 3: SBO-EI; 4: SBO-MSP)

Figure 3 – Distribution of samples by different SBOs.

3.2 Applications

3.2.1 Beam optimization

The SBO-FCM is applied in solving a beam optimization problem [15] with four design variables. The objective is to minimize the vertical deflection of an I-shape beam under the bending loads of P=600 kN and Q=50 kN. The design variables are the parameters describing the cross-section shape of the beam, as is shown in Figure 4. The Length of the beam L is 200 cm, and the Young's Modulus of Elasticity of the beam E is 20,000 kN/cm^2. There are two constraints: the cross-section area $A \leqslant 300$ cm^2, and the bending stress of the cross section $\sigma \leqslant 6$ kN/cm^2. The vertical deflection can be computed by $PL^3/48EI$, and the optimization problem is expressed as

$$\begin{aligned}
\min\ & f(\boldsymbol{x}) = \frac{5000}{x_3\left(x_1-2x_4\right)^3/12 + x_2x_4^3/6 + x_2x_4\left(x_1-x_4\right)^2/2} \\
\text{s.t.}\ & g_1(\boldsymbol{x}) = 2x_2x_4 + x_3\left(x_1-2x_4\right) \le 300 \\
& g_2(\boldsymbol{x}) = \frac{180000x_1}{x_3\left(x_1-2x_4\right)^3 + 2x_2x_4\left[4x_4^2 + 3x_1\left(x_1-2x_4\right)\right]} + \frac{15000x_2}{x_3^3\left(x_1-2x_4\right) + 2x_4x_2^3} \le 6 \\
& x_1 \in [10,80], x_2 \in [10,50], x_3 \in [0.9,5], x_4 \in [0.9,5]
\end{aligned} \tag{17}$$

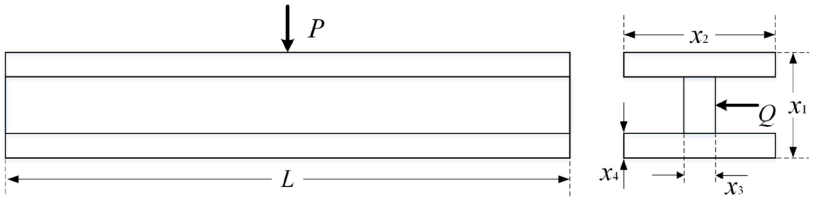


Figure 4 – Description of the beam optimization problem.

The problem in Eq. (17) is a nonlinear constrained problem, which can be solved by a variety of optimization problems. Assume that the objective function is computationally expensive to evaluate, hence SBO is chosen to search for the global optimum. According to the abovementioned policy, 15 initial samples are generated by the RLHS to start the refinement procedure. For the adaptive infilling, parameters are set as N_c=8, N_p=200 and t_r=0.6. To compare with the SBO-FCM, the differential evolution (DE) algorithm and the N-M simplex are also employed to solve the optimization problem. For DE, 20 populations and 400 generations are set; for N-M simplex, with respect to its local-search feature, two starts are set.

The results in Table 4 shows the SBO-FCM is the most efficient method, which only perform 305 function evaluations. The DE carries out 8,000 function evaluations, which is more expensive. Although the optima found by both methods can satisfy all the constraints, the bending stress by the SBO-FCM is smaller. The N-M simplex performs less function evaluations than DE, but it depends on the start. It finds a local optimum with start I, which it obtains the global optimum with start II. However, one of the constraints is violated for the global optimum. Generally, the proposed SBO-FCM is more efficient, and the adaptive infilling can benefit optimization quality.

Table 4 – Comparison of total samples, iteration cycles and optimum range by different SBOs.

Methods		optimum (cm)	cycles	samples	design variables of optimum (cm)	constraint 1 (cm^2)	constraint 2 (kN/cm^2)
SBO-FCM		0.013079	31	305	[79.9963, 49.9984, 0.9001, 2.3210]	299.9270	3.9944
DE		0.013176	400	8000	[79.9972, 45.7629, 0.9124, 2.5296]	299.9018	4.4757
N-M simplex	Start I	0.013269	346	621	[80.0000, 39.3533, 0.9004, 2.9646]	300.0267	5.3718
	Start II	0.013165	349	630	[80.0000, 44.3244, 0.9000, 2.6255]	300.0240	4.6408

3.2.2 Rocket interstage optimization

Mass reduction is usually a goal for aircraft structure design, and optimization is an efficient approach to reduce structure mass [16]. Here the mass of a rocket interstage-section with four holes is minimized. Between two shells, there are integral latticed stiffeners, consisting of axial stringers, circular stringers, and hole stringers. SHELL93 and SOLID95 elements are respectively used to model the shells and stringers. Then a minor node is defined as the intersection point of the upper end face and the axis, and it is connected with all other major nodes on the upper end face by the RBE3 elements. Hence the axis load of 1,800 kN and the bending moment of 500 kN·m are acted on the minor node. The minor node displacement is the average displacement of all the major nodes, which is exactly the displacement of the upper end face. The lower end face is fixed-supported. Detail features of the material are: density is 2,700 kg/m^3, yield strength is 313.8 MPa, Young modulus is 68,646 MPa, and Poisson ratio is 0.3.

There are four holes in the same size on the body surface of the rocket interstage-section, and the

hole stringers, the heights of the circular stringers, and the heights of the axial stringers are the same. Through sensitivity analysis, 8 design variables are defined for the optimization problem: thickness of the axial stringer, circular stringer, hole stringer are respectively C_1, C_2 and C_3; height of the loaded ring C_4; shell thickness T; stringer height H; the major and minor semi-axis are A and B. Those design variables are shown in Figure 5, and their metric is mm.

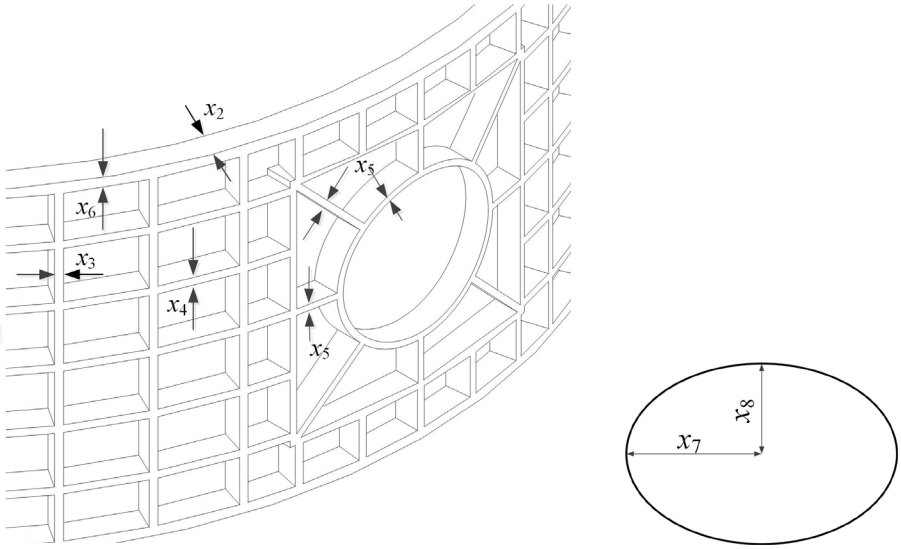


Figure 5 – Design variables of the rocket interstage-section optimization.

The optimization model of reducing interstage-section mass can be expressed as

$$\begin{aligned} &\text{find} \quad \boldsymbol{x} = [T, H, C_1, C_2, C_3, C_4, A, B]^T \\ &\min \quad w(\boldsymbol{x}) \\ &\text{s.t.} \quad \sigma(\boldsymbol{x}) \le \sigma_y \\ &\qquad s(\boldsymbol{x}) \le s_{\text{limit}} \\ &\qquad T, C_1, C_2, C_3, C_4 \in [3,15] \\ &\qquad H \in [10,20] \\ &\qquad A, B \in [150,220] \end{aligned} \tag{18}$$

in which, $w(\boldsymbol{x})$ is the total structure mass; $\boldsymbol{x}$ is the design vector; σ is the maximum equivalent stress, which should be smaller than the yield strength; s is the maximum structure displacement limited by $0.6\ \mathrm{mm}$. The other design variables are constrained by bound constraints.

The SBOs with EI-based infilling and the hybrid infilling, and the DE algorithm are utilized to compare with the proposed method. 45 samples are generated by the RLHS, and the same initial sampling is used for all the SBOs. The parameters for the SBO-FCM are set as: N_c=4, N_p=300 and t_r=0.9. For the DE, 20 populations and 100 generations are specified.

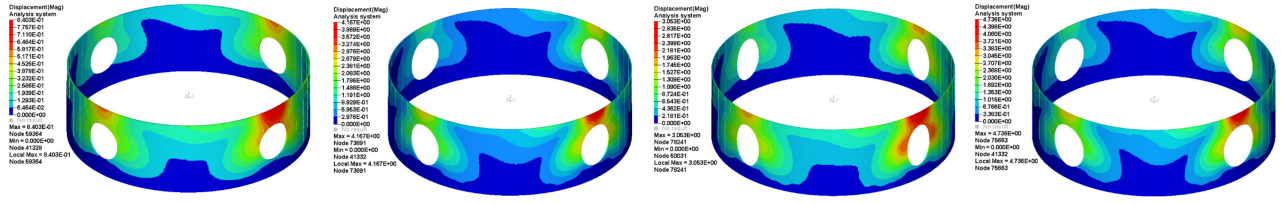
Results are compared in Table 5. The DE needs much more high-fidelity analyses; in comparison, the SBO-FCM performs less high-fidelity analyses. With respect to constraints, the constraint margin of stress is still large, which demonstrates the found minimum might be a local optimum. Thus, the parameters are reset as N_c=8, N_p=300 and t_r=0.9, and the optimization is rerun. The SBO-FCM (2) finds a better solution with a little less weight, which is closer to constraint bounds, but needs more high-fidelity analyses.

Figure 6 compares the displacements of the optimized rocket interstage-section by different methods. The optimized designs by the DE and SBO-FCM (2) have larger displacements around the holes due to weak shell and stringers. Since the momentum acted on the upper end face obeys the right-handed rule, the lower-right hole has largest displacement on its head. The optimum design by the SBO-FCM shows the largest stress around the holes, as is shown in Figure 7. The bilateral stress of the lower-right hole is close to the yield stress, which illustrates the performance of the material is

better used. Besides, the bilateral stress around the hole is higher than other regions due to less circular material.

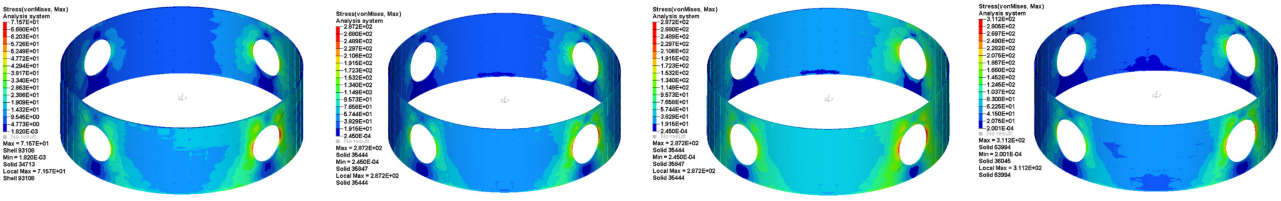
Table 5 – Optimization results for weight reduction of rocket interstage-section.

Methods	w (kg)	samples	cycles	σ (MPa)	s (mm)
Baseline	210.1840	-	-	71.57	0.1035
DE	83.9953	1864	100	287.18	0.5667
SBO-EI	84.0525	81	40	228.83	0.4924
SBO-hybrid	85.0626	243	79	295.90	0.5408
SBO-FCM	83.4198	493	100	297.14	0.5810
SBO-FCM (2)	82.4653	908	100	311.24	0.5866



(a) Baseline (b) DE (c) SBO-FCM (d) SBO-FCM (2)

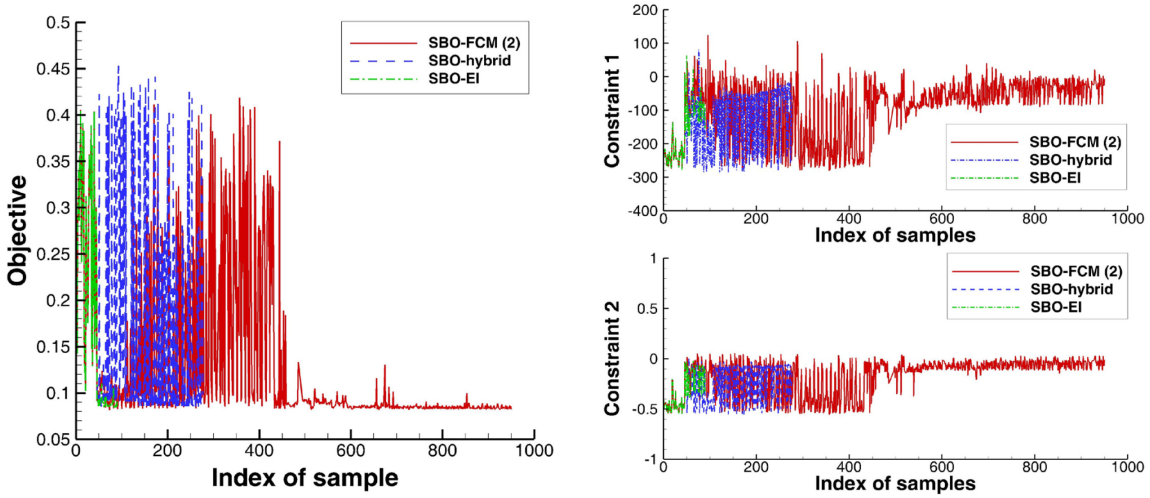
Figure 6 – Displacement contour comparison.



(a) Baseline (b) DE (c) SBO-FCM (d) SBO-FCM (2)

Figure 7 – Stress contour comparison.

Figure 8 shows the convergence histories of the objective and constraints. It illustrates that the optimization process converges quite well due to reduction of the design space. The SBO-hybrid does not converge, but terminates due to none update of the temporary optimum. The SBO-EI converges in 40 iterations due to fast converging to a local optimum. With respect to constraints, the optimum design found by the SBO-FCM is close to constraint bounds, and the constraints are satisfied.



(a) Convergence history of objective (b) Convergence histories of constraints

Figure 8 – Convergence histories from the SBO-FCM (2).

Above applications indicate that the proposed SBO-FCM is more efficient in searching global optimum for problems with expensive function evaluations, and the adaptive infilling can better balance global exploration and local exploitation.

4. Conclusions

The paper proposes an adaptive surrogate-based optimization (SBO) using the fuzzy clustering. The proposed method is validated by several benchmark test cases, and then applied to solve a beam optimization problem and a rocket interstage-section optimization problem. Following conclusions are obtained:

(1) The adaptive infilling using the fuzzy clustering can benefit both the global exploration and the local exploitation, thus it can handle complex system optimization problems with strong nonlinearity and multiple optima.

(2) By selecting infill samples in subspaces clustered, the infilling is in parallel which can further improve optimization efficiency in case of enough computer resources.

(3) The two applications indicate that the proposed SBO-FCM is more efficient than the DE and the N-M simplex in searching for the global optimum for real engineering problems.

5. Contact Author Email Address

chunnali@nwpu.edu.cn or tracy-li@hotmail.com

6. Copyright Statement

References

[1] Kenway GKW, Martins JRRA. Multipoint aerodynamic shape optimization investigations of the common research model wing. *AIAA Journal*, Vol. 54, No. 1, pp0 113-128, 2016.

[2] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. *journal of global optimization*, Vol. 13, No. 4, pp 455-492, 1998.

[3] Han ZH. Kriging surrogate model and its application to design optimization: review of recent progress. *ACTA Aeronautica et Astronautica Sinica*, Vol. 37, No. 11, pp 3197-3225, 2016.

[4] Forrester AIJ, Sóbester A, Keane AJ. *Engineering design via surrogate modeling: a practical guide*. 1st edition, American Institute of Aeronautics and Astronautics, 2008.

[5] Liu H, Ong YS, Cai J. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization*, Vol. 57, No. 1, pp 393-416, 2018.

[6] Ginsbourger BD, Riche RL, Carraro L. *Kriging is well-suited to parallelized optimization, Computational Intelligence in Expensive Optimization Problems*. Springer, 2010.

[7] Ponweiser W, Wagner T, Vincze M. Clustered multiple generalized expected improvement: a novel infill sampling criterion for surrogate models, *IEEE Congress on Evolutionary Computation*, Hong Kong, China, Vol. 1, pp 3515-3522, 2008.

[8] Li CN. *A surrogate-based framework with hybrid refinement strategies for aerodynamic shape optimization*. 1st edition, DLR-Forschungsbericht FB, 2013.

[9] Gustafsson E, Stroemberg N. Shape optimization of castings by using successive response surface methodology. *Structural and Multidisciplinary Optimization*, Vol. 35, No. 1, pp 11-28, 2008.

[10] Wang GG, and Simpson T W. Fuzzy clustering based hierarchical metamodeling for design space reduction and optimization. *Engineering Optimization*, Vol. 36, No. 3, pp 313-334, 2004.

[11] Guo X et al. RBF metamodel assisted global optimization method using particle swarm evolution and fuzzy clustering for sequential sampling. 1*5th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, Vol. 1, 2014.

[12] Ye P and Pan G. Global optimization method using ensemble of metamodels based on fuzzy clustering for design space reduction. *Engineering with Computers*, Vol. 33, No. 3, pp 573-585, 2017.

[13]Krige DG. A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, Vol. 53, No. 5, pp 159–162, 1952.

[14]Matheron G. Random functions and their applications in geology. 1st edition, Springer,1970.

[15]Long T, Guo XS, Peng L, and Liu L. Optimization strategy using dynamic radial basis function metamodel based on trust region. *Chinese Journal of Mechanical Engineering*, Vol. 50, No. 7, pp 184-190, 2014.

[16]Zhang Y, Gong C, Fang H et al. An efficient space division–based width optimization method for RBF network using fuzzy clustering algorithms. *Structural and Multidisciplinary Optimization*, Vol. 60, pp 461-480, 2019.