Structural and Multidisciplinary Optimization (2019) 59:439–460
https://doi.org/10.1007/s00158-018-2075-3

RESEARCH PAPER

A reliability-based optimization method using sequential surrogate model and Monte Carlo simulation

Xu Li[1] · Chunlin Gong[1] · Liangxian Gu[1] · Zhao Jing[2] · Hai Fang[1] · Ruichao Gao[1]

Received: 27 August 2017 / Revised: 26 June 2018 / Accepted: 14 August 2018 / Published online: 13 September 2018

Abstract

This paper presents a sequential surrogate model method for reliability-based optimization (SSRBO), which aims to reduce the number of the expensive black-box function calls in reliability-based optimization. The proposed method consists of three key steps. First, the initial samples are selected to construct radial basis function surrogate models for the objective and constraint functions, respectively. Second, by solving a series of special optimization problems in terms of the surrogate models, local samples are identified and added in the vicinity of the current optimal point to refine the surrogate models. Third, by solving the optimization problem with the shifted constraints, the current optimal point is obtained. Then, at the current optimal point, the Monte Carlo simulation based on the surrogate models is carried out to obtain the cumulative distribution functions (CDFs) of the constraints. The CDFs and target reliabilities are used to update the offsets of the constraints for the next iteration. Therefore, the original problem is decomposed to serial cheap surrogate-based deterministic problems and Monte Carlo simulations. Several examples are adopted to verify SSRBO. The results show that the number of the expensive black-box function calls is reduced exponentially without losing of precision compared to the alternative methods, which illustrates the efficiency and accuracy of the proposed method.

Keywords Expensive black box function · Radial basis function · Sequential sampling · Reliability-based optimization · Monte Carlo simulation

Nomenclature

$\mathbf{X}$	Vector of random variables
$\mathbf{x}$	Mean value of $\mathbf{X}$
$\mathbf{x}_\mathrm{L}$, $\mathbf{x}_\mathrm{U}$	Lower and upper bounds of $\mathbf{x}$
$\mathbf{U}$	Vector of random variables in standard normal space
$\mathbf{u}$	Mean value of $\mathbf{U}$
m	Number of variables
p	Number of constraint functions
$\boldsymbol{\varepsilon}$	Difference vector between $\mathbf{X}$ and $\mathbf{x}$
$\mathbf{R}$	Vector of constraint reliabilities
$\mathrm{J}(\cdot)$	Objective function
$\mathbf{g}(\cdot)$	Vector of constraint functions
$\hat{(\cdot)}$	Value of surrogate models
$(\cdot)_i$	The ith component of a vector
$E(\cdot)$	Expectation of a random variable
$P\{\cdot\}$	Probability of a random variable
CDF	Cumulative distribution function
$F_\varepsilon(\cdot)$	Vectorized CDF for ε
$F_\mathrm{U}(\cdot)$	Vectorized CDF for U
β_i	The ith reliability index of the constraint functions
$\phi(\cdot)$	CDF of the standard normal distribution
RBO	Reliability-based optimization
SSRBO	Sequential surrogate reliability-based optimization
MCS	Monte Carlo simulation
LSF	Limit state function
MPP	Most probable point
RBF	Radial basis function
AMA	Approximate moment approach
RIA	Reliability index approach

Responsible Editor: Shapour Azarm

✉ Chunlin Gong
leonwood@nwpu.edu.cn

[1] School of Astronautics, Northwestern Polytechnical University, Xi'an, People's Republic of China

[2] Department of Mechanics, Huazhong University of Science and Technology, Wuhan, People's Republic of China

PMA	Performance measure approach
SORA	Sequential optimization and reliability assessment
SLSV	Single loop single variable
ASORA	Advanced sequential optimization and reliability assessment
SLA	Single-loop approach
AHA	Adaptive hybrid approach
AH_SLM	Adaptive hybrid single-loop method

1 Introduction

In engineering design, deterministic optimization methods are used to improve the theoretical design performance. However, the optimization does not take into account the uncertainties of external loads and material properties etc., which may result in an infeasible design in practical applications. To ensure the reliability of the design, the safety factor is adopted in engineering, nevertheless, it is mainly based on the engineering experience which is inaccurate and not always available (Qu and Haftka 2004; Breitkopf and Coelho 2010). More recently, reliability-based optimization (RBO) with random variables has received much attention from the engineering community. With RBO, it is possible to get more accurate designs to meet the reliability requirements. However, this is a nested optimization problem including optimization and reliability analysis (or assessment), which faces the challenge of huge computational cost. To solve the problem, different approximate approaches have been developed in the past decades (Breitkopf and Coelho 2010; Tsompanakis et al. 2010; Valdebenito and Schuoller 2010).

As a typical method for RBO, the approximate moment approach (AMA) (Koch et al. 2004; Tsompanakis et al. 2010) transforms the reliability constraints into approximated deterministic constraints with the Taylor series expansion. AMA is very efficient since it does not require an extra cost for reliability analysis. However, this method implies many simplifications and cannot obtain accurate reliability result. In addition, the main disadvantage lies in the assumption that the random variables are normally distributed, which is not practical for most engineering structures. Consequently, the application field of AMA is limited.

To improve the accuracy of AMA, the reliability index approach (RIA) (Enevoldsen and Sørensen 1994; Grandhi and Wang 1998; Tu et al. 1999; Tsompanakis et al. 2010) was proposed. It converts the reliability constraints, at the current design point, into the constraints of reliability indexes by the first-order reliability method (FORM) (Verderaime 1994; Haldar and Mahadevan 1995; Cawlfield 2000; Du 2008) or the second-order reliability method (SORM) (Cizelj et al. 1994; Cawlfield 2000), which transforms the reliability-based optimization problem into a relatively simple deterministic optimization problem. However, in the process of the reliability analysis, the searching of the most probable point (MPP) needs extra optimization and expensive function calls. Thus, in the whole optimization process, RIA is still computationally expensive. Moreover, FORM and SORM have a poor accuracy for the higher nonlinear constraints. Therefore, RIA is suitable for the problems with low nonlinearity constraints.

To further reduce the computational cost of RBO, the performance measurement approach (PMA) (Youn et al. 2004; Tsompanakis et al. 2010) was proposed. This method transforms the reliability constraint into deterministic constraints by solving the optimization problem of inverse MPP, which is easier with respect to the searching process of MPP. Youn and Choi (2004) showed that PMA is more efficient, stable and less dependent on probabilistic distribution types than RIA. However, the reliability analysis of PMA is embedded into the optimization loop, which means PMA still requires a large number of function calls, due to the two-level optimization.

To avoid the computationally expensive two-level optimization, Du and Chen (2004) proposed the sequential optimization reliability assessment (SORA) method, which decouples the RBO into serial deterministic optimizations with shifted constraints and reliability assessments (or analyses). Within one iteration circle of SORA, two optimization procedures are included: the optimization with the modified constraints and the optimization searching for the inverse MPPs. SORA decomposes a two-level optimization to a series of single-level optimizations, which greatly reduces the computational cost. Yi et al. (2016) presented an approximate sequential optimization reliability assessment (ASORA) method, which is reported more efficient than the original SORA. However, such methods require a series of optimizations based on the computationally expensive black-box functions. Meanwhile, the methods based on the inverse MPPs have large errors on the boundary with high nonlinear constraints.

More recently, some modified methods based on the RIA, PMA, and SORA were suggested (Li et al. 2013; Yi et al. 2016). In addition, as a different methodology of improving efficiency, the surrogate models (also named meta-model or response surface method) were used to approximate the objective and constraint functions within the local or global region of interest (Li et al. 2016; Meng et al. 2017; Strömberg 2017). Jiang et al. (2017) proposed an adaptive hybrid single-loop method (AH_SLM) to search the MPP more efficiently and alleviate the oscillation in the search process. Doan et al. (2018) proposed an efficient approach for RBO combining SORA with radial basis function. Zhou et al. (2018) presented an enhanced version of single-loop approach (SLA) (Liang et al. 2008) where the adaptive surrogate models and screening strategies are adopted in the RBO process. Several representative RBO methods are introduced above, and more methods, such as adaptive hybrid approach (AHA) and single loop single variable (SLSV) etc., can be found in reviews of the literatures (Yang and Gu 2004; Tsompanakis et al. 2010; Jiang et al. 2017). However, most reliability analyses of the existing methods are based on the FORM, SORM, which also inherits the disadvantages of the accuracy and efficiency.

In this article, a new RBO method based on sequential radial basis function (RBF) and Monte Carlo simulation is proposed to reduce the computational cost. Assume that the optimal point for the RBO is near the deterministic optimal point, and then surrogate models are adopted to improve the local accuracy of the important region (the neighborhood of the deterministic optimal point) to reduce the computational cost. After the local surrogate models of the objective and constraint functions are constructed iteratively, Monte Carlo simulations based on the surrogate models are carried out at the deterministic optimal point, and then the cumulative distribution functions (CDF) of the constraints are obtained. Therefore, the quantiles of the prescribed reliabilities can be obtained by the inverse CDF as the offsets of the deterministic constraints. The offsets generate the new equivalent reliability constraints. The optimal point is then added to the sample set to update the surrogate model for a new loop until the offsets convergence.

The remainder of this article is structured as follows. In Section 2, the formulation and several typical methods of RBO are introduced. In Section 3, the surrogate model technology used in this paper is introduced, and then the detailed process of the proposed method is described and discussed. In Section 4, several numerical examples are used to validate the efficiency and accuracy of the proposed method. Finally, conclusions are given in Section 5.

2 Overview of reliability-based design optimization

In this section, the mathematical description of RBO is introduced, and then several typical RBO methods are described in uniform formulations. Finally, some features of the existing methods are discussed to verify why the proposed method is better than the existing methods.

2.1 Formulation of RBO problem

Reliability-based optimization is the extension of the deterministic optimization problem, in which the variables with uncertainties and the reliability constraints are included. Assuming $X_i = x_i + \varepsilon_i (i = 1, 2, \cdots, m)$, where X_i is the ith random variable; x_i is the mean value of X_i; ε_i is the difference between X_i and x_i, thus its expected value is written as $\mathrm{E}(\varepsilon_i) = \mathrm{E}(X_i - x_i) = 0$. The general formulation of a deterministic optimization, which does not consider uncertainties, is given by

$$\min_{\mathbf{x}} \ \mathrm{J}(\mathbf{x}) \qquad \text{s.t.} \ \begin{cases} \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x}_\mathrm{L} \leq \mathbf{x} \leq \mathbf{x}_\mathrm{U} \end{cases} \tag{1}$$

Thus, a typical formulation of RBO problem can be expressed as follows (Valdebenito and Schuoller 2010; Yi et al. 2016)

$$\min_{\mathbf{x}} \ \mathrm{E}[\mathrm{J}(\mathbf{x} + \boldsymbol{\varepsilon})] \approx \mathrm{E}\left[\mathrm{J}(\mathbf{x}) + \frac{\partial \mathrm{J}(\mathbf{x})}{\partial \mathbf{x}^\mathrm{T}} \boldsymbol{\varepsilon}\right] = \mathrm{J}(\mathbf{x}) \tag{2}$$

$$\text{s.t.} \ \begin{cases} \mathrm{P}\{\mathbf{g}(\mathbf{x} + \boldsymbol{\varepsilon}) \leq \mathbf{0}\} \geq \mathbf{R} \\ \mathbf{x}_\mathrm{L} \leq \mathbf{x} \leq \mathbf{x}_\mathrm{U} \end{cases}$$

where $\mathbf{x} \in \mathbb{R}^m$ is the mean value vector of design variables; $\mathbf{x}_\mathrm{L}$ and $\mathbf{x}_\mathrm{U}$ are the lower and upper bounds of $\mathbf{x}$, respectively; $\boldsymbol{\varepsilon}$ is the vector of the new random variables; m is the number of the random variables; J ($\mathbf{x}$) denotes the objective function; $\mathbf{g}(\cdot)$ is the vector of the constraint limit state functions; $\mathbf{R} \in \mathbb{R}^p$ is the vector of the prescribed target reliabilities, and p is the number of the constraints. In what follows, several typical RBO methods are introduced.

2.2 Typical RBO methods

2.2.1 Approximate moment approach

The approximate moment approach (AMA) (Tsompanakis et al. 2010) approximates the objective and constraint functions at the mean value $\mathbf{x}$ with a Taylor series expansion. The vector of constraint functions with uncertainties can be approximated as

$$\mathbf{g}(\mathbf{x} + \boldsymbol{\varepsilon}) \approx \mathbf{g}(\mathbf{x}) + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}^\mathrm{T}} \boldsymbol{\varepsilon} \tag{3}$$

Assuming the components of $\boldsymbol{\varepsilon}$ are independent and normally distributed, according to the additivity of independent normal distributions, the components of $\mathbf{g}(\mathbf{x} + \boldsymbol{\varepsilon})$ are also normally distributed. Therefore, the RBO problem is transformed into

$$\min_{\mathbf{x}} \ \mathrm{J}(\mathbf{x})$$

$$\text{s.t.} \ \begin{cases} \mathrm{g}_i(\mathbf{x}) + \sqrt{\dfrac{\partial \mathrm{g}_i(\mathbf{x})}{\partial \mathbf{x}^\mathrm{T}} \operatorname{cov}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}) \dfrac{\partial \mathrm{g}_i(\mathbf{x})}{\partial \mathbf{x}}} \cdot \phi^{-1}(R_i) \leq \mathbf{0} \\ \mathbf{x}_\mathrm{L} \leq \mathbf{x} \leq \mathbf{x}_\mathrm{U}, \quad i = 1, 2, \cdots, p \end{cases} \tag{4}$$

where $\phi^{-1}(\cdot)$ is the inverse cumulative distribution function (CDF) of the standard normal distribution, $\phi^{-1}(R_i)$ is the reliability index for the ith constraint function, p is the number of the constraint functions, $\operatorname{cov}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})$ is the covariance matrix of the random vector $\boldsymbol{\varepsilon}$, and the row vector $\partial \mathrm{g}_i(\mathbf{x}) / \partial \mathbf{x}^\mathrm{T}$ is the gradient of the ith constraint function $\mathrm{g}_i(\mathbf{x})$.

2.2.2 Reliability index approach

In the reliability index approach (RIA) (Tsompanakis et al. 2010), the constraints of the probability reliabilities are transformed into the constraints of reliability indexes. Therefore, the RBO problem is converted to

$$\min_{\mathbf{x}} \ \mathrm{J}(\mathbf{x})$$
$$\text{s.t.} \ \begin{cases} \beta_i(\mathbf{x}) = \left[\min\limits_{\mathbf{u}} \ \|\mathbf{u}\| \quad \text{s.t.} \ \mathrm{g}_i\left(\mathbf{x}+\mathbf{F}_{\boldsymbol{\varepsilon}}^{-1}\mathbf{F}_{\mathrm{U}}(\mathbf{u})\right)=0\right] \\ \beta_i(\mathbf{x}) \geq \phi^{-1}(R_i), \quad i=1,2,\cdots,p \\ \mathbf{x}_{\mathrm{L}} \leq \mathbf{x} \leq \mathbf{x}_{\mathrm{U}} \end{cases} \tag{5}$$

where $\beta_i(\mathbf{x})$ is the *i*th reliability index at the mean value point $\mathbf{x}$. The function $\phi^{-1}(\cdot)$ is the inverse CDF of the standard normal distribution. $\mathbf{F}_{\boldsymbol{\varepsilon}}^{-\mathbf{1}}(\boldsymbol{\varepsilon}) = \left[\mathrm{F}_{\varepsilon_1}^{-1}(\varepsilon_1), \mathrm{F}_{\varepsilon_2}^{-1}(\varepsilon_2), \cdots, \mathrm{F}_{\varepsilon_m}^{-1}(\varepsilon_m)\right]^{\mathrm{T}}$ is the inverse CDF vector of the random vector $\boldsymbol{\varepsilon}$, and $\mathbf{F}_{\mathbf{U}}(\mathbf{u}) = [\mathrm{F}_{U_1}(u_1), \mathrm{F}_{U_2}(u_2), \cdots, \mathrm{F}_{U_m}(u_m)]^{\mathrm{T}}$ is the CDF vector of the standard normal distribution.

2.2.3 Performance measure approach

In the performance measure approach (PMA) (Youn et al. 2004; Tsompanakis et al. 2010), a reliability constraint is converted into a deterministic constraint such that the inverse most probable point (IMPP) is feasible. IMPP is the point which minimizes the limit state function subjected to the prescribed reliability constraint in **U**-space. The RBO problem can be formulated as follows:

$$\min_{\mathbf{x}} \ \mathrm{J}(\mathbf{x})$$
$$\text{s.t.} \ \begin{cases} \alpha_i(\mathbf{x}) = \left[\min\limits_{\mathbf{u}} \ \mathrm{g}_i\left(\mathbf{x}+\mathbf{F}_{\boldsymbol{\varepsilon}}^{-1}\mathbf{F}_{\mathrm{U}}(\mathbf{u})\right) \ \text{s.t.} \ \|\mathbf{u}\| = \phi^{-1}(R_i)\right] \\ \alpha_i(\mathbf{x}) \geq 0, \quad i=1,2,\cdots,p \\ \mathbf{x}_{\mathrm{L}} \leq \mathbf{x} \leq \mathbf{x}_{\mathrm{U}} \end{cases} \tag{6}$$

where, $\alpha_i(\mathbf{x})$ is the value of the limit state function at the IMPP, and the other symbols are the same as the those described in the previous sections.

2.2.4 Sequential optimization and reliability assessment

The method of sequential optimization and reliability assessment (SORA) (Du and Chen 2004; Yi et al. 2016) transforms the two-level optimization into a sequence of deterministic optimizations and reliability analyses. For each loop, the performance measure is performed after the current deterministic optimization. The new value of the performance measure is then used in the next loop as a limit state constraint function. The RBO problem can thus be formulated as

$$\begin{cases} \mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \ \mathrm{J}(\mathbf{x}) \\ \text{s.t.} \ \begin{cases} \mathrm{g}_i\left(\mathbf{x}-\boldsymbol{\xi}_k^i\right) \leq \mathbf{0} \\ \mathbf{x}_{\mathrm{L}} \leq \mathbf{x} \leq \mathbf{x}_{\mathrm{U}} \end{cases} \boldsymbol{\xi}_{k+1}^i = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \ \mathrm{g}_i(\mathbf{x}_{k+1}+\boldsymbol{\xi}) \\ \text{s.t.} \ \left\|\mathbf{F}_{\mathbf{U}}^{-1}\mathbf{F}_{\boldsymbol{\varepsilon}}(\boldsymbol{\xi})\right\| = \phi^{-1}(R_i) \\ \text{Start}: \boldsymbol{\xi}_0^i = 0 \\ i=1,2,\cdots,p \end{cases} \tag{7}$$

The flowchart of SORA is shown in Fig. 1. This formulation solves the deterministic problem with a classical optimization algorithm. However, the whole process requires a series of optimizations, which reduces the efficiency of the approach.

2.3 Discussion of RBO methods

The computational complexity of RBO comes from three aspects: first, the analysis model is time-consuming; second, the optimization process requires a large number of function calls; and third, the reliability analysis process requires a large number of function calls to search for the MPPs with a sub-optimization process. Therefore, methods to reduce the cost of computation are based on these three aspects. AMA just simplifies the reliability analysis process of the constraint functions through the Taylor series expansion, and transforms the reliability optimization problem into a deterministic optimization problem, but brings a loss of precision. Both RIA and PMA solve nested optimization problem based on the original time-consuming models, therefore the number of function calls of the original models is large. SORA avoids the nested optimization process, but it is still based on the original time-consuming models, which still requires a large number of function calls during the optimization process. In

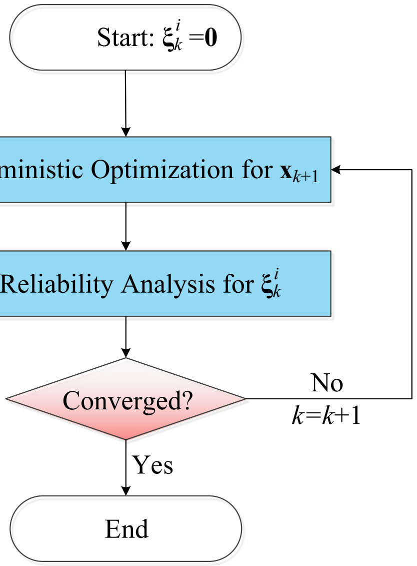

Start: $\boldsymbol{\xi}_k^i = \mathbf{0}$
Deterministic Optimization for $\mathbf{x}_{k+1}$
Reliability Analysis for $\boldsymbol{\xi}_k^i$
Converged?
No
$k=k+1$
Yes
End

Fig. 1 Flowchart of the SORA method

order to avoid the time-consuming function calls, we use surrogate models instead of original time-consuming target and constraint functions based on SORA. Meanwhile, since the result of RBO is near the deterministic optimal point, we can increase the local accuracy near the deterministic optimal point, which reduces the number of time-consuming model evaluations further. In order to avoid the approximate error caused by FORM and SORM in the reliability analysis process, Monte Carlo simulation can be used for reliability analysis. As the surrogate model is very fast, the time consumed by a direct Monte Carlo simulation is relatively small compared to a single evaluation of the time-consuming model.

3 Sequential surrogate models for reliability-based optimization

As discussed in Section 2, the existing methods perform Taylor expansion of the limit state function in **X** or **U** space, which only use the local information of the expansion point. When the nonlinearity of the limit state function is strong, the approximation error is large. However, surrogate models contain all the sample information and have a better approximation accuracy for the limit state function in the failure region. In addition, since the time-consuming optimization processes are performed based on the cheap surrogate models, a large number of function evaluations are avoided and the computational efficiency is improved.

In this section, the surrogate techniques used in the proposed method are introduced, and then the details of the process are described and discussed.

3.1 Surrogate model

A surrogate model $\hat{y}(\mathbf{x})$ is an approximate prediction model of a complex or unknown model $y(\mathbf{x})$ with a set of input-output training samples $S = \{(\mathbf{x}_i, y_i)|i = 1, 2, \cdots, n\}$. Instinctually, the surrogate model is an interpolation or regression model, belonging to a branch of machine learning (Hastie et al. 2008). Common surrogate models include polynomial response surface method (PRSM) (Forrester et al. 2008), radial basis function (RBF) (Regis and Shoemaker 2005; Forrester et al. 2008), Kriging (Laurenceau and Sagaut 2008; Ronch et al. 2011), support vector regression (SVR) (Forrester et al. 2008) and artificial neutral net (ANN) (Hurtado and Alvarez 2001; Forrester et al. 2008).

Since the proposed method requires an accurate local surrogate model near the deterministic optimal point, it is necessary for the surrogate model to go through the sample points. PRSM is a global regression method with poor local accuracy, so it is not appropriate here. The Kriging and ANN methods are promising surrogate models, but due to too many model parameters, the training process is time-consuming. SVR is generally regarded as a special form of support vector machine (SVM) in most of the literature. SVM generally refers to classification, while SVR refers to model prediction or regression.

RBF is considered as a special form of SVR (Forrester et al. 2008). The model is more flexible (it can determine the number of parameters according to the needs). It is a method based on other methods and is easy to implement. This is why RBF always has its place when discussing the surrogate models. Moreover, RBF has two advantages: first, it has strong nonlinear adaptability and goes through the samples, which illustrates RBF has good local accuracy; second, it has only one hyperparameter, which means less time for model construction and parameter optimization. It is worth noting that when the original model itself is very time-consuming, the time of training model is negligible, and RBF can also be replaced with SVR, Kriging and ANN. This article uses RBF to construct the sequential surrogate model. RBF uses a linear combination of radial basis functions to approximate the expensive black-box function, and the general expression of RBF is given by

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{n} \beta_i \mathrm{f}(\|\mathbf{x}-\mathbf{x}_i\|) = \mathbf{f}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta} \tag{8}$$

where $\mathbf{x} \in \mathbb{R}^m$, m is the number of the variables; n is the number of the samples; β_i is the ith component of the coefficient vector $\boldsymbol{\beta}$; $\mathrm{f}(\|\mathbf{x}-\mathbf{x}_i\|)$ is the ith component of the radial basis function vector with the common forms shown in Table 1. Assume r to be the Euclidean distance between two points and c is the shape parameter of the radial basis function.

Substitute the samples into Eq. (8),

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \mathrm{f}(\|\mathbf{x}_1-\mathbf{x}_1\|) & \mathrm{f}(\|\mathbf{x}_1-\mathbf{x}_2\|) & \cdots & \mathrm{f}(\|\mathbf{x}_1-\mathbf{x}_n\|) \\ \mathrm{f}(\|\mathbf{x}_2-\mathbf{x}_1\|) & \mathrm{f}(\|\mathbf{x}_2-\mathbf{x}_2\|) & \cdots & \mathrm{f}(\|\mathbf{x}_2-\mathbf{x}_n\|) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{f}(\|\mathbf{x}_n-\mathbf{x}_1\|) & \mathrm{f}(\|\mathbf{x}_n-\mathbf{x}_2\|) & \cdots & \mathrm{f}(\|\mathbf{x}_n-\mathbf{x}_n\|) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} \tag{9}$$

And Eq. (9) can be denoted by a matrix form,

$$\mathbf{y} = \mathbf{F}\boldsymbol{\beta} \tag{10}$$

As $\mathbf{F} \in \mathbb{R}^{n \times n}$ is a non-singular matrix, Eq. (10) has a unique solution $\boldsymbol{\beta} = \mathbf{F}^{-1}\mathbf{y}$. Thus the prediction model is given by

$$\hat{y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^{\mathrm{T}}\mathbf{F}^{-1}\mathbf{y} \tag{11}$$

Table 1 Radial basis functions

Function type	$f(r)$
Gaussian	$\exp(-cr^2)$
Multi-quadric (MQ)	$(1+cr^2)^{1/2}$
Inverse multi-quadric (IMQ)	$(1+cr^2)^{-1/2}$
Thin plate spline	$r^2 \ln(1+cr^2)$

where $\mathbf{f}(\mathbf{x})$ is the radial basis function vector for the prediction point, and hence the prediction model is determined by the prediction point $\mathbf{x}$ and the training sample set S. It should be pointed out that the shape parameter c, which has a great influence on the accuracy of the model, is included in $\mathbf{f}(\mathbf{x})$ and $\mathbf{F}$. In general, c is determined by experience or other optimization criteria. This paper uses the cross-validation criteria to optimize the shape parameter c.

3.2 Validation of the surrogate model

A common method to validate the accuracy of the surrogate model is the root mean square error (RMSE) with another validation set (Ronch et al. 2017). However, there is not always enough samples available in engineering. Therefore, the cross-validation (CV) method is adopted here (Hastie et al. 2008). The samples are divided into K roughly equal-sized parts. For the kth ($k = 1, 2, \cdots, K$) part, the model is constructed with the other $K - 1$ parts of the samples, and calculates the prediction error of the approximate model when predicting the kth part of the samples. Cross-validation can fully reflect the matching degree between the samples and the model. In particular, when K is equal to the sample size n, it is called leave-one-out cross-validation error (LOOCV). Thus the RBF shape parameter c can be estimated by the following sub-optimization problem:

$$\min_{c} \ \text{LOOCV}(c) = \sum_{i=1}^{n} \left[y_i - \hat{y}(\mathbf{x}_i; S - \{(\mathbf{x}_i, y_i)\})\right]^2 \qquad (12)$$

In Eq. (12), it can be seen that the evaluation of LOOCV error requires n times construction of the surrogate model. However, the LOOCV error does not require additional verification points, which is capabl1e of describing the matching degree between the samples and the prediction model. According to the surrogate model and obtained the shape parameter c, a sequential surrogate model for RBO can be constructed.

3.3 Process of the proposed method

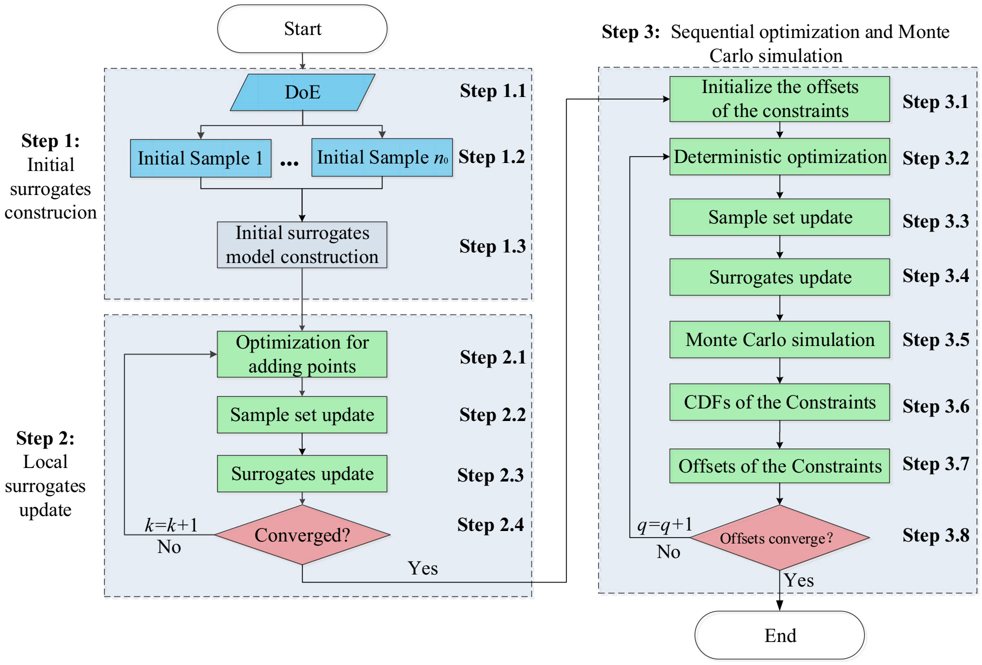
The method of sequential surrogate model for reliability-based design optimization (SSRBO) constructs the RBF surrogate models of the objective and constraint functions with the initial training samples. Then, a series of optimizations for adding points is performed to update the surrogate models in important regions (the neighborhood of the determined optimum point). When the refined local surrogate models are constructed, the Monte Carlo simulation is carried out at the determined optimal point by using the surrogate model to obtain the cumulative probability density functions (CDF) of the limit state functions (Bowman and Azzalini 1997). Thus the offset of the constraints can be obtained with the inverse CDFs.

The flowchart of the SSRBO method is shown in Fig. 2, and the more detailed process is described as follows.

Step 1: Initial surrogate construction.

Step 1.1: Select the initial input samples $\mathbf{x}_i (i = 1, 2, \cdots, n_0)$ with the design of experiment (DoE) method Latin hypercube sampling (LHS). There are two reasons for choosing LHS. First, LHS can obtain well-distributed and representative

Fig. 2 Flowchart of the SSRBO method



design points with less cost from the design space, so as to obtain accurate model information more effectively. Second, LHS has the freedom to define the number of sample points, thereby providing greater flexibility and broader applicability. The number of the initial samples is chosen as $n_0 = 2m + 1$, where m is the dimension of the design variables.

Step 1.2: Evaluate the input samples with the expensive black-box model to obtain the response values of the objective J ($\mathbf{x}_i$) and constraint vector $\mathbf{g}$ ($\mathbf{x}_i$). Thus, the initial sample set S_0 is given by

$$S_0 = \{(\mathbf{x}_i, \mathrm{J}(\mathbf{x}_i), \mathbf{g}(\mathbf{x}_i))|i = 1, 2, \cdots, n_0\} \quad (13)$$

Step 1.3: Use the initial sample set S_0 to construct the surrogate objective function $\hat{\mathrm{J}}(\mathbf{x}|S_0)$ and the surrogate constraint vector $\hat{\mathbf{g}}(\mathbf{x}|S_0)$. Then the shape parameters of the surrogate models are optimized with LOOCV (See Section 3.2). Here the surrogate models are constructed with RBF, which strictly goes through the samples and has strong nonlinear adaptability. The RBF code is complemented with the authors' in-house MATLAB toolbox.

Step 2: Local surrogates update. This step is an iteration process as follows:

$$\begin{cases} \mathbf{x}_k^* = \underset{\mathbf{x}}{\mathrm{argmin}}\hat{\mathcal{J}}\,(\mathbf{x}|S_k) \ \ \text{s.t.}\begin{cases} \hat{\mathbf{g}}(\mathbf{x}|S_k)\le\mathbf{0} \\ \min\limits_{i=1}^{|S_k|} \ \ \|\mathbf{x}-\mathbf{x}_i\|\ge d_{\min} \\ \mathbf{x}_{\mathrm{L}}\le\mathbf{x}\le\mathbf{x}_{\mathrm{U}} \end{cases} S_{k+1} = S_k\cup\left\{\left(\mathbf{x}_k^*, \mathrm{J}\left(\mathbf{x}_k^*\right), \mathbf{g}\left(\mathbf{x}_k^*\right)\right)\right\}\text{Start : } \ S_0\text{Stop : }k\ge k_{\max} \ \text{ or } \ \left\|\mathbf{g}\left(\mathbf{x}_k^*\right)-\hat{\mathbf{g}}\left(\mathbf{x}_k^*|S_k\right)\right\| < 1.0\times10^{-4} \end{cases} \quad (14)$$

where, S_k is the set with k added samples; $\hat{\mathrm{J}}(\mathbf{x}|S_k)$ and $\hat{\mathbf{g}}(\mathbf{x}|S_k)$ are the surrogate objective function and surrogate constraint vector constructed with S_k; $\mathbf{x}_k^*$ is the kth added sample; $d_{\min}$ is the minimum distance. In order to facilitate the description of the following steps, assume the total number of the samples added in this step is n_1.

Step 2.1: Solve the optimization problem of Eq. (14) to find the new point $\mathbf{x}_k^*$. As the optimization problem has an inequality constraint of minimum distance which is not differentiable, gradient-based optimization algorithm, such as sequential quadratic programming (SQP), cannot be utilized here. Therefore, the genetic algorithm (GA) which is not restricted by the differentiability is adopted here. The optimization problem is solved with the MATLAB optimization toolbox.

Step 2.2: Evaluate the new point to obtain the response values of the objective and constraint functions, and then update the sample set with $S_{k+1} = S_k\cup\left\{\mathbf{x}_k^*, \mathrm{J}\left(\mathbf{x}_k^*\right), \mathbf{g}\left(\mathbf{x}_k^*\right)\right\}$.

Step 2.3: Reconstruct the surrogate models with the updated sample set S_{k+1}, and then optimize the model shape parameters. This step is similar to that of Step 1.3.

Step 2.4: Convergence check. If one of the termination criterions, (a) the number of added points reaches the maximum number($k = k_{\max}$), (b) the difference between the real constraint vector and the surrogate constraint vector is below a target threshold ($\left\|\mathbf{g}\left(\mathbf{x}_k^*\right) -\hat{\mathbf{g}}\left(\mathbf{x}_k^*|S_k\right)\right\| \le 1.0\times 10^{-4}$), is satisfied, go to Step 3, otherwise, set $k = k + 1$ and then go to Step 2.1.

Step 3: Sequential optimization and Monte Carlo simulation. The key process is given by

$$\begin{cases} \mathbf{x}_q^* = \underset{\mathbf{x}}{\mathrm{argmin}}\hat{\mathcal{J}}\,(\mathbf{x}|S_q) \\ \text{s.t.}\begin{cases} \hat{g}_i(\mathbf{x}|S_q) + \Delta_q^i\le\mathbf{0} \\ \mathbf{x}_{\mathrm{L}}\le\mathbf{x}\le\mathbf{x}_{\mathrm{U}} \end{cases} \\ \Delta_{q+1}^i = \mathrm{CDF}_{q,i}^{-1}\left(R_i|S_q, \mathbf{x}_q^*\right)\text{-}\hat{g}_i\left(\mathbf{x}_q^*|S_q\right) \\ i = 1, 2, \cdots, p \\ S_{q+1} = S_q\cup\left\{\left(\mathbf{x}_q^*, \mathrm{J}\left(\mathbf{x}_q^*\right), \mathbf{g}\left(\mathbf{x}_q^*\right)\right)\right\} \\ \text{Start : }\Delta_{q_0}^i = 0, \ \ q_0 = n_0 + n_1 + 1 \\ \text{Stop : }q\ge q_{\max} \ \text{ or } \ \left\|\boldsymbol{\Delta}_{q+1}-\boldsymbol{\Delta}_q\right\|\le1.0\times10^{-3} \end{cases} \quad (15)$$

where, $\mathbf{x}$ is the expected mean value of the random vector $\mathbf{X}$; $\Delta_q^{\boldsymbol{i}}$ is the ith constraint offset in the qth iteration; R_i denotes the ith prescribed target reliability; $\mathrm{CDF}^{-1}(\cdot)$ is the inverse cumulative distribution function, which is determined by 1×10^6 Monte Carlo simulations (MCS) with the surrogate constraint functions and the function "fitdist" and "icdf" in Matlab. Furthermore, the process of the constraint shifting is shown in Fig. 3. The process is similar to that of AMA in Eq. (4), however, the offset variable Δ is not based on the gradient and normal distribution but is based on the CDF of MCS, which captures more failure information of the constraint.

Step 3.1: Set the initial offsets of the constraint functions as $\Delta_{q_0}^{i}=0, (i = 1, 2, \cdots, p)$. In fact, when $\Delta_{q_0}^{i}=0$, the optimization problem in Eq. (15) is approximately equivalent to the original deterministic optimization problem.

Step 3.2: Solve the optimization problem with the shifted constraint functions in Eq. (15) to search for the current optimum. Since the offsets have great influence on the result of the optimization problem, the optimization algorithm with high accuracy is required. In order to balance the global search ability and local search accuracy, a search strategy combing GA and SQP is adopted. In this strategy, GA finds an approximated global optimum, and then SQP uses the optimum as the initial value to find a more accurate optimization point as the current optimum $\mathbf{x}_q^*$.

Step 3.3: Evaluate the current optimum to obtain the response values of the objective and constraint functions, and then update the sample set with $S_{q+1} = S_q \cup \left\{ \left(\mathbf{x}_q^*, \mathrm{J}\left(\mathbf{x}_q^*\right), \mathbf{g}\left(\mathbf{x}_q^*\right) \right) \right\}$.

Step 3.4: Reconstruct the surrogate models with the updated sample set. However, it is not necessary to optimize the shape parameters of the surrogate models, because the added point varies within a smaller range around the deterministic optimum.

Step 3.5: Generate random samples $\mathbf{x}_j (j = 1, 2, \cdots, 1 \times 10^6)$ with the given probability density function (PDF) of random vector $\mathbf{X}$, the expected value of which is $\mathrm{E}(\mathbf{X}) = \mathbf{x}_q^*$. Then evaluate the response values of the surrogate objective $\hat{\mathrm{J}}\left(\mathbf{x}_j\right)$ and surrogate constraint vector $\hat{\mathbf{g}}\left(\mathbf{x}_j\right)$.

Step 3.6: Fit cumulative distribution functions (CDFs) of the constraints with the 1×10^6 surrogate samples $\hat{\mathbf{g}}\left(\mathbf{x}_j\right)$ $\left(j = 1, 2, \cdots, 1 \times 10^6\right)$. Therefore, the surrogate CDF of each constraint function at the current optimum $\mathbf{x}_q^*$ can be formulated as $\mathrm{CDF}_{q,i}\left(\cdot | S_q, \mathbf{x}_q^*\right)$ $(i = 1, 2, \cdots, p)$. It can be complemented with the "fitdist" function in Matlab.

Step 3.7: Calculate offsets of the constraints. As the target reliability of each constraint is $R_i (i = 1, 2, \cdots, p)$ and the

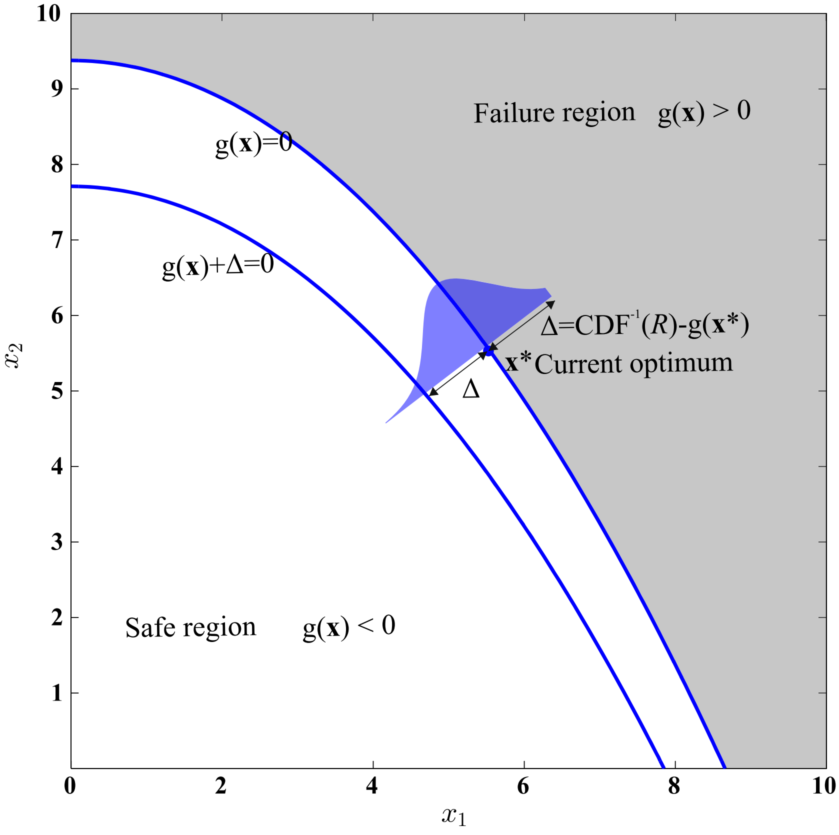
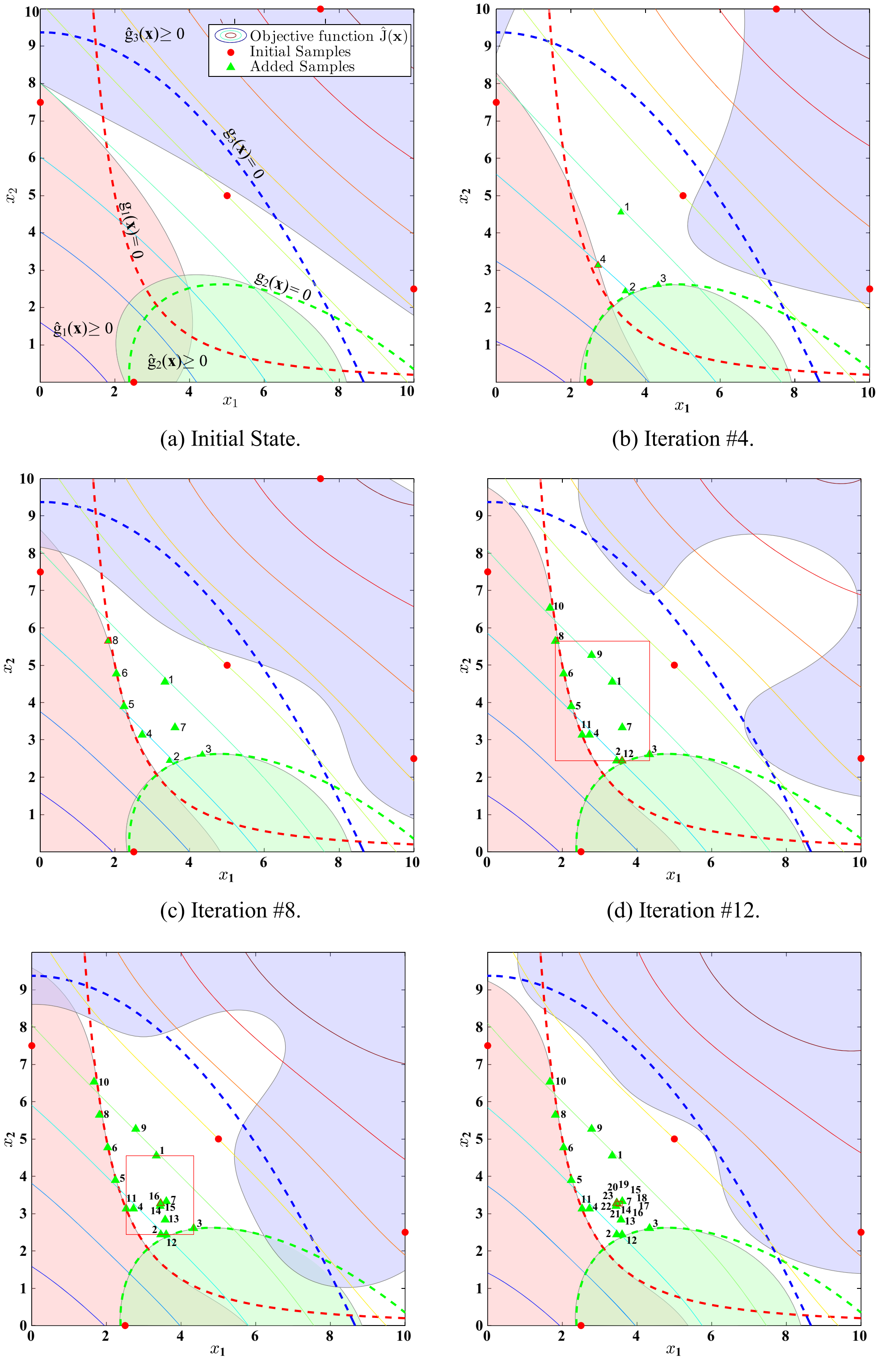


Fig. 3 The constraint offset of the SSRBO method, where x_1 and x_2 are the expectations of the random variables, respectively

Fig. 4 Iteration of adding samples for the 2D problem



(a) Initial State.

(b) Iteration #4.

(c) Iteration #8.

(d) Iteration #12.

(e) Iteration #16.

(f) Iteration #23.

surrogate CDF of each constraint function is obtained in step 3.7, the offset of each constraint is given by $\Delta^i_{q+1} = \mathrm{CDF}^{-1}_{q,i}\left(R_i|S_q, \mathbf{x}^*_q\right)$ -$\hat{g}_i\left(\mathbf{x}^*_q|S_q\right)$.

Step 3.8: Offsets convergence check. If the number of the added samples reaches the maximum value($q \geq q_{\max}$) or the error between two adjacent offset vectors is below a certain threshold ($\|\Delta_{q+1} - \Delta_q\| \leq 1.0 \times$

10^{-3}), go to the end of SSRBO, otherwise, go to Step 3.2. Repeat Step 3.2 to Step 3.8 until the stopping criterion is satisfied.

In the process of SSRBO, the surrogate models are refined by adding samples in the neighborhood of the deterministic optimum sequentially, which improves the local precision in the important region with high failure probability. The following step carries on the iterative cycle of reliability analysis and deterministic optimization until the offsets converge. All the reliability analysis and optimization processes of SSRBO are based on the surrogate models. The time-consuming function evaluation only selects the points in the important region, which greatly reduces the number of sample points, and also ensures the accuracy of the reliability-based optimization.

4 Numerical examples

In this section, several mathematical and engineering examples are carried out to demonstrate the performance of SSRBO, and the results are compared with the existing RBO methods. The initial sample number is $n_0 = 2m + 1$, where m is the variable dimension. For each function call, the values of the objective and constraints are estimated simultaneously. In Step 1, the initial surrogate is constructed via the samples generated by LHS. GA is used in Step 2 to update the local surrogate models, while a strategy combining GA and SQP is used in Step 3. The examples are implemented in Matlab.

4.1 2D problem

This example is from literature (Cho and Lee 2011; Li et al. 2013; Yi et al. 2016), which is a typical two-variable problem with three reliability constraints. In this problem, the number of the variables is small, and the objective and constraint functions are relatively simple. The problem is described as

$$\min_{\mathbf{x}} \ \mathrm{E}(\mathrm{J}(\mathbf{x}+\boldsymbol{\varepsilon})) = x_1 + x_2 \qquad (16)$$

$$\text{s.t.}\begin{cases} \Pr\{\mathrm{g}_1(\mathbf{x}+\boldsymbol{\varepsilon}) \le 0\} \ge \phi(\beta_1) \\ \Pr\{\mathrm{g}_2(\mathbf{x}+\boldsymbol{\varepsilon}) \le 0\} \ge \phi(\beta_2) \\ \Pr\{\mathrm{g}_3(\mathbf{x}+\boldsymbol{\varepsilon}) \le 0\} \ge \phi(\beta_3) \end{cases}$$

where, $x_i \in [0, 10] (i = 1, 2)$ is the expected value of the random variable $X_i (\mathrm{E}(X_i) = x_i)$; the difference variable ε_i between X_i and x_i is dependent and normally distributed, $\varepsilon_i \sim \mathbb{N}(0, 0.\ 3^2)$; the prescribed target reliability index is $\beta_j = 3.0, \quad (j = 1, 2, 3)$. The detailed formulation of the constraint functions is expressed as

$$\begin{aligned} \mathrm{g}_1(\mathbf{x}) &= 1 - \frac{x_1^2 x_2}{20} \\ \mathrm{g}_2(\mathbf{x}) &= 1 - \frac{(x_1 + x_2 - 5)^2}{30} - \frac{(x_1 - x_2 - 12)^2}{120} \\ \mathrm{g}_3(\mathbf{x}) &= 1 - \frac{80}{x_1^2 + 8x_2 + 5} \end{aligned} \qquad (17)$$

As the result of the 2D problem, Fig. 4 shows the limit state functions, the failure region and the additional samples in the iteration process, Fig. 5 shows the iteration process of the reliability indexes corresponding to the constraint functions, and Fig. 6 shows the iteration process of the objective function value.

As shown in Fig. 4, the error between the approximate limit state functions (LSFs) and the true LSFs is larger at the beginning of the iteration, however, the approximate LSFs near the active constraints (g_1 and g_2) are becoming more and more accurate. At the same time, since the non-active constraint g_3 is far from the important region, the samples are sparse and the accuracy is poor. However, g_3 always meets the reliability constraints, so the poor accuracy does not affect the property of the non-active constraint, which reduces the samples and improves the efficiency of the algorithm. As shown in Fig. 5, the first three iterations in Step 3 fluctuates greatly, due to the large shift of the constraints. However, with the iteration going on, the constraint shift change smaller and smaller, and the reliability indexes converge gradually. As shown in Fig. 6, the iteration process of the SSRBO method is divided into three parts: Step 1 (samples 1 to 5), the values of the objective function are irregular, due to the Latin hypercube sampling; Step 2, the local surrogate models update step (samples 6~15), the overall trend of the objective function value is increasing, due to the minimum distance constraints, which forces the added points to be far away from the deterministic optimal point; Step 3, the optimization and Monte Carlo simulation step, since the constraints move inside the feasible domain, the objective function value becomes larger and larger until it converges. It is worth noting that both Fig. 5 and Fig. 6 show the iteration process of the optimization and Monte Carlo simulation step, however, the number of the objective function is less than that of the reliability indexes iteration, because some too close samples are not added to the sample set.

The detailed iteration process of SSRBO for the 2D problem is listed in Table 2 and the compared results of various RBO methods are shown in Table 3, which includes the objective function value at the optimal point, the reliability indexes of the constraints and the number of the function calls. $\beta_j^{\mathrm{MCS}} (j = 1, 2, 3)$ is the reliability index of the jth constraint with the Monte Carlo simulations for 1×10^6 times, which is used to validate the accuracy of the reliability constraints. As shown in Table 3, the objective function values of different methods are close. The objective function of SSRBO is a little larger than the existing methods, however, the reliability

Fig. 5 Iteration of the reliability indexes for the 2D problem

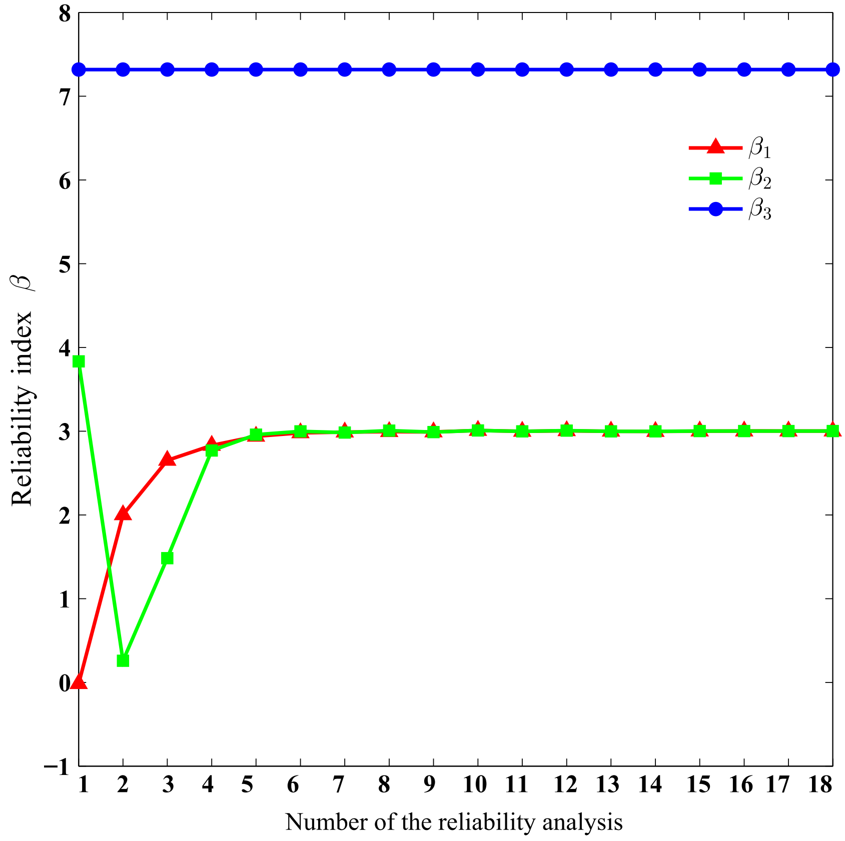


Fig. 6 Iteration of the objective function for the 2D problem

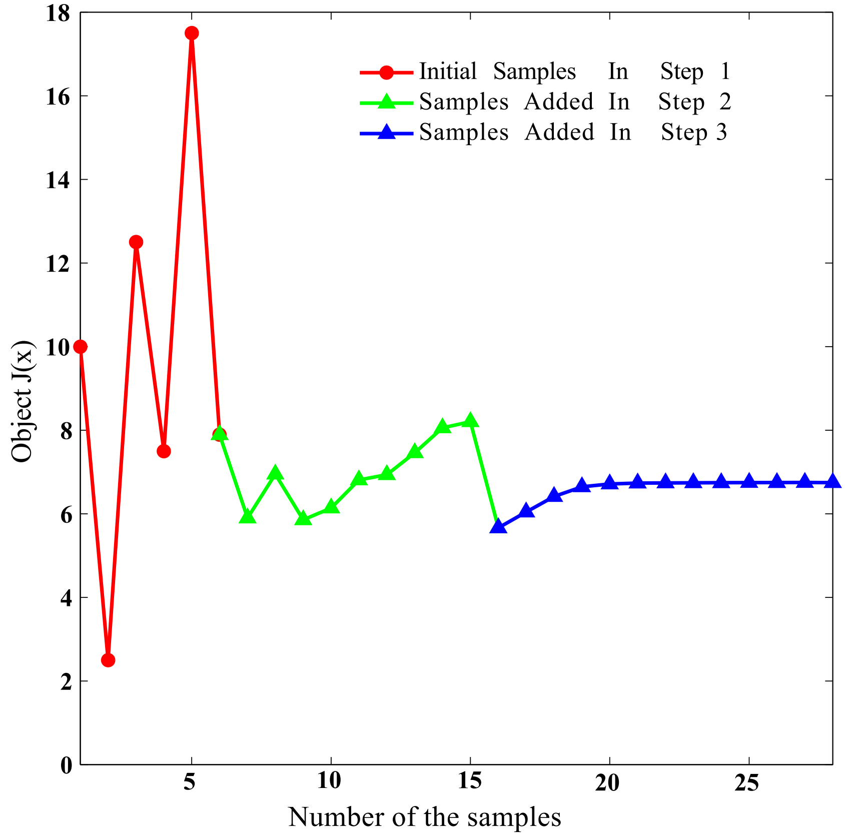


Table 2 The added samples for the 2D problem

Iteration	x_1	x_2	$J(\boldsymbol{x})$	$g_1(\boldsymbol{x})$	$g_2(\boldsymbol{x})$	$g_3(\boldsymbol{x})$
0	5.0000	5.0000	10.0000	−5.2500	−1.0333	−0.1429
	2.5000	0.0000	2.5000	1.0000	0.0396	−6.1111
	10.0000	2.5000	12.5000	−11.5000	−1.0438	0.3600
	0.0000	7.5000	7.5000	1.0000	−2.3771	−0.2308
	7.5000	10.0000	17.5000	−27.1250	−5.9604	0.4336
1	3.3381	4.5529	7.8910	−1.5367	−0.7338	−0.5219
2	3.4550	2.4419	5.8969	−0.4575	−0.0327	−1.1935
3	4.3391	2.6103	6.9494	−1.4573	−0.0058	−0.7893
4	2.7245	3.1326	5.8571	−0.1627	−0.3075	−1.1343
5	2.2415	3.8961	6.1375	0.0213	−0.5969	−0.9421
6	2.0351	4.7721	6.8072	0.0118	−0.9187	−0.6907
7	3.6045	3.3297	6.9342	−1.1631	−0.2704	−0.7925
8	1.8152	5.6449	7.4601	0.0700	−1.2899	−0.4966
9	2.7852	5.2694	8.0547	−1.0439	−1.0593	−0.4569
10	1.6703	6.5332	8.2036	0.0886	−1.7117	−0.3321
11	2.5298	3.1326	5.6624	−0.0024	−0.3382	−1.1941
12	3.5996	2.4419	6.0415	−0.5820	−0.0158	−1.1338
13	3.5746	2.8362	6.4107	−0.8120	−0.1232	−0.9769
14	3.4399	3.2033	6.6432	−0.8952	−0.2432	−0.8841
15	3.4459	3.2673	6.7133	−0.9399	−0.2624	−0.8599
16	3.4529	3.2803	6.7332	−0.9555	−0.2659	−0.8534
17	3.4565	3.2799	6.7364	−0.9593	−0.2654	−0.8524
18	3.4561	3.2861	6.7422	−0.9625	−0.2674	−0.8504
19	3.4587	3.2841	6.7428	−0.9643	−0.2666	−0.8503
20	3.4597	3.2900	6.7497	−0.9690	−0.2683	−0.8480
21	3.4586	3.2865	6.7451	−0.9656	−0.2673	−0.8496
22	3.4590	3.2878	6.7469	−0.9669	−0.2677	−0.8490
23	3.4584	3.2854	6.7438	−0.9648	−0.2670	−0.8500

constraints of the existing methods are not satisfied strictly. In this example, 5 initial samples and 23 additional samples are used in SSRBO, with a total number of 28 function calls, which is 1.96%~58.3% of the existing RBO methods.

Compared with the existing RBO methods (RIA, PMA, SORA and ASORA (Yi et al. 2016); SLSV (Yang and Gu 2004)), the SSRBO method does not carry out the optimization with the computationally expensive model directly, without nesting the optimizations for searching the MPPs, which reduces the function calls of the two processes dramatically. Moreover, SSRBO uses multiple samples in the neighborhood of the important region, which has a higher reliability accuracy than the single point approximation of the Taylor series expansion at the MPP. Most of the computation cost of SSRBO is the function calls of the initial and additional samples. Since the computation cost of the surrogate model itself is much lower than the time of a single function call, the time of each sub-optimization process is negligible.

4.2 Hock and Schittkowski problem

Hock and Schittkowski problem is a typical RBO problem with 10 random variables and 8 reliability constraints from literature (Lee and Lee 2005; Yi et al. 2016). In this problem, the number of the variables and constraint functions increases, and the objective function is more complex. The formulation is given by

$$\begin{aligned}\min_{\mathbf{x}} \quad & \mathrm{J}(\mathbf{x}) = x_1^2 + x_2^2 + x_1 + x_2 - 14x_1 - 16x_2 + (x_3-10)^2 + 4(x_4-5)^2 + (x_5-3)^2 + \\ & \qquad 2(x_6-1)^2 + 5x_7^2 + 7(x_8-11)^2 + 2(x_9-10)^2 + (x_{10}-7)^2 + 45 \\ \text{s.t.} & \begin{cases} \mathrm{P}\left\{\mathrm{g}_j(\mathbf{x}+\boldsymbol{\varepsilon}) \le 0\right\} \ge \phi\left(\beta_j\right), \quad j = 1, 2, \cdots, 8 \\ 0 \le x_i \le 10, i = 1, 2, \cdots, 10 \end{cases}\end{aligned} \tag{18}$$

where, the random variables $\varepsilon_i \sim \mathbb{N}(0, 0.02^2)$, $(i = 1, 2)$ are independent and normally distributed. The prescribed target reliability indexes $\beta_j = 3.0$ $(j = 1, 2, \cdots, 8)$. More details of the constraint functions are described by

Table 3 Comparison of different methods for the 2D problem

Method	Objective	Variables	β_1^{MCS}	β_2^{MCS}	β_3^{MCS}	Number of function calls
AMA	7.077	(3.633,3444)	3.775	3.377	+∞	694
RIA	6.726	(3.439,3.287)	2.970	3.050	+∞	882
PMA	6.728	(3.439,3.289)	2.970	3.060	+∞	1431
SORA	6.726	(3.439,3.287)	2.970	3.050	+∞	210
SLSV[a]	6.731	(3.434,3.297)	2.970	3.090	+∞	484
ASORA[b]	6.725	(3.440,3.285)	2.970	3.050	+∞	48
SSRBO	6.743	(3.458,3.285)	3.009	3.012	+∞	28

[a] Single loop single variable

[b] Approximate sequential optimization and reliability assessment

$$\begin{aligned}
&\mathrm{g}_1(\mathbf{x}) = 1-\frac{4x_1+5x_2-3x_7+9x_8}{105}\\
&\mathrm{g}_2(\mathbf{x}) = 10x_1-8x_2-17x_7+2x_8\\
&\mathrm{g}_3(\mathbf{x}) = 1-\frac{-8x_1+2x_2+5x_9-2x_{10}}{12}\\
&\mathrm{g}_4(\mathbf{x}) = 1-\frac{3(x_1-2)^2+4(x_2-3)^2+2x_3^2-7x_4}{120}\\
&\mathrm{g}_5(\mathbf{x}) = 1-\frac{5x_1^2+8x_2+(x_3-6)^2-2x_4}{40}\\
&\mathrm{g}_6(\mathbf{x}) = 1-\frac{0.5(x_1-8)^2+2(x_2-4)^2+3x_5^2-x_6}{30}\\
&\mathrm{g}_7(\mathbf{x}) = x_1^2+2(x_2-2)^2-2x_1x_2+14x_5-6x_6\\
&\mathrm{g}_8(\mathbf{x}) = -3x_1+6x_2+12(x_9-8)^2-7x_{10}
\end{aligned} \tag{19}$$

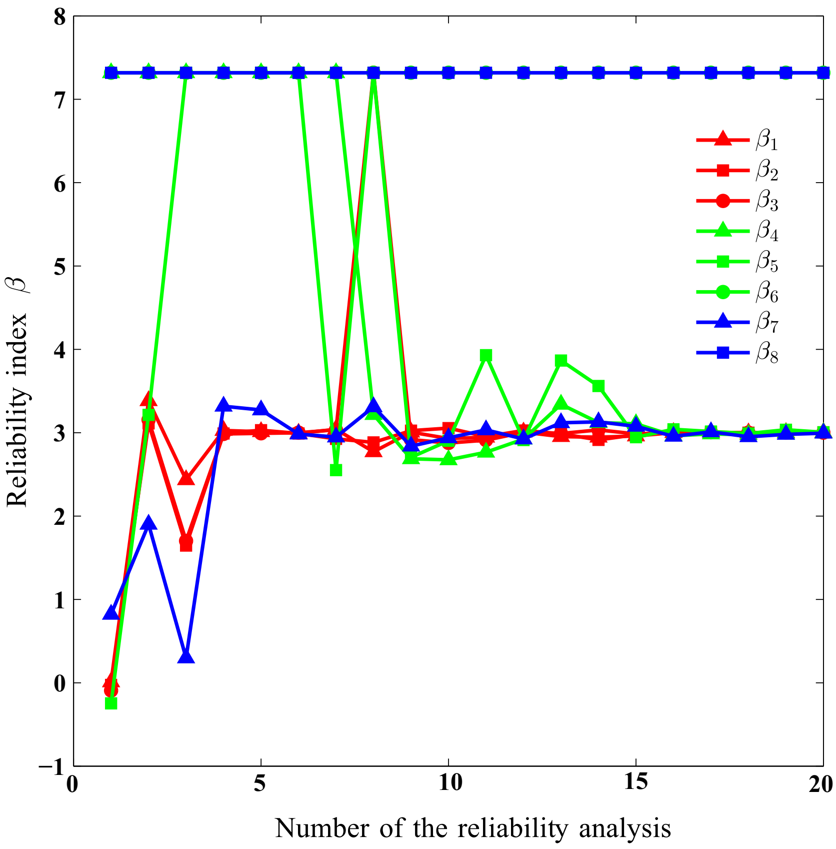
Figure 7 and Fig. 8 show the iterative process of the reliability indexes of the constraint functions and the value of the objective function, respectively. As shown in Fig. 7, the eight reliability index constraints satisfy the convergence condition after 20 iterations. At the beginning of the iteration, due to the large offsets, the reliability indexes changes dramatically. However, with the iteration going on, the offsets decrease gradually, leading to the reliability indexes converge gradually. The active constraints converge to the prescribed target value of 3.0, while the non-active constraints remain a high reliability index about 7.34. It can be seen from Fig. 8 that the initial sample is 21, the number of samples in Step 2 is 15, and the number of samples in Step 3 is 25. In Step 1, the initial sample has no obvious trend due to the use of Latin hypercube sampling. In Step 2, due to the use of the minimum distance as a constraint in the optimization of adding points, the objective function value changes from small to large. In Step 3, the new samples are obtained by the deterministic optimization with the shifted boundaries. Since the algorithm adopts the SQP, the function value converges relatively smoothly.

Table 4 and Table 5 show the results of different RBO methods (Lee and Lee 2005) with the number of function calls, where $\beta_j^{\mathrm{MCS}}(j=1,2,\cdots,8)$ represents the 1×10^6 times Monte Carlo simulations of the reliability indexes. As shown in Table 4, the number of the function calls required for SSRBO is about 0.0036%~0.16% of the alternative methods, reducing the computational cost remarkably. Meanwhile, as shown in Table 5, the Monte Carlo simulation reliability indexes of SSRBO are close to 3.0, meeting closely the reliability constraints.

4.3 The speed reducer problem

The speed reducer problem (Cho and Lee 2011; Yi et al. 2016) shown in Fig. 9 is an engineering application, which deals with the reliability-based optimization during the design of a gearbox. The speed reducer consists of a gear and pinion and

Fig. 7 Iteration of the reliability indexes for Hock and Schittkowski problem



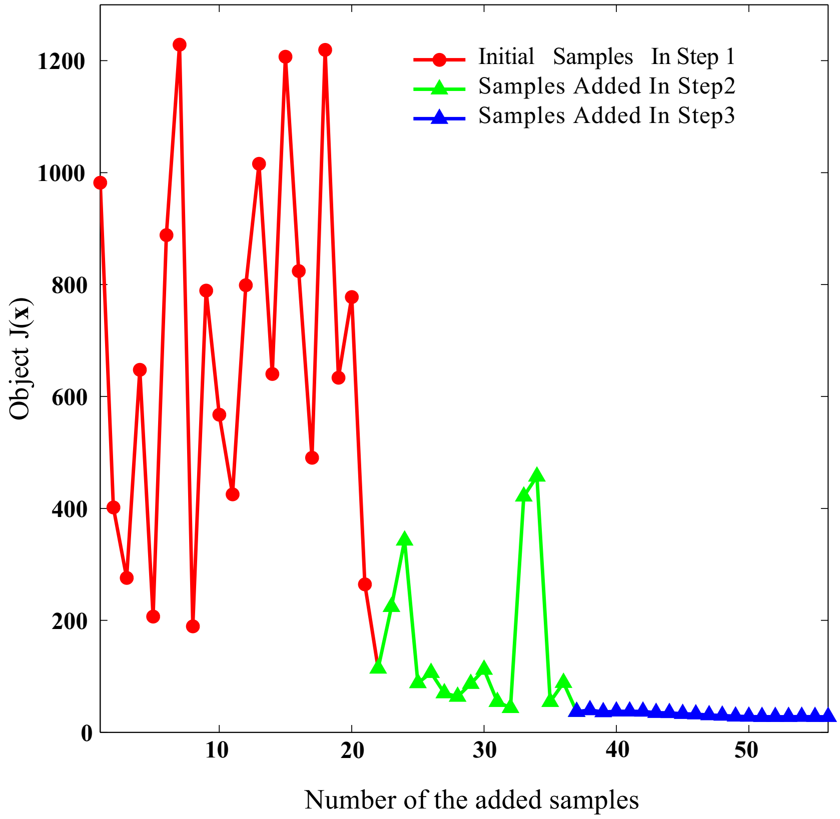


Fig. 8 Iteration of the objective function for Hock and Schittkowski problem

respective shafts. There are 7 design variables and 11 reliability constraints. The random design variables are gear width (x_1), gear module (x_2), the number of pinion teeth (x_3), distance between bearings (x_4, x_5) and shaft diameters (x_6, x_7). The objective is to minimize the weight of the speed reducer, subjected to the physical properties such as the bending stress, contact stress, longitudinal displacement, shaft stress and geometry constraints.

Table 4 Results of different methods for Hock and Schittkowski problem

Method	Objective	Variables	Number of function calls
AMA	27.75	(2.13,2.33,8.71,5.10,0.92,1.44,1.39,9.81,8.16,8.47)	1288
RIA	27.62	(2.13,2.35,8.72,5.14,1.00,1.61,1.38,9.79,8.16,8.47)	93629
PMA	27.80	(2.13,2.34,8.72,5.12,0.99,1.60,1.38,9.80,8.14,8.46)	157650
RIA+EF[a]	34.31	(2.16,2.37,8.00,5.08,1.03,1.84,1.49,9.69,8.00,8.29)	65545
PMA+EF[a]	32.64	(2.19,2.35,7.91,5.12,1.02,1.83,1.48,9.81,8.07,8.56)	34694
SSRBO	27.75	(2.13,2.34,8.71,5.12,0.93,1.46,1.38,9.80,8.17,8.52)	56

[a] Envelope function (Lee and Lee 2005)

Table 5 Reliability indexes of different methods for Hock and Schittkowski problem

Method	β_1^{MCS}	β_2^{MCS}	β_3^{MCS}	β_4^{MCS}	β_5^{MCS}	β_6^{MCS}	β_7^{MCS}	β_8^{MCS}
AMA	2.966	3.026	2.958	3.041	2.970	$+\infty$	2.950	$+\infty$
RIA	3.328	3.218	2.439	2.988	2.949	$+\infty$	2.871	$+\infty$
PMA	3.154	2.989	2.955	2.727	3.026	$+\infty$	3.037	$+\infty$
RIA+EF	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
PMA + EF	2.796	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
SSRBO	3.005	2.982	3.005	3.063	2.991	$+\infty$	2.996	$+\infty$

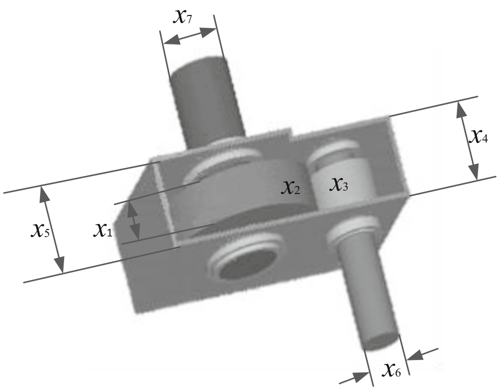


Fig. 9 Structure of the speed reducer

In this problem, the number of the variables and constraint functions is large, and the objective and constraint functions are more complex. The problem is formulated as:

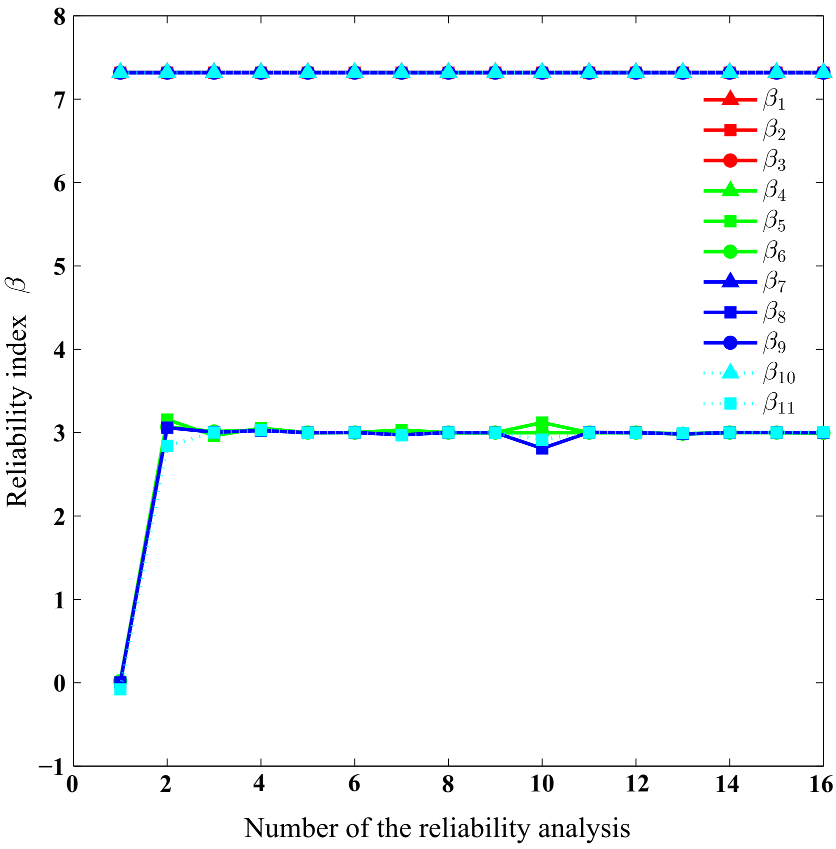
$$\begin{aligned}\min \mathrm{J}(\mathbf{x}) = {} & 0.7854x_1x_2^2\left(3.3333x_3^2+14.9334x_3{-}43.0934\right){-}1.5079x_1\left(x_6^2+x_7^2\right)\\ & +7.477\left(x_6^2+x_7^3\right)+0.7854\left(x_4x_6^2+x_5x_7^2\right)\\ \text{s.t.} & \begin{cases}\mathrm{P}\{\mathrm{g}_i(\mathbf{x}+\boldsymbol{\varepsilon})\le 0\}\ge\phi(\beta_i), \quad i=1,2,\cdots,11\\ \mathbf{x}_\mathrm{L}\le\mathbf{x}\le\mathbf{x}_\mathrm{U}\end{cases}\end{aligned} \tag{20}$$

where $\mathbf{x}_\mathrm{L}=[2.6,0.7,17,7.3,7.3,2.9,5.0]^\mathrm{T}$, $\mathbf{x}_\mathrm{U}=[3.6,0.8,28,8.3,8.3,3.9,5.5]^\mathrm{T}$ are the lower and upper bounds, respectively. $\boldsymbol{\varepsilon}$ is the vector random variables, and ε_i is normally distributed, $\varepsilon_i \sim N(0,0.02^2)$. $\beta_i = 3.0$ $(i=1,2,\cdots,11)$ is the ith reliability index. Moreover, the deterministic constraint functions are given by

$$\begin{aligned}
&\mathrm{g}_1(\mathbf{x})=\frac{27}{x_1x_2^2x_3}{-}1\le 0 && \mathrm{g}_2(\mathbf{x})=\frac{397.5}{x_1x_2^2x_3^2}{-}1\le 0\\
&\mathrm{g}_3(\mathbf{x})=\frac{1.93x_4^3}{x_2x_3x_6^4}{-}1\le 0 && \mathrm{g}_4(\mathbf{x})=\frac{1.93x_5^3}{x_2x_3x_7^4}{-}1\le 0\\
&\mathrm{g}_5(\mathbf{x})=\frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2+16.9\times 10^6}}{110x_6^3}{-}1\le 0 && \mathrm{g}_6(\mathbf{x})=\frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2+157.5\times 10^6}}{85x_7^3}{-}1\le 0\\
&\mathrm{g}_7(\mathbf{x})=x_2x_3{-}40\le 0 && \mathrm{g}_8(\mathbf{x})=5{-}\frac{x_1}{x_2}\le 0,\\
&\mathrm{g}_9(\mathbf{x})=\frac{x_1}{x_2}{-}12\le 0 && \mathrm{g}_{10}(\mathbf{x})=\frac{1.5x_6+1.9}{x_4}{-}1\le 0\\
&\mathrm{g}_{11}(\mathbf{x})=\frac{1.1x_7+1.9}{x_5}{-}1\le 0
\end{aligned} \tag{21}$$

As the results of the speed reducer problem, Fig. 10 shows the iterative process of the reliability indexes of the constraints. Figure 11 shows the iterative process of the objective function value of three steps. In the initial sample step, there are no obvious patterns of the change of the function value due to the use of optimum Latin hypercube sampling. In the step of the local surrogate model update, with the number of the samples increases,

Fig. 10 Iteration of the reliability indexes for the speed reducer problem



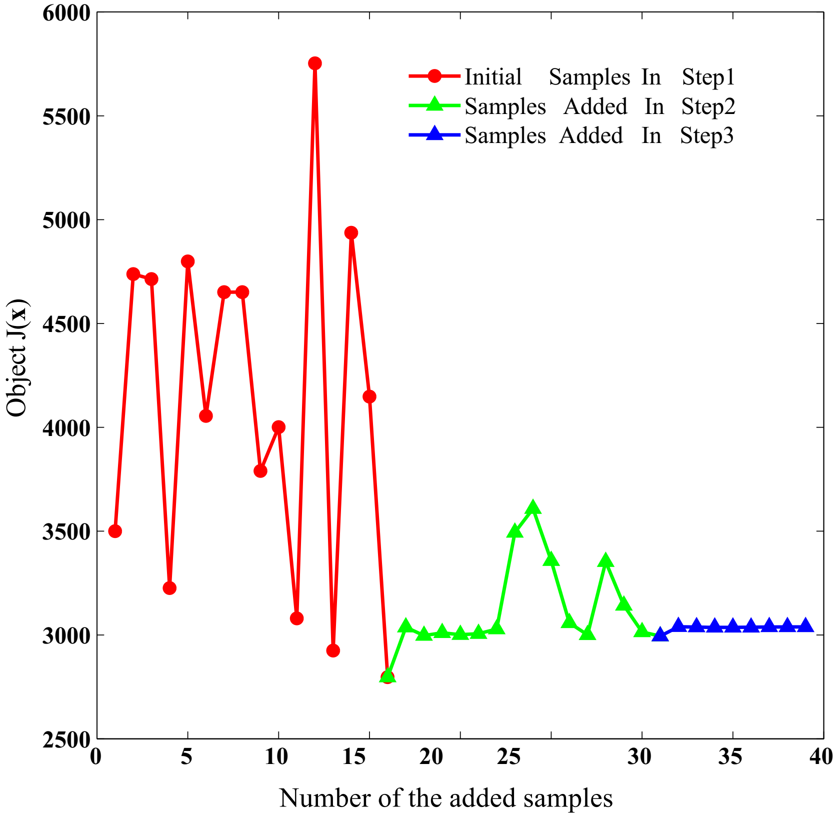


Fig. 11 Iteration of the objective function for the speed reducer problem

the function value is getting bigger and bigger. However, when the sample increases to a certain extent, the surrogate models change greatly, so the objective function value begins to reduce. In the optimization and Monte Carlo simulation step, the initial constraint boundary offset is zero, and then the constraint is shifted according to the approximated cumulative distribution function until the final convergence.

Table 6 and Table 7 show the results of different RBO methods (Cho and Lee 2011) for the speed reducer problem, where β_j^{MCS} $(j = 1, 2, \cdots, 11)$ represents the reliability indexes of 1×10^6 Monte Carlo simulations with the practical model at the corresponding reliability optimum points. As shown in Table 6, the objective function value of the SSRBO method is similar with the alternative methods. However, the number of the function evaluations is only $0.13\% \sim 6.22\%$ of the alternative methods, which indicates that the computational cost is greatly reduced. As shown in Table 7, the sixth reliability index of SSRBO is slightly less than 3.0 (the required reliability index constraint), due to the local inaccuracy of the surrogate model. The error is equivalent to that of the fifth reliability index of RIA. In addition, AMA fails to get a feasible solution, while the SORA and ASORA's 11th active constraint reliability index is far greater than 3, making the design more conservative and therefore the objective function is larger.

4.4 Design of crashworthiness of vehicle side impact

This example is a vehicle crashworthiness design (Youn and Choi 2004; Jiang et al. 2017), a more complex practical engineering problem with 11 random variables and 10 reliability constraints.

Table 6 Results of different methods for the speed reducer problem

Methods	Objective	Variables	Number of function calls
AMA	3126.94	(3.600,0.700,17.322,8.166,7.646,3.396,5.314)	280
RIA	3038.58	(3.577,0.700,17.000,7.300,7.754,3.365,5.302)	29096
PMA	3040.02	(3.578,0.700,17.000,7.300,7.764,3.366,5.302)	5852
SORA	3040.02	(3.578,0.700,17.000,7.300,7.764,3.366,5.302)	1023
SLSV[a]	3048.45	(3.589,0.700,17.000,7.300,7.783,3.369,5.307)	1771
ASORA[b]	3040.61	(3.580,0.700,17.000,7.300,7.764,3.366,5.302)	627
SSRBO	3038.53	(3.576,0.700,17.000,7.300,7.754,3.366,5.301)	39

[a] Single loop single variable

[b] Approximate sequential optimization and reliability assessment. (Cho and Lee 2011)

Table 7 Reliability indexes of different methods for the speed reducer problem

Methods	β_1^{MCS}	β_2^{MCS}	β_3^{MCS}	β_4^{MCS}	β_5^{MCS}	β_6^{MCS}	β_7^{MCS}	β_8^{MCS}	β_9^{MCS}	β_{10}^{MCS}	β_{11}^{MCS}
AMA	2.095	+∞	+∞	+∞	2.193	1.393	+∞	0.988	+∞	+∞	−3.333
RIA	+∞	+∞	+∞	+∞	2.960	3.079	+∞	3.006	+∞	+∞	2.935
PMA	+∞	+∞	+∞	+∞	3.163	3.062	+∞	3.072	+∞	+∞	4.265
SORA	+∞	+∞	+∞	+∞	3.152	3.064	+∞	3.066	+∞	+∞	4.159
SLSV	+∞	+∞	+∞	+∞	3.772	4.159	+∞	3.503	+∞	+∞	+∞
ASORA	+∞	+∞	+∞	+∞	3.174	3.077	+∞	3.134	+∞	+∞	4.376
SSRBO	+∞	+∞	+∞	+∞	3.129	2.946	+∞	3.010	+∞	+∞	3.085

The finite element model is shown in Fig. 12. The original nonlinear transient dynamic finite element simulation takes CPU time about 20 h for a single run. Therefore, the approximated model is adopted for the demonstration. In this work, the optimization task is to minimize the weight, subject to the reliability constraints, which is formulated as

$$\begin{aligned} \min \mathrm{J}(\mathbf{x}) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 \\ + 1.78x_5 + 2.73x_7 \\ \text{s.t.} \begin{cases} \mathrm{P}\{\mathrm{g}_i(\mathbf{x}+\boldsymbol{\varepsilon}) \le 0\} \ge \phi(\beta_i), & \beta_i = 3.0, i = 1, 2, \cdots, 11 \\ \mathbf{x}_{\mathrm{L}} \le \mathbf{x} \le \mathbf{x}_{\mathrm{U}} \end{cases} \end{aligned} \tag{22}$$

where, the random variables are normally distributed, and the distribution parameters are shown in Table 8. The target reliability index is $\beta_i = 3.0$, and more detailed expressions of the constraint functions are given by

$$\begin{aligned}
&\mathrm{g}_1(\mathbf{x}) = Load_{\text{Abdomen}}\text{-}1 = 1.16{-}0.3717x_2x_4{-}0.00931x_2x_{10}{-}0.484x_3x_9 + 0.01343x_6x_{10}\text{-}1 \\
&\mathrm{g}_2(\mathbf{x}) = Deflectionrib_{\text{upper}}\text{-}32 = 28.98 + 3.818x_3{-}4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9{-}7.7x_7x_8 + 0.32x_9x_{10} \\
&\mathrm{g}_3(\mathbf{x}) = Deflectionrib_{\text{middle}}\text{-}32 = 33.86 + 2.95x_3 + 0.1792x_{10}{-}5.057x_1x_2{-}11.0x_2x_8{-} \\
&\qquad 0.0215x_5x_{10}{-}9.98x_7x_8 + 22.0x_8x_9\text{-}32 \\
&\mathrm{g}_4(\mathbf{x}) = Deflectionrib_{\text{1ower}}\text{-}32 = 46.36{-}9.9x_2{-}12.9x_1x_8 + 0.1107x_3x_{10} \\
&\mathrm{g}_5(\mathbf{x}) = VC_{\text{upper}}\text{-}32 = 0.261{-}0.0159x_1x_2{-}0.188x_1x_8{-}0.019x_2x_7 + 0.0144x_3x_5 + \\
&\qquad 0.0008757x_5x_{10} + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11}\text{-}32 \\
&\mathrm{g}_6(\mathbf{x}) = VC_{\text{middle}}\text{-}32 = 0.214 + 0.00817x_5{-}0.131x_1x_8{-}0.0704x_1x_9 + 0.03099x_2x_6{-}0.018x_2x_7 + \\
&0.0208x_3x_8 + 0.121x_3x_9{-}0.00364x5x6 + 0.0007715x5x10{-}0.0005354x6x10 + 0.00121x_8x_{11}\text{-}32 \\
&\mathrm{g}_7(\mathbf{x}) = VC_{\text{lower}}\text{-}32 = 0.74{-}0.061x_2{-}0.163x_3x_8 + 0.001232x_3x_{10}{-}0.166x_7x_9 + 0.227x_2^2\text{-}32 \\
&\mathrm{g}_8(\mathbf{x}) = Force_{\text{public}}\text{-}4.01 = 4.72{-}0.5x4{-}0.19x_2x_3{-}0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}^2\text{-}4.01 \\
&\mathrm{g}_9(\mathbf{x}) = Velocity_{\text{pillar}}\text{-}9.9 = 10.58{-}0.674x_1x_2{-}1.95x_2x_8 + 0.02054x_3x_{10}{-}0.0198x_4x_{10} + 0.028x_6x_{10}\text{-}9.9 \\
&\mathrm{g}_{10}(\mathbf{x}) = Velocity_{\text{door}}\text{-}15.69 = 16.45{-}0.489x_3x_7{-}0.843x_5x_6 + 0.0432x_9x_{10}{-}0.0556x_9x_{11}{-}0.000786x_2\text{-}15.69
\end{aligned} \tag{23}$$

As shown in Table 8, the random variables X_1 to X_7 are structure variables, X_8 and X_9 are material property parameters, X_{10} and X_{11} express barrier height and barrier hitting position respectively. As X_8 to X_{11} are parameters, the upper bound and the lower bound are equal.

The RBO results are listed in Table 9 and Table 10. As shown in Table 9, SSRBO obtains an objective value 28.3229 with 66 function calls, the computation cost of which is about 0.345 to 15.9% of the compared methods (Jiang et al. 2017). The reliability indexes of the reliability constraint functions in Table 10 are verified by MCS with 1 × 10^6 samples. As the target reliability index is 3.0, the eighth and tenth reliability constraints are violated in SSRBO. Thus, the total violation of SSRBO is 0.522, while the violation of SLA is 2.098, and the violations of the other methods are 0.567~0.684. Therefore, SSRBO performs higher efficiency and comparable accuracy in the vehicle side impact problem.

4.5 Design of aircraft wing structure

This example is an RBO problem for a high-speed aircraft wing structure. The design variables are the positions of

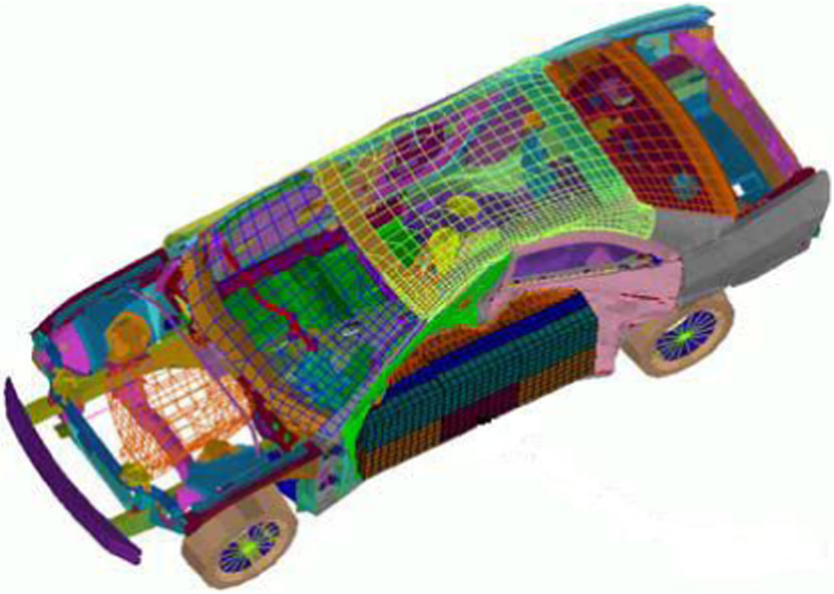


Fig. 12 Vehicle impact model

Table 8 Distribution of random variables for the vehicle side impact problem

Random variables	Mean	$\mathbf{x}_\mathrm{L}$	$\mathbf{x}_\mathrm{U}$	Standard deviation
B-Pillar inner X_1	x_1	0.5	1.5	0.03
B-Pillar reinforcement X_2	x_2	0.45	1.35	0.03
Floor side inner X_3	x_3	0.5	1.5	0.03
Cross member X_4	x_4	0.5	1.5	0.03
Door beam X_5	x_5	0.875	2.625	0.05
Door belt line reinforcement X_6	x_6	0.4	1.2	0.03
Roof rail X_7	x_7	0.4	1.2	0.03
Material of B-Pillar inner X_8	x_8	0.345	0.345	0.006
Material of floor side inner X_9	x_9	0.345	0.345	0.006
Barrier height X_{10}	x_{10}	0	0	10
Barrier height X_1position X_{11}	x_{11}	0	0	10

Table 9 Results of different methods for the vehicle side impact problem

Methods	Objective	Variables	Number of function calls
AMA	30.6840	(0.8722,1.3498,0.8169,1.5000,0.8850,0.9744,0.7821,0.345,0.345,0,0)	1596
PMA	29.5578	(0.7872,1.3500,0.6887,1.5000,1.0706,1.2000,0.7284,0.345,0.345,0,0)	19128
SORA	29.5578	(0.7872,1.3500,0.6887,1.5000,1.0706, 1.2000,0.7284,0.345,0.345,0,0)	2148
SLA[a]	28.9740	(0.9220,1.3500,0.6887,1.5000,0.8750, 1.2000,0.4000,0.345,0.345,0,0)	415
AHA[b]	29.5578	(0.7872,1.3500,0.6887,1.5000,1.0706, 1.2000,0.7284,0.345,0.345,0,0)	2184
AH_SLM[c]	29.5578	(0.7872,1.3500,0.6887,1.5000,1.0705, 1.2000,0.7284,0.345,0.345,0,0)	759
SSRBO	28.3229	(0.7701,1.5000,0.5078,1.5000,0.6307,1.4376,0.6895,0.345,0.345,0,0)	66

[a] Single-loop approach

[b] Adaptive hybrid approach

[c] Adaptive hybrid single-loop method. (Jiang et al. 2017)

the beams and ribs, and 26 design variables are involved (See Fig. 13). The objective function is to minimize the weight of the wing structure. The design constraint is that the maximum stresses of each component do not exceed the allowable stress of the material. There are 29 stresses involved in the problem. As shown in Fig. 14, the structural model is a finite element analysis model without explicit expression. The model contains 18504 unstructured meshes, and the single analysis (with the solver Nastran) takes an average of 64.8 s (Environment: Windows, 64 bits, 2 GHz).

$$\min \mathrm{J}(\mathbf{x}) = \mathrm{Weight}(\mathbf{x})$$
$$\text{s.t.} \begin{cases} \mathrm{P}\{\mathrm{g}_i(\mathbf{x}+\boldsymbol{\varepsilon})\leq 0\}\geq\phi(\beta_i), \quad \beta_i = 3.0, i = 1,2,\cdots,29 \\ \mathbf{x}_\mathrm{L}\leq\mathbf{x}\leq\mathbf{x}_\mathrm{U}, \varepsilon_i\sim\mathbb{N}\left(0, \left(0.02\left(x_{\mathrm{U},i}-x_{\mathrm{L},i}\right)\right)^2\right) \end{cases} \tag{24}$$

The SSRBO algorithm is used to solve this problem. The number of initial samples is 53. The stress distribution for the optimized wing structure is shown in Fig. 15. The iteration processes of reliability indexes and objective function are shown in Fig. 16 and Fig. 17 respectively. The calculation results are shown in Table 11.

Table 10 Reliability indexes of different methods for the vehicle side impact problem

Methods	β_1^{MCS}	β_2^{MCS}	β_3^{MCS}	β_4^{MCS}	β_5^{MCS}	β_6^{MCS}	β_7^{MCS}	β_8^{MCS}	β_9^{MCS}	$\beta_{10}^{\mathrm{MCS}}$
AMA	+∞	2.991	4.107	3.527	+∞	+∞	+∞	2.325	+∞	+∞
PMA	+∞	3.003	4.344	3.005	3.003	+∞	+∞	2.433	+∞	3.206
SORA	+∞	3.003	4.244	3.005	3.358	+∞	+∞	2.433	+∞	3.206
SLA	+∞	2.998	4.173	3.707	3.358	+∞	+∞	2.431	+∞	1.473
AHA	+∞	3.003	4.344	3.005	4.056	+∞	+∞	2.433	+∞	3.206
AH_SLM	+∞	3.003	4.344	3.005	3.358	+∞	+∞	2.433	+∞	3.206
SSRBO	+∞	+∞	+∞	3.554	+∞	+∞	+∞	2.528	4.753	2.950

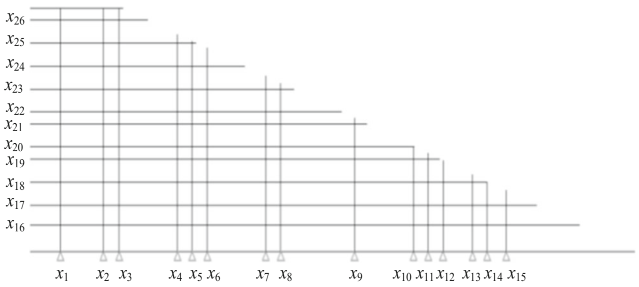


Fig. 13 Positions of the beams and ribs

As shown in Fig. 16, since β_1 trends to 3.0, $g_1(\mathbf{x})$ is the active constraint. The constraint boundary begins to move in a large range, and the oscillation is performed for a longer time to satisfy the convergence condition. Figure 17 shows the improvement of the objective function with the increase of the samples. It can be seen that the value of the objective function undergoes a large degree of change during the step 2 to step 3, and the value of the objective function also increases as the reliability increases. In step 3, due to the strict convergence conditions, the convergence speed is slow. Table 11 shows the calculation results of different methods and the reliability indexes come from 10,000 times Monte Carlo simulations with the original model. Since the RIA, PMA, and SORA methods have taken much time and there are no obvious signs of convergence, the results when the number of calculations reaches 50,000 are taken as the final results. In order to compare the effect of different surrogate models on the performance of SSRBO, SVR and ANN are used instead of RBF. It can be seen that different surrogate models have similar results in the problem. As different surrogate models have different construction and forecasting time, it will eventually have a certain impact on the time cost of the entire optimization problem, but the computing time is generally at an order of magnitude. It can be seen from Table 11 that the SSRBO method has comparable results with the existing methods, but the calculation time is only 0.28 to 1.21% of the comparison methods. This shows that the proposed method improves the efficiency of RBO in the practical engineering problem.

5 Conclusions

In this paper, a reliability-based design optimization method with sequential surrogate model (SSRBO) is proposed. SSRBO involves three key steps. First, the global surrogate models of the objective function and constraint functions are constructed with the initial samples. Second, the surrogate models are updated with the optimization criterion of adding points in the important region with high failure probability. Third, the cumulative distribution function of the constraints at the current point is obtained by the kernel method with the Monte Carlo simulation; therefore, the offsets of the constraint for the next iteration are determined.

Fig. 14 Meshes of the wing structure

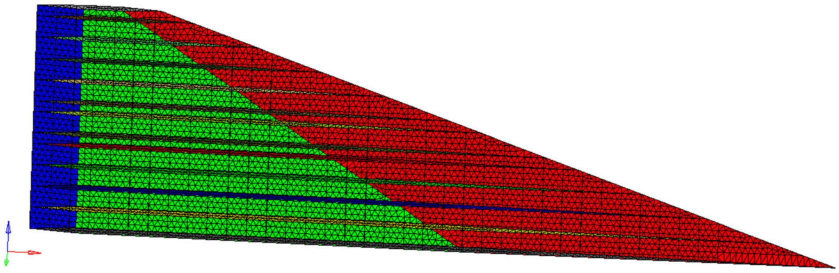


Fig. 15 The stress distribution for the optimized wing structure with SSRBO

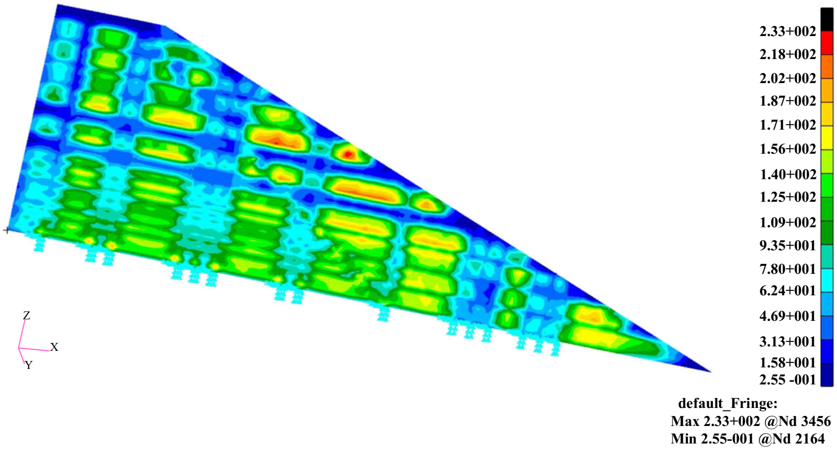


Fig. 16 Iteration of the reliability indexes for the design of aircraft wing structure

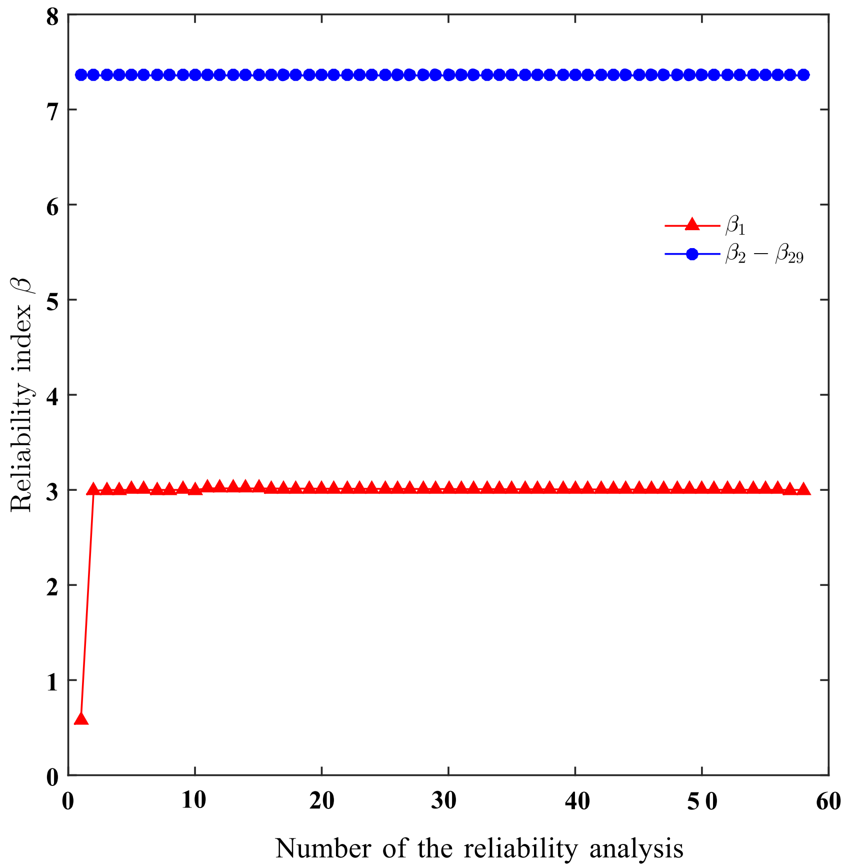


Fig. 17 Iteration of the objective function for the design of aircraft wing structure

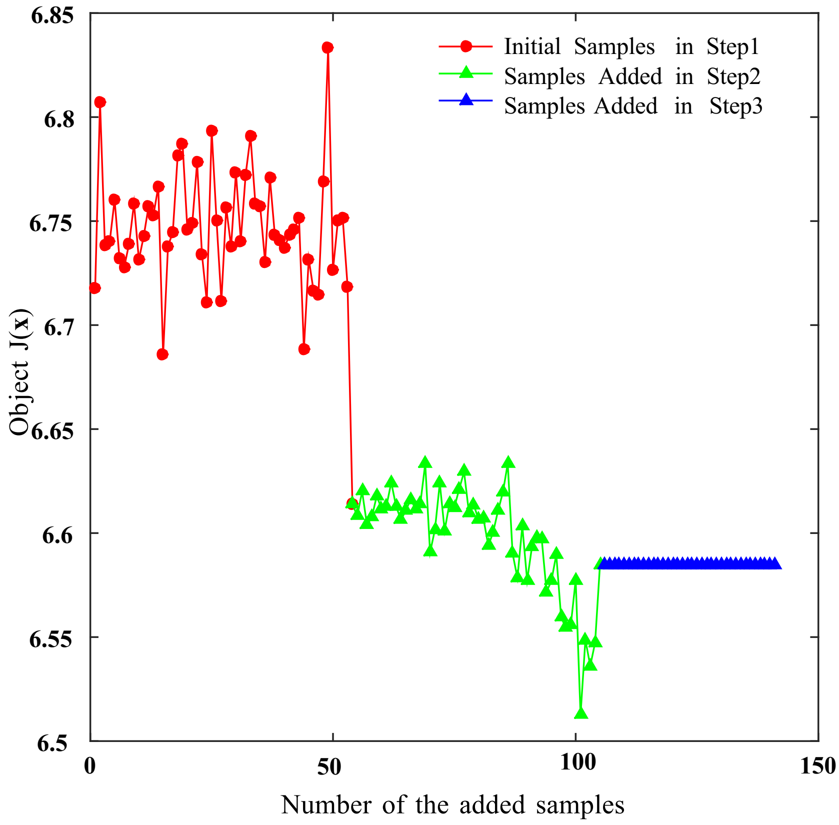


Table 11 Results of RBO for the design of aircraft wing structure

Methods	Objective (ton)	β_1^{MCS}	β_2^{MCS}~β_{29}^{MCS}	Number of function calls	Total time (hour)
AMA	6.604	+∞	+∞	17523	315.477
RIA	6.596	2.775	+∞	50000	910.218
PMA	6.588	2.793	+∞	50000	906.387
SORA	6.563	2.539	+∞	50000	920.673
SSRBO-RBF	6.584	2.766	+∞	141	2.603
SSRBO-SVR	6.635	2.553	+∞	157	2.917
SSRBO-ANN	6.616	2.627	+∞	189	3.814

The numerical and engineering examples with different number of variables and constraints show that SSRBO effectively reduces the number of the function calls without losing accuracy of the reliabilities compared with the existing methods. SSRBO updates the local surrogate models of objective and constraint functions in the vicinity of the deterministic optimization point. Moreover, it carries out Monte Carlo simulation with surrogate models to obtain the offsets, which avoids the functions calls of reliability analysis and retains the accuracy. Since this paper mainly solves the RBO problems of time-consuming models, the number of function calls can be used to approximate the overall time of SSRBO. For the non-linear active constraints, in order to satisfy a given accuracy, more sample points are required; for the weak non-linear constraints, with only a small number of samples the results converge quickly.

The number of additional samples has great influence on the convergence performance. How to adopt the additional sample in step 2 of SSRBO according to the complexity of the model and to increase the stability of the algorithm is a major direction for future research. Moreover, SSRBO is highly depended on the surrogate model; therefore, how to improve the construction accuracy and efficiency of the surrogate model for a multiple-input-multiple-output system is also an interesting research topic.

Acknowledgments The authors also thank Dr. Xueyu Li for the helpful work to improve the study.

Funding information The research is supported by the Fundamental Research Funds for the Central Universities (No. G2016KY0302) and the National Natural Science Foundation of China (No. 11572134).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

Bowman AW, Azzalini A (1997) Applied smoothing techniques for data analysis: the kernel approach with S-PLUS illustrations. Oxford University Press, New York

Breitkopf P, Coelho RF (2010) Multidisciplinary design optimization in computational mechanics. Wiley-ISTE, Hoboken

Cawlfield JD (2000) Application of first-order (FORM) and second-order (SORM) reliability methods: analysis and interpretation of sensitivity measures related to groundwater pressure decreases and resulting ground subsidence. In: Sensitivity analysis. Wiley, Chichester, pp 317–327

Cho TM, Lee BC (2011) Reliability-based design optimization using convex linearization and sequential optimization and reliability assessment method. Struct Saf 24:42–50. https://doi.org/10.1007/s12206-009-1143-4

Cizelj L, Mavko B, Riesch-Oppermann H (1994) Application of first and second order reliability methods in the safety assessment of cracked steam generator tubing. Nucl Eng Des 147:359–368. https://doi.org/10.1016/0029-5493(94)90218-6

Doan BQ, Liu G, Xu C, Chau MQ (2018) An efficient approach for reliability-based design optimization combined sequential optimization with approximate models. Int J Comput Methods 15:1–15

Du X (2008) Unified uncertainty analysis by the first order reliability method. J Mech Des 130:1404. https://doi.org/10.1115/1.2943295

Du X, Chen W (2004) Sequential optimization and reliability assessment for probabilistic design. J Mech Des 126:225–233

Enevoldsen I, Sørensen JD (1994) Reliability-based optimization in structural engineering. Struct Saf 15:169–196

Forrester DAIJ, Sóbester DA, Keane AJ (2008) Engineering design via surrogate modelling: a practical guide. John Wiley & Sons Ltd., London

Grandhi RV, Wang L (1998) Reliability-based structural optimization using improved two-point adaptive nonlinear approximations. Finite Elem Anal Des 29:35–48

Haldar A, Mahadevan S (1995) First-order and second-order reliability methods. Springer US

Hastie T, Robert T, Friedman J (2008) The elements of statistical learning: data mining, inference and prediction, 2nd edn. Springer, London

Hurtado JE, Alvarez DA (2001) Neural-network-based reliability analysis: a comparative study. Comput Methods Appl Mech Eng 191: 113–132

Jiang C, Qiu H, Gao L et al (2017) An adaptive hybrid single-loop method for reliability-based design optimization using iterative control strategy. Struct Multidiscip Optim 56:1–16

Koch PN, Yang RJ, Gu L (2004) Design for six sigma through robust optimization. Struct Multidiscip Optim 26:235–248

Laurenceau J, Sagaut P (2008) Building efficient response surfaces of aerodynamic functions with kriging and cokriging. AIAA J 46: 498–507

Lee JJ, Lee BC (2005) Efficient evaluation of probabilistic constraints using an envelope function. Eng Optim 37:185–200

Li F, Wu T, Badiru A et al (2013) A single-loop deterministic method for reliability-based design optimization. Eng Optim 45:435–458

Li X, Qiu H, Chen Z et al (2016) A local kriging approximation method using MPP for reliability-based design optimization. Comput Struct 162:102–115

Liang J, Mourelatos ZP, Tu J (2008) A single-loop method for reliability-based design optimisation. Int J Prod Dev 5:76–92 (17)

Meng Z, Zhou H, Li G, Hu H (2017) A hybrid sequential approximate programming method for second-order reliability-based design optimization approach. Acta Mech 228:1–14

Qu X, Haftka RT (2004) Reliability-based design optimization using probabilistic sufficiency factor. Struct Multidiscip Optim 27:314–325

Regis RG, Shoemaker CA (2005) Constrained global optimization of expensive black box functions using radial basis functions. J Glob Optim 31:153–171

Ronch AD, Ghoreyshi M, Badcock KJ (2011) On the generation of flight dynamics aerodynamic tables by computational fluid dynamics. Prog Aerosp Sci 47:597–620

Ronch AD, Panzeri M, Bari MAA et al (2017) Adaptive design of experiments for efficient and accurate estimation of aerodynamic loads. In: 6th Symposium on Collaboration in Aircraft Design (SCAD)

Strömberg N (2017) Reliability-based design optimization using SORM and SQP. Struct Multidiscip Optim 56:631–645

Tsompanakis Y, Lagaros ND, Papadrakakis M (2010) Structural design optimization considering uncertainties. Taylor and Francis, London

Tu J, Choi KK, Park YH (1999) A new study on reliability-based design optimization. J Mech Des 121:557–564

Valdebenito MA, Schuoller GI (2010) A survey on approaches for reliability-based optimization. Struct Multidiscip Optim 42:645–663. https://doi.org/10.1007/s00158-010-0518-6

Verderaime V (1994) Illustrated structural application of universal first-order reliability method. NASA STI/Recon Tech Rep N 95:11870

Yang RJ, Gu L (2004) Experience with approximate reliability-based optimization methods. Struct Multidiscip Optim 26:152–159

Yi P, Zhu Z, Gong J (2016) An approximate sequential optimization and reliability assessment method for reliability-based design optimization. Struct Multidiscip Optim 54:1367–1378

Youn BD, Choi KK (2004) A new response surface methodology for reliability-based design optimization. Comput Struct 82:241–256

Youn BD, Choi KK, Yang RJ, Gu L (2004) Reliability-based design optimization for crashworthiness of vehicle side impact. Struct Multidiscip Optim 26:272–283

Zhou M, Luo Z, Yi P, Cheng G (2018) A two-phase approach based on sequential approximation for reliability-based design optimization. Struct Multidiscip Optim 1–20