Engineering with Computers (2021) 37:1295–1314
https://doi.org/10.1007/s00366-019-00884-0

ORIGINAL ARTICLE

Reliability-based design optimization using adaptive surrogate model and importance sampling-based modified SORA method

Kunling Song[1] · Yugang Zhang[1] · Xinchen Zhuang[1] · Xinshui Yu[1] · Bifeng Song[1]

Received: 29 August 2019 / Accepted: 30 October 2019 / Published online: 13 November 2019

Abstract
Reliability-based design optimization (RBDO) has been an important research field with the increasing demand for product reliability in practical applications. This paper presents a new RBDO method combining adaptive surrogate model and Importance Sampling-based Modified Sequential Optimization and Reliability Assessment (IS-based modified SORA) method, which aims to reduce the number of calls to the expensive objective function and constraint functions in RBDO. The proposed method consists of three key stages. First, the samples are sequentially selected to construct Kriging models with high classification accuracy for each constraint function. Second, the samples are obtained by Markov Chain Monte Carlo in the safety domain of design space. Then, another Kriging model for the objective function is sequentially constructed by adding suitable samples to update the Design of Experiment (DoE) of the objective function. Third, the expensive objective and constraint functions of the original optimization problem are replaced by the surrogate models. Then, the IS-based modified SORA method is performed to decouple reliability optimization problem into a series of deterministic optimization problems that are solved by a Genetic Algorithm. Several examples are adopted to verify the proposed method. The optimization results show that the proposed method can reduce the number of calls to the original objective function and constraint functions without loss of precision compared to the alternative methods, which illustrates the efficiency and accuracy of the proposed method.

Keywords Reliability-based design optimization · Sequential Optimization and Reliability Assessment · Kriging model · Markov Chain Monte Carlo · Importance Sampling

1 Introduction

In traditional engineering design optimization, the optimum design is obtained by deterministic design optimization (DDO) with regard to the limits of the problem constraints [1]. However, there are various uncertainties (such as material properties, structural dimensions, boundary conditions and loads) in practical engineering problems, and this can lead to the constraint functions with high failure probability at the deterministic optimal solution [2]. Therefore, reliability-based design optimization (RBDO) methods have been developed to achieve a balance between cost and safety, and thus produce designs that are not only reliable but also economical for the engineering systems with uncertainties [3–5]. A typical RBDO problem can be formulated as:

$$\begin{aligned} &\text{Find } \boldsymbol{d} \\ &\min\ C(\boldsymbol{d}) \\ &\text{s.t.} \begin{cases} P\left[G_j(\boldsymbol{x}) \le 0 | \boldsymbol{d}\right] \le \Phi\left(-\beta_j^t\right), \\ j = 1, 2, \ldots, n_y, \\ \boldsymbol{d}_L \le \boldsymbol{d} \le \boldsymbol{d}_U \end{cases} \end{aligned} \tag{1}$$

where C is the objective or cost function; G_j is the jth constraint function (i.e. performance function); β_j^t is the target reliability index for the jth probabilistic constraint; $\boldsymbol{\Phi}$ is the standard normal cumulative distribution function; $\boldsymbol{d}$ represents distribution parameters of the random variable vector $\boldsymbol{x}$; n_y is the number of probabilistic constraints; $\boldsymbol{d}_L$ and $\boldsymbol{d}_U$ denote the lower and upper limits of distribution parameters $\boldsymbol{d}$, respectively.

Equation (1) shows that RBDO intrinsically involves a so-called double-loop procedure where the outer optimization loop includes inner loops of reliability analysis. In general, RBDO methods include simulation-based methods and analytical methods for solving the probabilistic constraints in Eq. (1) [2, 3].

✉ Kunling Song
songkunling@mail.nwpu.edu.cn

[1] School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

Simulation-based methods are accurate and robust methods, which are mainly used for reliability assessment of the constraint functions of a RBDO problem in the complex engineering application. In the simulation-based methods, Monte Carlo simulation (MCS) is one of the most accurate simulation methods for approximating the failure probability, but it always involves heavy computation work [6, 7]. Many sampling techniques have been proposed to reduce the required samples, such as Importance Sampling (IS) method [8], Subset Simulation (SS) method [9, 10] and Line Sampling (LS) method [11]. Nevertheless, these techniques are more difficult to solve RBDO problems with small probabilities or implicit constraint functions. Many researchers have proposed and developed some simulation-based methods for solving the RBDO problems to improve the computational efficiency and effectiveness, such as cell evolution method [12], sampling-based method with a reweighting scheme [13] and weighed simulation method (WSM) [14, 15]. The WSM method is one of the most promising simulation-based methods, and it involves generating uniformly distributed samples and applying the product of probability distribution functions (PDFs) of the variables as a weight index at any sample. The failure probability is then regarded as the ratio of the sum of the weight indices in failure domain over the sum of the indices in the entire optimization design domain. Many practical methods have been proposed based on WSM for solving RBDO problems [1, 2, 16, 17], such as WSM–RBDO method [1], hybrid PSO–WSM method [2], improved weighted average simulation approach [16], and hybrid WSM–NSGA-II method [17]. However, the above-mentioned simulation methods require many expensive black-box function calls, which are often unacceptable, especially for the RBDO problems with time-consuming simulation models.

In general, analytical methods are more efficient than simulation-based methods for special RBDO problems such as linear systems with normal random variables since they are almost gradient based. Those methods include the moment matching method [18] and the Most Probable Point (MPP) [19, 20]-based method. In analytical methods, reliability index approach (RIA) and performance measurement approach (PMA) are the most commonly used MPP-based methods [2, 3]. RIA can convert the reliability constraints into the constraints of reliability indexes at the current design point by First-Order Reliability Method (FORM) and Second-Order Reliability Method (SORM), and then the RBDO problem is transformed into a relatively simple DDO problem [21, 22]. PMA, a more stable and efficient method than RIA, is proposed to reduce the computational cost of RBDO. PMA 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 [23, 24]. Lee et al. proved that PMA is more stable and efficient than RIA [25]. Moreover, Du et al. proposed the SORA method, and two optimization procedures in each iteration circle of the SORA method are included: the optimization with the modified constraints and the optimization searching for the inverse MPPs [3, 26]. In this way, the two-level optimization of RBDO is decoupled into serial single-level optimizations, which greatly reduces the computational cost. More recently, some modified methods based on the RIA, PMA, and SORA and some other analytical methods [27–35] were suggested, such as single loop approach (SLA) [27], sequential approximate programming [28], reliable design space (RDS) method [5], penalty-based decoupled approach (PDA) [29], approximate sequential optimization reliability assessment (ASORA) method [30]. In recent years, Keshtegar and his team have an indispensable contribution to enrich the analytical methods for RBDO [36–42], such as adaptive hybrid approach (AHA) [36], modified mean value (MMV) method [37], self-adaptive modified chaos control (SMCC) method [38], Hybrid descent mean value (HDMV) approach [39], enhanced single-loop method (ESM) [40], second-order reliability method-based stepped up sequential optimization and reliability assessment (SSORA–SORM) approach [41], and adaptive conjugate single-loop approach (AC-SLA) [42]. Moreover, some relatively efficient analytical methods for RBDO have recently been proposed, such as enriched self-adjusted mean value (ESMV) method [43], augmented step size adjustment (ASSA) method [44] and hybrid conjugate mean value (HCMV) method [45]. However, such methods still require a large number of calls. Meanwhile, these methods based on the MPPs or gradient have large errors for the constraint functions with high nonlinear, even difficult for complex engineering problems with implicit function.

In addition, with the increasing complexity of engineering problems, more and more researchers are turning their attention to approximation techniques (also named surrogate model or meta-model method). Surrogate models are often constructed to replace the original objective or constraint functions to solve RBDO problems [6, 26], thereby the computational efficiency is increased. Chen et al. proposed a variety of Kriging based methods for RBDO, such as local adaptive sampling (LAS) method [46], important boundary sampling (IBS) method [47] and local approximation method using the most probable point (LMPP) [48]. Shi et al. [49] developed a RBDO method using first-order score function analysis and adaptive response surface that is obtained by integrating Bayesian metric and Gaussian process-based model bias correction method. Lehký et al. [50] proposed an inverse reliability method based on Artificial Neural Network (ANN) and small-sample simulation. Yang et al. [6] proposed a RBDO framework that combines Particle Swarm Optimization (PSO), Support Vector Machine (SVM) and SS to enhance computational efficiency. Li

et al. [26] proposed a sequential surrogate model method for reliability-based optimization (SSRBO) based on radial basis function (RBF) and MCS to reduce the number of the expensive black-box function calls in RBDO. Although these surrogate model-based RBDO methods can greatly reduce the number of calls to the original model, their computational efficiency is limited by the reliability assessment method. For example, the SSRBO with MCS [26] and the method proposed in [6] with SS have lower computational efficiency to assess the reliability of constraints for solving RBDO problems.

In this paper, a new RBDO method based on adaptive surrogate model and importance sampling-based modified SORA method is proposed to reduce the number of original objective function and constraint functions calls and ensure computational efficiency. In practical engineering problems, constraint conditions or boundary conditions are usually dependent with each other, which makes the required samples to construct the surrogate models of the constraint functions independent. So, we assume that the constraint functions are dependent on each other and the constraint function is independent of the objective function in this paper. First, the adaptive Kriging models of the constraint functions are sequentially constructed until the accuracy requirements are met. Second, sufficient samples are generated by MCMC in the safety domain of design space, and then another Kriging model for the objective function is constructed by adaptively adding suitable samples to update the DoE of the objective function. After the Kriging models of the constraint functions and the Kriging model of the objective function are constructed, the expensive objective and constraint functions of the original optimization problem are replaced by these Kriging models. Third, the IS-based modified SORA method that combines the advantage of IS in accuracy and the advantage of SORA in efficiency is performed to decouple reliability optimization problem into a series of deterministic optimization problems that are solved by a Genetic Algorithm (GA). At last, the optimal solution of the original RBDO problem is obtained.

The remainder of this paper is presented as follows. The basic theories and formulations of main methods used in our method are introduced in Sect. 2. The detailed principles and process of the proposed method are described and discussed in Sect. 3. Four numerical examples are used to validate the efficiency and accuracy of the proposed method in Sect. 4. Finally, conclusions and future work are summarized in Sect. 5.

2 Basic theories and formulations

2.1 Kriging model

Kriging model is a statistical theory-based interpolation technique, and it consists of a parametric linear regression model and a nonparametric stochastic process [51–54]. Assuming that those input variables are defined as $\boldsymbol{x}$ and the corresponding response is defined as $G(\boldsymbol{x})$, Kriging is given as:

$$\widehat{G}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{\beta},\boldsymbol{x}) + z(\boldsymbol{x}) = \boldsymbol{f}^T(\boldsymbol{x})\boldsymbol{\beta} + \boldsymbol{\delta}(\boldsymbol{x}), \tag{2}$$

where the averaged approximation response $F(\boldsymbol{\beta},\boldsymbol{x})$ is defined as an ordinary polynomial regression of $\boldsymbol{x}$, and it is simplified to be a constant in the ordinary Kriging, which is taken as $F(\boldsymbol{\beta},\boldsymbol{x}) = \beta$. All the following formulas are deduced with the simplification of ordinary Kriging, and $\widehat{G}(\boldsymbol{x})$ can be simplified as

$$\widehat{G}(\boldsymbol{x}) = \beta + \boldsymbol{\delta}(\boldsymbol{x}), \tag{3}$$

Here $\boldsymbol{\delta}(\boldsymbol{x})$ is a zero-mean stationary Gaussian process, and the auto-covariance of samples $\boldsymbol{x}$ and $\boldsymbol{w}$ is expressed as

$$\mathrm{cov}(\boldsymbol{\delta}(\boldsymbol{x}), \boldsymbol{\delta}(\boldsymbol{w})) = \sigma^2 R(\boldsymbol{x},\boldsymbol{w}), \tag{4}$$

where σ^2 is the variance of the process. Auto-correlation function $R(\boldsymbol{x},\boldsymbol{w})$ can also be considered by several functions, such as cubic correlation function, exponential correlation function, and gauss correlation function. The gauss correlation function is employed in this paper, and it is defined as Eq. (5).

$$R(\boldsymbol{x},\boldsymbol{w}) = \exp\left(-\sum_{i=1}^{n_x}\theta_i\left|x_i - w_i\right|^2\right), \tag{5}$$

where n_x is the dimension of design variables. x_i and w_i denote the ith component of variable $\boldsymbol{x}$ and variable $\boldsymbol{w}$, respectively. θ_i is the correlation parameter to ensure the Kriging model with an high flexibility.

$\boldsymbol{X} = \left[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{N_d}\right]^T\left(\boldsymbol{x}_i \in R^n\right)$ and $\boldsymbol{Z} = \left[G\left(\boldsymbol{x}_1\right),G\left(\boldsymbol{x}_2\right),\ldots,G\left(\boldsymbol{x}_{N_d}\right)\right]^T\left(G\left(\boldsymbol{x}_i\right) \in R\right)$ indicate N_d experimental samples and the true responses of these samples in DoE. We define $\boldsymbol{R} = \left[R\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\right]_{N_d\times N_d}$ and $\boldsymbol{F}$ as a $N_d \times 1$ unit vector, and the unbiased estimator $\widehat{\beta}$ and $\widehat{\sigma}^2$ of the unknown parameters β and σ^2 can be deduced with least squares method:

$$\widehat{\beta} = \left(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{Z}, \tag{6}$$

$$\widehat{\sigma}^2 = \frac{1}{N_d}\left(\boldsymbol{Z} - \boldsymbol{F}\widehat{\beta}\right)^T\boldsymbol{R}^{-1}\left(\boldsymbol{Z} - \boldsymbol{F}\widehat{\beta}\right), \tag{7}$$

The parameter β of regression coefficients and the parameter σ^2 of constant process are dependent on the correlation parameters $\boldsymbol{\theta}$ through the correlation matrix. Therefore, the correlation parameters $\boldsymbol{\theta}$ must first be determined by maximum likelihood estimation with Eq. (8).

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left\{ \boldsymbol{R}(\boldsymbol{\theta}) \left[\widehat{\sigma}^2\right]^{N_d} \right\}, \tag{8}$$

After determining $\boldsymbol{\theta}$, β and σ^2 are defined, the linear unbiased estimator $\widehat{G}(\boldsymbol{x}_p)$ and $\widehat{\sigma}^2_{\widehat{G}}(\boldsymbol{x}_p)$ of the response and Kriging variance for the sample $\boldsymbol{x}_p$ to be predicted can be computed by Eq. (9) and Eq. (10), respectively.

$$\widehat{G}(\boldsymbol{x}_p) = \widehat{\beta} + \boldsymbol{r}(\boldsymbol{x}_p)^T \boldsymbol{R}^{-1} \left(\boldsymbol{Z} - \boldsymbol{F}\widehat{\beta} \right), \tag{9}$$

$$\widehat{\sigma}^2_{\widehat{G}}(\boldsymbol{x}_p) = \widehat{\sigma}^2 \left(1 + \boldsymbol{u}(\boldsymbol{x}_p)^T \left(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F} \right)^{-1} \boldsymbol{u}(\boldsymbol{x}_p) - \boldsymbol{r}(\boldsymbol{x}_p)^T \boldsymbol{R}^{-1} \boldsymbol{r}(\boldsymbol{x}_p) \right), \tag{10}$$

where $\boldsymbol{r}(\boldsymbol{x}_p)^T$ is a N_d-dimensional row vector representing the dependence between the predicted sample $\boldsymbol{x}_p$ and each sample in experimental samples $\boldsymbol{X}$, and it can be expressed as $\boldsymbol{r}(\boldsymbol{x}_p)^T = \left[R(\boldsymbol{x}_p, \boldsymbol{x}_1), R(\boldsymbol{x}_p, \boldsymbol{x}_2), \ldots, R(\boldsymbol{x}_p, \boldsymbol{x}_{N_d})\right]^T$, and $\boldsymbol{u}(\boldsymbol{x}_p)^T = \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{r}(\boldsymbol{x}_p) - 1$.

The calculating process of the predictor response $\widehat{G}(\boldsymbol{x}_p)$ and the Kriging variance $\widehat{\sigma}^2_{\widehat{G}}(\boldsymbol{x}_p)$ can be implemented by the MATLAB DACE toolbox [52], which has been applied in several references [53, 54].

2.2 Markov chain Monte Carlo

Markov Chain Monte Carlo (MCMC) simulation is a powerful approach for sampling according to an arbitrary probability distribution. In MCMC, samples are simulated as the states of a Markov Chain which has the target probability distribution as its limit stationary distribution under the Traversal Hypothesis. Since the MCMC method with the Metropolis–Hastings algorithm becomes inefficient to simulate samples with many independent components, Au and Beck [10] proposed a component-wise (or modified M–H) algorithm to improve the sampling efficiency of MCMC. Instead of using an n-dimensional proposal PDF in the original method, each coordinate ξ_j of the pre-candidate ξ is generated from a one-dimensional proposal PDF $p_j^*(\xi_j|x_j)$, which depends on the jth coordinate x_j of the current state. And the proposal PDF $p_j^*(\xi_j|x_j)$ has a symmetry property, i.e. $p_j^*(\xi|x) = p_j^*(x|\xi)$.

In this paper, the MCMC simulation with component-wise algorithm is employed to generate N_m samples that are lying in the required safety region. The process of generating a sequence of samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_m}\}$ from a given or randomly generated sample $\boldsymbol{x}_1$ to $\boldsymbol{x}_{N_m}$ is briefly summarized as follows [55, 56]:

(1) Generate candidate sample $\widetilde{\boldsymbol{x}}$

(a) Generate ξ_j by sampling from the proposal PDF $p_j^*(\xi_j|x_{kj})$.

In the process of Markov Chain, the transfer from a state to another state is controlled by the proposal distribution. In this paper, an n-dimensional uniform distribution within the interval $[x_{kj} - l_j/2, x_{kj} + l_j/2](j = 1, 2, \ldots, n)$ is selected as the proposal distribution, that is,

$$p_j^*(\xi_j|x_{kj}) = \begin{cases} 1/l_j & \text{if } \left|\xi_j - x_{kj}\right| \le l_j/2 \\ 0 & \text{else} \end{cases}, \tag{11}$$

where ξ_j and x_{kj} are the jth component of n dimensional vector $\boldsymbol{\xi}$ and the kth sample $\boldsymbol{x}$ of MCMC, respectively. l_j is the side length of x_{kj} component of n-dimensional hypercube, and it determines the maximum allowable range of the next sample deviate from the current sample. l_j takes the empirical value $6N_m^{-1/(n+4)}$.

(b) Accept or reject ξ_j

$$\widetilde{x}_j = \begin{cases} \xi_j, \text{with probability } \min\left\{1, \frac{q_j(\xi_j)}{q_j(x_{kj})}\right\} \\ x_{kj}, \text{with probability} 1 - \min\left\{1, \frac{q_j(\xi_j)}{q_j(x_{kj})}\right\} \end{cases}, \tag{12}$$

where $q_j(\cdot)$ is the jth coordinate x_j of the limit stationary distribution $\boldsymbol{q}(\boldsymbol{x})$ of Markov Chain.

(2) Accept or reject $\widetilde{\boldsymbol{x}}$

$$\boldsymbol{x}_{k+1} = \begin{cases} \widetilde{\boldsymbol{x}} & \widetilde{\boldsymbol{x}} \text{in} \boldsymbol{D}_{\boldsymbol{S}} \\ \boldsymbol{x}_k & \widetilde{\boldsymbol{x}} \text{not in} \boldsymbol{D}_{\boldsymbol{S}} \end{cases}, \tag{13}$$

where $\boldsymbol{D}_{\boldsymbol{S}}$ is the required safety region. Equation (13) expresses that the next sample $\boldsymbol{x}_{k+1}$ is set to candidate sample $\widetilde{\boldsymbol{x}}$ or the last sample $\boldsymbol{x}_k$ according to whether candidate sample $\widetilde{\boldsymbol{x}}$ lies in $\boldsymbol{D}_{\boldsymbol{S}}$ or not.

Here, only a brief introduction to MCMC simulation is provided. For more details, the reader may refer to the references [10, 56].

2.3 Importance sampling for reliability assessment

Generally, failure probability P_F can be expressed as follows [8, 57],

$$P_F = P[G(\boldsymbol{x}) \le 0] = \int_{R^n} I_F(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{14}$$

where $\boldsymbol{x}$ denotes basic input variables, $G(\boldsymbol{x})$ is the performance function of constraint function, R^n shows the n-dimension variable space, I_F is the denoting function, $I_F(\boldsymbol{x}) = 1$ when $G(\boldsymbol{x}) \le 0$ and $I_F(\boldsymbol{x}) = 0$ when $G(\boldsymbol{x}) > 0$ as an opposite, and $f_{\boldsymbol{x}}(\boldsymbol{x})$ is the joint probability density function of basic variables.

The basic principle of importance sampling is to shift the sampling center towards the failure region to gain information more efficiently. Equation (14) can be expressed in Eq. (15) by introducing importance the sampling density function $h_{\boldsymbol{x}}(\boldsymbol{x})$.

$$P_F = \int_{R^n} I_F(\boldsymbol{x}) \frac{f_{\boldsymbol{x}}(\boldsymbol{x})}{h_{\boldsymbol{x}}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \approx \frac{1}{N_{IS}} \sum_{i=1}^{N_{IS}} I_F(\boldsymbol{x}_i) \frac{f_{\boldsymbol{x}}(\boldsymbol{x}_i)}{h_{\boldsymbol{x}}(\boldsymbol{x}_i)}, \tag{15}$$

In this paper, the sampling density function is determined by moving sampling center to the design point (also the most probable failure point) that obtained by simulation-based method.

2.4 Sequential optimization and reliability assessment

The sequential optimization and reliability assessment (SORA) method was proposed to decouple a double-loop process of the conventional RBDO method [3]. The flowchart of the SORA approach is shown in Fig. 1.

In SORA, a sequence of deterministic optimization and reliability estimation are repeatedly performed at each cycle. The PMA method is used to estimate the MPP $\boldsymbol{x}_j^{*(k)}$ with the design variables $\boldsymbol{d}^{(k)}$ of the kth cycle. A shift vector $\boldsymbol{s}_j^{(k)}$ is calculated as follows:

$$\boldsymbol{s}_j^{(k)} = \boldsymbol{d}^{(k)} - \boldsymbol{x}_j^{*(k)}, \tag{16}$$

The failure surface of the probabilistic constraint is shifted to the feasible direction based on the obtained shift vector $\boldsymbol{s}_j^{(k)}$. Deterministic optimization is performed with shifted constraints to find new design variables as follows:

$$\begin{aligned} &\text{Find } \boldsymbol{d}^{(k+1)} \\ &\min C\left(\boldsymbol{d}^{(k+1)}\right) \\ &\text{s.t.} \begin{cases} G_j\left(\boldsymbol{d}^{(k+1)}\right) \ge 0, \\ G_j\left(\boldsymbol{d}^{(k+1)} - \boldsymbol{s}_j^{(k)}\right) \ge 0, \\ \boldsymbol{d}_L \le \boldsymbol{d} \le \boldsymbol{d}_U, \quad j = 1, 2, n_y \end{cases} \end{aligned} \tag{17}$$

The process continues until the convergence condition is satisfied. The SORA method uses the reliability information from the previous cycle to shift to the violated deterministic constraints in the feasible domain.

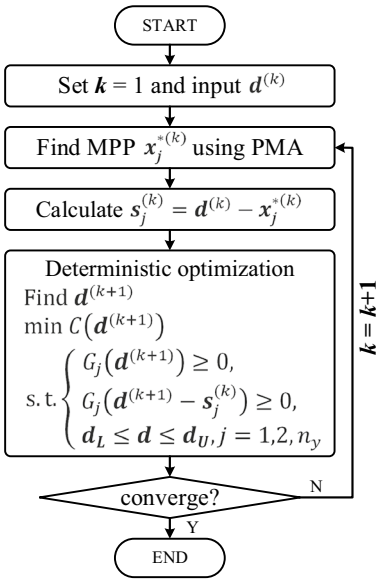

Fig. 1 The flowchart of the SORA method

3 The proposed RBDO method

3.1 Basic principles

In this paper, the proposed method can be treated as an improvement of the sequential optimization and reliability assessment (SORA) method by surrogate model and importance sampling. It constructs surrogate models of the constraint functions and the objective function in original RBDO problem and replaces them. Then, the surrogate model-based RBDO is solved by the Importance Sampling-based modified Sequential Optimization and Reliability Assessment (IS-based modified SORA) method that is proposed in this paper.

As shown in Fig. 2, the proposed method can be divided into three stages: constructing surrogate models of the constraint functions, constructing surrogate model of the objective function and solving RBDO by IS-based modified SORA. In the first phase, the surrogate models are constructed to approximate the classification boundaries of constraint functions with the active learning mechanism, which makes the constructed surrogate models with higher classification accuracy. In the second phase, a large number of samples are generated by the above-mentioned MCMC in the safety region with the constructed surrogate models of constraint functions. Subsequently, an initial surrogate model of the objective function is constructed, and it is sequentially updated with another active learning mechanism until the accuracy requirements are satisfied. After the first two phases are completed, the objective function and

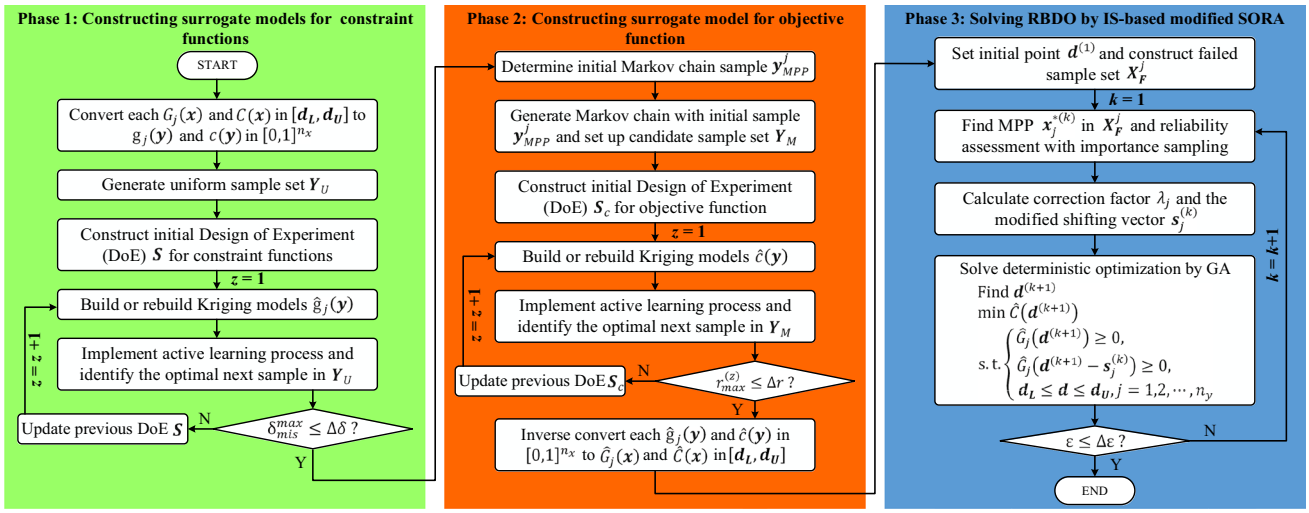

Fig. 2 The flowchart of the proposed method

constraint functions in original RBDO problem are replaced by the constructed surrogate models, respectively. At last, the third phase is executed with an IS-based modified SORA method, and then the original RBDO problem is decoupled into a sequence of deterministic optimization problem that is solved by a Genetic Algorithm (GA).

Detailed computational steps of each stage in the proposed method are explained in Sects. 3.2, 3.3 and 3.4, respectively.

3.2 Phase 1: constructing surrogate models of constraint functions

To further reduce the number of calls to the constraint functions and the objective function in the process of RBDO, the approximation technique is used to construct available surrogate models. Subsequently, the constraint functions and the objective function in the original RBDO problem are replaced by the corresponding surrogate models, respectively. Kriging model is one of the most popular surrogate models currently, and it is used in this paper.

In this section, the 'Phase 1' for constructing surrogate models of constraint functions in proposed method is introduced as the following steps. The flowchart of 'Phase 1' is shown in Fig. 2.

Step 1.1 **Transform design space** For the purpose of eliminating the influence of the variable magnitudes, the first step is the transformation of design variable or design space $[\boldsymbol{d_L}, \boldsymbol{d_U}]$, constraint functions $(G_j(\boldsymbol{x})(j = 1, 2, \ldots, n_y))$ and objective function $(C(\boldsymbol{x}))$. The optimization design interval of each variable is usually a closed interval that can be determined in reliability optimization problems, while the optimization range of half-open or full-open can also be determined according to the "6σ criterion". For simplicity, we assume that the optimization design interval of each variable is a closed interval. The design space $[\boldsymbol{d_L}, \boldsymbol{d_U}]$ is converted to $[0, 1]^{n_x}$ (n_x is the number of random variables), and the transformation between $\boldsymbol{y} \in [0, 1]^{n_x}$ and $\boldsymbol{x} \in [\boldsymbol{d_L}, \boldsymbol{d_U}]$ can be expressed as Eq. (18). Then, the functions $(g_j(\boldsymbol{y})(j = 1, 2, \ldots, n_y))$ are also converted to the corresponding constraint functions $G_j(\boldsymbol{x})(j = 1, 2, \cdots, n_y)$ by Eq. (19), and the function $(c(\boldsymbol{y}))$ is converted to the corresponding objective function $C(\boldsymbol{x})$ by Eq. (20)

$$\boldsymbol{y} = \frac{\boldsymbol{x} - \boldsymbol{x}_L}{\boldsymbol{d}_U - \boldsymbol{d}_L}, \tag{18}$$

$$G_j(\boldsymbol{x}) = G_j(\boldsymbol{y}(\boldsymbol{d}_U - \boldsymbol{d}_L) + \boldsymbol{d}_L) = g_j(\boldsymbol{y}), \tag{19}$$

$$C(\boldsymbol{x}) = C(\boldsymbol{y}(\boldsymbol{d}_U - \boldsymbol{d}_L) + \boldsymbol{d}_L) = c(\boldsymbol{y}). \tag{20}$$

Step 1.2 **Generate uniform sample set** The constructed surrogate models $\hat{g}_j(\boldsymbol{y})$ of constraint functions $g_j(\boldsymbol{y})$ should have higher classification accuracy in reliability analysis, and the training samples used to construct surrogate model should be updated to improve the accuracy of constructed surrogate models. So, a uniform sample set is adopted to catch the global capacity of the model classification and update the training samples, and it is termed as $\boldsymbol{Y}_U$. In this paper, the Halton sampling approach [58] as the one of available

uniform sampling approaches is employed to generate N_{us} uniform samples for constructing the set $\boldsymbol{Y}_U = \left[\boldsymbol{y}_U^1, \boldsymbol{y}_U^2, \ldots, \boldsymbol{y}_U^{N_{us}}\right]^T$ in $[0,1]^{n_x}$. The reference value of N_{us} is recommended to be set as $\left[a\sqrt{n_x}\right]$ (the symbol '$[\cdot]$' is an upwardly integer operator symbol), and it means that the average number of samples projected on the unit diagonal is equal to a. As a result of the inexpensive calculations of constructed surrogate models for constraint functions, the parameter a can be set to a larger value, and $a = 10^5$ is sufficient for a more accurate result in the proposed method.

Step 1.3 **Construct initial Design of Experiment (DoE) for constraint functions** An initial DoE is required to construct the initial surrogate models of constraint functions, and set $z = 1$. To ensure the spatial filling characteristics of the initial samples, the Maximum Minimum Distance Criterion is adopted to select N_d samples in $\boldsymbol{Y}_U$, sequentially. The number N_d of initial DoE is taken as $\left[b\sqrt{n_x}\right]$. b can generally be selected as 6–12, and it is taken as 6 in this paper. For more accurate initial surrogate model, a larger one should be taken. After determining the N_d samples, the functions $\mathrm{g}_j(\boldsymbol{y})\left(j = 1, 2, \ldots, n_y\right)$ are called to compute the values of corresponding constraint functions $G_j(\boldsymbol{x})\left(j = 1, 2, \ldots, n_y\right)$. The initial DoE $\boldsymbol{S}$ can be expressed as the following equation.

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{y}_S^1, \mathrm{g}_1\left(\boldsymbol{y}_S^1\right), \mathrm{g}_2\left(\boldsymbol{y}_S^1\right), \ldots, \mathrm{g}_{n_y}\left(\boldsymbol{y}_S^1\right) \\ \boldsymbol{y}_S^2, \mathrm{g}_1\left(\boldsymbol{y}_S^2\right), \mathrm{g}_2\left(\boldsymbol{y}_S^2\right), \ldots, \mathrm{g}_{n_y}\left(\boldsymbol{y}_S^2\right) \\ \vdots \\ \boldsymbol{y}_S^{N_d}, \mathrm{g}_1\left(\boldsymbol{y}_S^{N_d}\right), \mathrm{g}_2\left(\boldsymbol{y}_S^{N_d}\right), \ldots, \mathrm{g}_{n_y}\left(\boldsymbol{y}_S^{N_d}\right) \end{bmatrix}, \tag{21}$$

Step 1.4 **Build or rebuild Kriging models for constraint functions** In this step, the surrogate model $\widehat{g}_j(\boldsymbol{y})$ is constructed or reconstructed with the current DoE $\boldsymbol{S}_j$, and $\boldsymbol{S}_j$ expressed in Eq. (22) denotes the DoE for the jth function $\mathrm{g}_j(\boldsymbol{y})$ corresponding to the jth constraint function $G_j(\boldsymbol{x})$.

$$\boldsymbol{S}_j = \begin{bmatrix} \boldsymbol{y}_S^1, \mathrm{g}_j\left(\boldsymbol{y}_S^1\right) \\ \boldsymbol{y}_S^2, \mathrm{g}_j\left(\boldsymbol{y}_S^2\right) \\ \vdots \\ \boldsymbol{y}_S^{N_d}, \mathrm{g}_j\left(\boldsymbol{y}_S^{N_d}\right) \end{bmatrix}, \tag{22}$$

Step 1.5 **Implement active learning process and identify the optimal next sample** For adaptive surrogate model methods, one of the main concerns is to determine the location of the selected new training sample in iterations. In the adaptive process for constructing surrogate models, it is necessary to select the optimal new training samples with a learning strategy. Therefore, a learning strategy for constraint functions is developed to select the optimal new training samples that not only locate far away from the existing training samples, but also have a higher chance of being misclassified.

To make the optimal next sample far away from the existing training samples, the Euclidean distance d_i^j between ith sample $\boldsymbol{y}_U^i$ in $\boldsymbol{Y}_U$ and jth sample $\boldsymbol{y}_S^j$ in $\boldsymbol{S}$ can be calculated by Eq. (23). Then, the distance matrix $\boldsymbol{D}$ is obtained and presented in Eq. (24). Let $D_{\min}^i$ be the minimum value of the ith row of distance matrix $\boldsymbol{D}$, and it denotes the Euclidean distance between the sample $\boldsymbol{y}_U^i$ and each existing sample in $\boldsymbol{S}$. $D_{\min}^i$ is calculated by Eq. (25), and then the minimum distance matrix$\boldsymbol{D}_{\min}$ is obtained as presented in Eq. (26).

$$d_i^j = \| \boldsymbol{y}_U^i - \boldsymbol{y}_S^j \|, \tag{23}$$

$$\boldsymbol{D} = \begin{bmatrix} d_1^1 & d_1^2 & \cdots & d_1^{N_d} \\ d_2^1 & d_2^2 & & d_2^{N_d} \\ \vdots & & \ddots & \vdots \\ d_{N_{us}}^1 & d_{N_{us}}^2 & \cdots & d_{N_{us}}^{N_d} \end{bmatrix}, \tag{24}$$

$$D_{min}^i = \min[\boldsymbol{D}(i,:)], i = 1, 2, \ldots, N_{us}, \tag{25}$$

$$\boldsymbol{D}_{\min} = \left[D_{\min}^1, D_{\min}^2, \ldots, D_{\min}^{N_{us}}\right]^T, \tag{26}$$

After obtaining the minimum distance matrix $\boldsymbol{D}_{\min}$, the samples in $\boldsymbol{Y}_U$ are evaluated by $\widehat{\mathrm{g}}_j(\boldsymbol{y})$, and the predictor response $\widehat{\mathrm{g}}_j\left(\boldsymbol{y}_U^i\right)$ and the Kriging variance $\widehat{\sigma}_{\widehat{\mathrm{g}}_j}\left(\boldsymbol{y}_U^i\right)$ can be obtained. The results are stored in $\boldsymbol{Y}_P$ and $\boldsymbol{\sigma}_P$ as presented in Eq. (27) and Eq. (28), respectively. y_j^i and σ_j^i denote that the predictor response and the standard deviation of the ith sample $\boldsymbol{y}_U^i$ in $\boldsymbol{Y}_U$ are predicted by the jth function $\widehat{\mathrm{g}}_j(\boldsymbol{y})$, respectively. Subsequently, the misclassification probability p_m^{ij} of the ith sample $\boldsymbol{y}_U^i$ in $\boldsymbol{Y}_U$ for the jth function $\widehat{\mathrm{g}}_j(\boldsymbol{y})$ can be calculated by Eq. (29) [50, 54], and then the misclassification probability matrix $\boldsymbol{P}_m$ is obtained as presented in Eq. (30).

$$\boldsymbol{Y}_P = \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_{n_y}^1 \\ y_1^2 & y_2^2 & & y_{n_y}^2 \\ \vdots & & \ddots & \vdots \\ y_1^{N_{us}} & y_2^{N_{us}} & \cdots & y_{n_y}^{N_{us}} \end{bmatrix}, \tag{27}$$

$$\boldsymbol{\sigma}_P = \begin{bmatrix} \sigma_1^1 & \sigma_2^1 & \cdots & \sigma_{n_y}^1 \\ \sigma_1^2 & \sigma_2^2 & & \sigma_{n_y}^2 \\ \vdots & & \ddots & \vdots \\ \sigma_1^{N_{us}} & \sigma_2^{N_{us}} & \cdots & \sigma_{n_y}^{N_{us}} \end{bmatrix}, \tag{28}$$

$$p_m^{ij}\left(\boldsymbol{y}_U^i\right) = \boldsymbol{\Phi}\left(-\frac{\left|y_j^i\right|}{\sigma_j^i}\right), \tag{29}$$

$$\boldsymbol{P}_m = \begin{bmatrix} p_m^{11} & p_m^{12} & \cdots & p_m^{1n_y} \\ p_m^{21} & p_m^{22} & & p_m^{2n_y} \\ \vdots & & \ddots & \vdots \\ p_m^{N_{us}1} & p_m^{N_{us}2} & \cdots & p_m^{N_{us}n_y} \end{bmatrix}, \tag{30}$$

Step 1.6 For the purpose of meeting the requirement of the aforementioned learning strategy, the product $L_{DP}(i,j)$ of minimum distance $\boldsymbol{D}_{\min}(i)$ and $\boldsymbol{P}_m(i,j)$ is calculated by Eq. (31), and then the matrix $\boldsymbol{L}_{DP}$ is obtained as presented in Eq. (32). The maximum value $L_{\max}$ of matrix $\boldsymbol{L}_{DP}$ is computed by Eq. (33). Then, the element $L_{DP}\left(i_{\text{opt}},j_{\text{opt}}\right)$ in the matrix $\boldsymbol{L}_{DP}$ can be positioned by judging whether its value is equal to the maximum value $L_{\max}$, that is, $L_{DP}\left(i_{\text{opt}},j_{\text{opt}}\right) = L_{\max}$. Finally, the optimal next sample is determined in $\boldsymbol{Y}_U$, and it is named as $\boldsymbol{y}_U^{i_{\text{opt}}}$. Based on this learning strategy, it can be ensured that the selected optimal sample with a higher chance of being misclassified is located far away from the existing training samples.

$$L_{DP}(i,j) = D_{\min}^i p_m^{ij}, \tag{31}$$

$$\boldsymbol{L}_{DP} = \begin{bmatrix} L_{DP}(1,1) & L_{DP}(1,2) & \cdots & L_{DP}\left(1,n_y\right) \\ L_{DP}(2,1) & L_{DP}(2,2) & & L_{DP}\left(2,n_y\right) \\ \vdots & & \ddots & \vdots \\ L_{DP}\left(N_{us},1\right) & L_{DP}\left(N_{us},2\right) & \cdots & L_{DP}\left(N_{us},n_y\right) \end{bmatrix}, \tag{32}$$

$$L_{\max} = \max_{i,j}\left[L_{DP}(i,j)\right], \quad i = 1,2,\ldots,N_{us}, j = 1,2,\ldots,n_y, \tag{33}$$

Step 1.7 **Calculate convergence index and judge convergence criteria for Kriging models of constraint functions** The DoE and the constructed Kriging models of constraint functions are successively updated, and a stopping criteria is required. To fully utilize the binary classification feature in reliability analysis with constraint functions, a misclassification index between two adjacent iterations is proposed to judge whether the classification boundary of the constructed Kriging model is stable.

At the zth iteration, the classification label $C_{\text{label}}^{(z)}(i,j)$ with the prediction value y_j^i at the ith sample $\boldsymbol{y}_U^i$ in $\boldsymbol{Y}_U$ by the jth Kriging model $\hat{\mathrm{g}}_j(\boldsymbol{y})$ should be calculated by Eq. (34). Assume that $C_{\text{label}}^{(0)}(i,j)$ equals to 0. Subsequently, the classification label matrix $\boldsymbol{C}_{\text{label}}^{(z)}$ can be written as Eq. (35) according the Eq. (27). The misclassification index $\delta_{\text{mis}}^{(z)}(j)$ can describe the difference of classification boundary between jth constructed Kriging model $\hat{\mathrm{g}}_j$ at the zth iteration and the one at the (z-1)th iteration, and it can be calculated by Eq. (36).

$$C_{\text{label}}^{(z)}(i,j) = \begin{cases} 1 & y_j^i > 0 \\ -1 & y_j^i \le 0 \end{cases}, \tag{34}$$

$$\boldsymbol{C}_{\text{label}}^{(z)} = \begin{bmatrix} C_{\text{label}}^{(z)}(1,1) & C_{\text{label}}^{(z)}(1,2) & \cdots & C_{\text{label}}^{(z)}\left(1,n_y\right) \\ C_{\text{label}}^{(z)}(2,1) & C_{\text{label}}^{(z)}(2,2) & & C_{\text{label}}^{(z)}\left(2,n_y\right) \\ \vdots & & \ddots & \vdots \\ C_{\text{label}}^{(z)}\left(N_{us},1\right) & C_{\text{label}}^{(z)}\left(N_{us},2\right) & \cdots & C_{\text{label}}^{(z)}\left(N_{us},n_y\right) \end{bmatrix}, \tag{35}$$

$$\delta_{\text{mis}}^{(z)}(j) = \frac{1}{2N_{us}}\sum_{i=1}^{N_{us}}\left|C_{\text{label}}^{(z)}(i,j) - C_{\text{label}}^{(z-1)}(i,j)\right|, \tag{36}$$

For the purpose of making the constructed Kriging model of each constraint function meet the requirements of classification accuracy, the stopping criterion can be defined as

$$\delta_{\text{mis}}^{\max} \le \Delta\delta, \tag{37}$$

where $\delta_{\text{mis}}^{\max} = \max\left[\delta_{\text{mis}}^{(z)}(j)\right], \left(j = 1,2,\ldots,n_y\right)$,.$\Delta\delta$. is a small positive number that can be set by user. To balance accuracy and efficiency, $\Delta\delta$ can be taken as a number in the range: $1\times10^{-5} \sim 1\times10^{-4}$. A smaller value should be taken for a more accurate result. In this paper, $\Delta\delta$ is set to 5×10^{-5}, which is sufficient for obtaining Kriging models of each constraint function with an accurate and efficient classification result in the proposed method.

If the stopping criterion is satisfied, the active learning process is stopped, and the algorithm goes to next step, otherwise it goes to next Step 1.7.

Step 1.8 **Update previous DoE** If the stopping criterion is not satisfied, the previous Kriging models of each constraint function are not considered to be precise enough to approximate classification boundaries of the actual constraint functions. Consequently, previous DoE $\boldsymbol{S}$ should be updated. Firstly, the optimal next sample identified in Step 1.5 is evaluated by calling the functions $g_j(\boldsymbol{y})\left(j = 1, 2, \ldots, n_y\right)$, and then the evaluations and the sample are added to previous DoE $\boldsymbol{S}$. Simultaneously, set $z = z + 1$ and $N_d = N_d + 1$.

3.3 Phase 2: constructing surrogate model of objective function

Since the optimal design needs to meet the requirements of reliability constraints, it should be located in the safety domain determined by the constraint functions. Therefore, it is necessary that the constructed surrogate model of objective function has sufficient prediction accuracy for evaluating samples within the safety domain, and the prediction accuracy for samples within the failure domain can be relaxed. In this section, the 'Phase 2' for constructing surrogate models of objective function in proposed method is introduced in the following steps, and the flowchart of 'Phase 2' is presented in Fig. 2.

Step 2.1 **Determine initial sample of each Markov chain** In general, at least one constraint function is included in RBDO problem. To more uniformly simulate the samples in the safety domain, the samples are uniformly simulated using Markov chain Metropolis algorithm with multiple Markov chains, and the number of Markov chains is equal to the number n_y of constraint functions. The initial point $\boldsymbol{y}_0$ of the optimization iteration is defined as the mean point of the optimized design space, which is $\boldsymbol{y}_0 = [0.5, 0.5, \ldots, 0.5]_{1\times n_x}$. The joint probability density value of each sample in $\boldsymbol{Y}_U$ is evaluated with this initial point as the mean point of the reliability analysis, and then the maximum possible failure point $\boldsymbol{y}^j_{\mathrm{MPP}}$ in $\boldsymbol{Y}_U$ **is** determined by each Kriging model $\hat{g}_j(\boldsymbol{y})$.

Step 2.2 **Generate Markov chains and set up candidate sample set** n_y Markov chains are generated, and each chain contains N_m samples with the initial point $\boldsymbol{y}^j_{\mathrm{MPP}}$ of each constraint function. Subsequently, all samples in these Markov chains are used to set up the candidate sample set $\boldsymbol{Y}_M$, and so the number of samples in $\boldsymbol{Y}_M$ is $N_M = 3N_m$. In this work, the reference value of N_m is recommended to set as $\left\lceil h\sqrt{n_x}\right\rceil$. This means that the average number of samples projected on the unit diagonal is equal to h. As a result of the inexpensive calculations of constructed surrogate models for constraint functions in generating Markov chains, the parameter h can be set to a larger value. To balance the generation efficiency and filling ability of each Markov chain, $h = 10^4$ is set in the proposed method.

Step 2.3 **Construct initial Design of Experiment (DoE) for objective function** As mentioned in Step 1.3, an initial DoE is also required to construct the initial surrogate model of objective function, and set $z = 1$. To ensure the spatial filling characteristics of the initial samples, the maximum minimum distance criterion is adopted to select N_c samples in $\boldsymbol{Y}_M$, sequentially. The number N_c of initial DoE is the same as N_d in Step 1.3. After determining the N_c samples, the function $c(\boldsymbol{y})$ is called to compute the values of corresponding objective function $C(\boldsymbol{x})$. The initial DoE $\boldsymbol{S}_c$ can be expressed as

$$\boldsymbol{S}_c = \begin{bmatrix} \boldsymbol{y}^1_{S_c}, c\left(\boldsymbol{y}^1_{S_c}\right) \\ \boldsymbol{y}^2_{S_c}, c\left(\boldsymbol{y}^2_{S_c}\right) \\ \vdots \\ \boldsymbol{y}^{N_c}_{S_c}, c\left(\boldsymbol{y}^{N_c}_{S_c}\right) \end{bmatrix}, \tag{38}$$

Step 2.4 **Build or rebuild Kriging model for objective function** In this step, the surrogate model $\hat{c}(\boldsymbol{y})$ is constructed or reconstructed with the current DoE $\boldsymbol{S}_c$.

Step 2.5 **Implement active learning process and identify the optimal next sample** As constructing surrogate models for constrained functions, constructing surrogate model for objective function also requires an adaptive strategy to determine the optimal next sample. A learning strategy for constructing surrogate model of the objective function is also developed to select the optimal new training samples that not only locate far away from the existing training samples, but also have a higher chance to change current surrogate modelUsing the same calculation method in Step 1.5, the minimum distance matrix $\boldsymbol{D}^c_{\min}$ between samples in $\boldsymbol{Y}_M$ and samples in $\boldsymbol{S}_c$ is calculated, and it is presented in Eq. (39).

$$\boldsymbol{D}^c_{\min} = \left[D^{c1}_{\min}, D^{c2}_{\min}, \cdots, D^{cN_M}_{\min}\right]^T, \tag{39}$$

After obtaining the minimum distance matrix $\boldsymbol{D}^{c}_{\min}$, the samples in $\boldsymbol{Y}_M$ are evaluated by $\hat{c}(\boldsymbol{y})$. Then, the predictor response $\hat{c}\left(\boldsymbol{y}_M^i\right)$ and the Kriging variance $\hat{\sigma}_{\hat{c}}\left(\boldsymbol{y}_M^i\right)$ can be obtained. The results are stored in $\boldsymbol{Y}_c$ and $\boldsymbol{\sigma}_c$ as presented in Eq. (40) and Eq. (41). y_c^i and σ_c^i denote that the predictor response and the standard deviation of the ith sample $\boldsymbol{y}_M^i$ in $\boldsymbol{Y}_M$ are predicted, respectively.

$$\boldsymbol{Y}_C = \left[y_c^1, y_c^2, \ldots, y_c^{N_M}\right]^T, \tag{40}$$

$$\boldsymbol{\sigma}_C = \left[\sigma_c^1, \sigma_c^2, \ldots, \sigma_c^{N_M}\right]^T, \tag{41}$$

For the purpose of meeting the requirement of the aforementioned learning strategy for objective function, the product $L_{D\sigma}(i)$ of minimum distance $\boldsymbol{D}^{c}_{\min}(i)$ and $\boldsymbol{\sigma}_C(i)$ is calculated by Eq. (42), and then the $N_M \times 1$ matrix $\boldsymbol{L}_{D\sigma}$ is obtained as presented in Eq. (43). The maximum value $L_{\max}$ of matrix $\boldsymbol{L}_{DP}$ is computed by Eq. (44). Then, the element $L_{D\sigma}\left(i^M_{\text{opt}}\right)$ in the matrix $\boldsymbol{L}_{D\sigma}$ is positioned by judging whether its value is equal to the maximum value $L^{\max}_{D\sigma}$, that is, $L_{D\sigma}\left(i^M_{\text{opt}}\right) = L^{\max}_{D\sigma}$. Finally, the optimal next sample for updating the surrogate model of objective function is determined in $\boldsymbol{Y}_M$, and it is termed as $\boldsymbol{y}_M^{i^M_{\text{opt}}}$.

$$L_{D\sigma}(i) = D^{ci}_{\min}\sigma_c^i, \tag{42}$$

$$\boldsymbol{L}_{D\sigma} = \left[L_{D\sigma}(1), L_{D\sigma}(2), \ldots, L_{D\sigma}\left(N_M\right)\right]^T, \tag{43}$$

$$L^{\max}_{D\sigma} = \max_i \left[L_{D\sigma}(i)\right], i = 1, 2, \ldots, N_M, \tag{44}$$

Step 2.6 **Calculate convergence index and judge convergence criteria for surrogate model of objective function** The DoE and the constructed Kriging model of objective function are sequentially updated, a stopping criterion is required. To obtain a Kriging model with global precision for objective function in safety domain, a relative error between two adjacent iterations is used as an indicator to judge whether the prediction accuracy of the constructed Kriging model is enough at the zth iteration. The predictor response vector $\boldsymbol{Y}_c$ in Step 2.5 is stored in another vector $\boldsymbol{Y}_c^{(z)}$. Assume that $\boldsymbol{Y}_c^{(0)}(i) = 0\left(i = 1, 2, \ldots, N_M\right)$. The relative error $r^{(z)}(i)$ describes the difference of predictor response between constructed Kriging model $\hat{c}(\boldsymbol{y})$ at the zth iteration and the one at the (z-1)th iteration, and it can be calculated by Eq. (45).

$$r^{(z)}(i) = \left|\frac{\boldsymbol{Y}_c^{(z)}(i) - \boldsymbol{Y}_c^{(z-1)}(i)}{\boldsymbol{Y}_c^{(z)}(i)}\right|, \tag{45}$$

For the purpose of making the constructed Kriging model of the objective function meet the accuracy requirement, the stopping criterion can be defined as

$$r^{(z)}_{\max} \le \Delta r, \tag{46}$$

where $r^{(z)}_{\max} = \max\left[r^{(z)}(i)\right], \left(i = 1, 2, \ldots, N_M\right)$, Δr is a small positive number that can be set by user. In order to balance accuracy and efficiency, Δr can usually be taken as a number in the range: $1 \times 10^{-5} \sim 1 \times 10^{-4}$. A smaller value should be taken for a more accurate result. In this paper, Δr is set to 5×10^{-5}, which is sufficient to obtain an accurate and efficient Kriging model of the objective function.

If the stopping criterion is satisfied, the active learning process is stopped, and the algorithm goes to Step 2.8, otherwise it goes to Step 2.7.

Step 2.7 **Update previous DoE** If the stopping criterion is not satisfied, the previous Kriging model for objective function is considered not to be precise enough to approximate the function $c(\boldsymbol{y})$. Consequently, previous DoE $\boldsymbol{S}_c$ should be updated. Firstly, the optimal next sample identified in Step 2.5 is evaluated with the function $c(\boldsymbol{y})$. Then, the sample and its evaluation is added to update previous DoE $\boldsymbol{S}_c$. Simultaneously, set $z = z + 1$ and $N_c = N_c + 1$.

Step 2.8 **Inverse transform design space** After the Kriging model $\hat{c}(\boldsymbol{y})$ for objective function is determined, all constructed Kriging models should be converted to the original design space so as to replace the objective function and the constraint functions in original RBDO problem. Therefore, the Kriging models $\hat{c}(\boldsymbol{y})$ and $\hat{g}_j(\boldsymbol{y})\left(j = 1, 2, \ldots, n_y\right)$ are converted to the functions $\widehat{C}(\boldsymbol{x})$ and $\widehat{G}_j(\boldsymbol{x})\left(j = 1, 2, \ldots, n_y\right)$ by Eq. (47) and Eq. (48). Subsequently, the function $\widehat{C}(\boldsymbol{x})$ and the functions $\widehat{G}_j(\boldsymbol{x})\left(j = 1, 2, \ldots, n_y\right)$ are used to replace the objective function $C(\boldsymbol{x})$ and the constraint functions $G_j(\boldsymbol{x})\left(j = 1, 2, \ldots, n_y\right)$ in original RBDO problem as presented in Eq. (1), and then go to next phase for solving the RBDO problem with surrogate models as presented in Eq. (49).

$$\hat{c}(\boldsymbol{y}) = \hat{c}\left(\frac{\boldsymbol{x} - \boldsymbol{x}_L}{\boldsymbol{d}_U - \boldsymbol{d}_L}\right) = \widehat{C}(\boldsymbol{x}), \tag{47}$$

$$\hat{g}_j(\boldsymbol{y}) = \hat{g}_j\left(\frac{\boldsymbol{x}-\boldsymbol{x}_L}{\boldsymbol{d}_U-\boldsymbol{d}_L}\right) = \widehat{G}_j(\boldsymbol{x}), \tag{48}$$

$$\begin{aligned} &\text{Find } \boldsymbol{d} \\ &\min \widehat{C}(\boldsymbol{d}) \\ &\text{s.t.} \begin{cases} P\left[\widehat{G}_j(\boldsymbol{x}) \le 0 | \boldsymbol{d}\right] \le \Phi\left(-\beta_j^t\right), \\ j = 1, 2, \ldots, n_y, \\ \boldsymbol{d}_L \le \boldsymbol{d} \le \boldsymbol{d}_U \end{cases} \end{aligned} \tag{49}$$

3.4 Phase 3: solving RBDO by IS-based modified SORA

After the first two phases of the proposed method being executed, objective function $C(\boldsymbol{x})$ and the constraint functions $G_j(\boldsymbol{x})\left(j = 1, 2, \ldots, n_y\right)$ in the original RBDO problem (as presented in Eq. (1)) are replaced with the function $\widehat{C}(\boldsymbol{x})$ and the functions $\widehat{G}_j(\boldsymbol{x})\left(j = 1, 2, \ldots, n_y\right)$ (as presented in Eq. (49)), respectively. In this section, the RBDO problem as presented in Eq. (49) is solved by IS-based modified SORA. The IS-based modified SORA method can be treated as an improvement of SORA. The proposed IS-based MSORA method consists of the following steps, and the flowchart of 'Phase 3' is shown in Fig. 2.

Step 3.1 **Set initial point and construct failed sample set of each constraint function** Like most analytical methods of RBDO, the initial point for solving the RBDO problem is also needed, in the proposed IS-based MSORA method. So, set $k = 1$, and the initial point $\boldsymbol{d}^{(1)}$ is defined as the mean point of the optimization design space, which is $\boldsymbol{d}^{(1)} = 0.5\left(\boldsymbol{d}_L + \boldsymbol{d}_U\right)$. To avoid the disadvantages of solving the MPP-based gradient information of performance function in traditional method, the simulation-based method is used to find the MPP with the highest probability density value in the failure domain. Therefore, the failed sample sets $\boldsymbol{X}_F^j\left(j = 1, 2, \ldots, n_y\right)$ can be constructed. All samples in $\boldsymbol{Y}_U$ are converted to the design space $\left[\boldsymbol{d}_L, \boldsymbol{d}_U\right]$ with Eq. (50), and the sample set $\boldsymbol{X}_U$ is obtained. Then, all samples in $\boldsymbol{X}_U$ are classified by the sign of the response value that is calculated by calling the function $\widehat{G}_j(\boldsymbol{x})$, and the samples with negative response value are stored in $\boldsymbol{X}_F^j$.

$$\boldsymbol{x} = \boldsymbol{y}\left(\boldsymbol{d}_U - \boldsymbol{d}_L\right) + \boldsymbol{d}_L. \tag{50}$$

Step 3.2 **Find MPPs and reliability assessment with importance sampling** The sample $\boldsymbol{x}_j^{*(k)}$ with the highest probability density value in $\boldsymbol{X}_F^j$ for the corresponding function $\widehat{G}_j(\boldsymbol{x})$ is found at the current optimal solution $\boldsymbol{d}^{(k)}$, and it is considered the MPP of the function $\widehat{G}_j(\boldsymbol{x})$. Subsequently, N_{IS} important samples at each $\boldsymbol{x}_j^{*(k)}$ are generated by executing importance sampling algorithm, and these samples are stored in $\boldsymbol{X}_{\boldsymbol{IS}}^j$, respectively. Then, all samples in $\boldsymbol{X}_{\boldsymbol{IS}}^j$ are classified by the sign of the response value that is calculated by calling the function $\widehat{G}_j(\boldsymbol{x})$, and the samples with negative response value are added in $\boldsymbol{X}_{\boldsymbol{F}}^j$. Simultaneously, the failure probability P_{IS}^j of the function $\widehat{G}_j(\boldsymbol{x})$ is calculated by Eq. (15). In this paper, the number N_{IS} of important samples is taken as 1×10^5, and it is sufficient to obtain an accurate assessment for failure probability.

Step 3.3 **Calculate correction factors and modified shifting vectors** Failure probability evaluated by MPP is based on linear assumption in the traditional SORA method, and it is accurate if and only if the performance function is a linear function. Therefore, the SORA should be modified to obtain an accurate optimal design in the RBDO problem with the non-linear constraint function. In this paper, the shifting vector $\boldsymbol{s}_j^{(k)}$ is modified with a correction factor λ_j. The correction factor λ_j can be calculated by Eq. (51), and the shifting vector $\boldsymbol{s}_j^{(k)}$ can be calculated by Eq. (52).

$$\lambda_j = \frac{\parallel \left(\boldsymbol{d}^{(k)} - \boldsymbol{x}_j^{*(k)}\right)\boldsymbol{\sigma}^{-1} \parallel}{\boldsymbol{\Phi}^{-1}\left(1 - P_{IS}^j\right)}, \tag{51}$$

$$\boldsymbol{s}_j^{(k)} = -\beta_j^t \lambda_j \frac{\left[\left(\boldsymbol{d}^{(k)} - \boldsymbol{x}_j^{*(k)}\right)\boldsymbol{\sigma}^{-1}\right]}{\parallel \left(\boldsymbol{d}^{(k)} - \boldsymbol{x}_j^{*(k)}\right)\boldsymbol{\sigma}^{-1} \parallel}\boldsymbol{\sigma}, \tag{52}$$

Step 3.4 **Solve deterministic optimization by GA** After determining each shifting vector $\boldsymbol{s}_j^{(k)}$, the failure surfaces of the probabilistic constraints of the RBDO problem in Eq. (49) can be shifted to the feasible direction. Deterministic optimization is performed with shifted constraints to find a new design as presented in Eq. (53), and it is solved by GA in the proposed method.

$$\begin{aligned} &\text{Find } \boldsymbol{d}^{(k+1)} \\ &\min \widehat{C}\left(\boldsymbol{d}^{(k+1)}\right) \\ &\text{s.t.} \begin{cases} \widehat{G}_j\left(\boldsymbol{d}^{(k+1)}\right) \ge 0, \\ \widehat{G}_j\left(\boldsymbol{d}^{(k+1)} - \boldsymbol{s}_j^{(k)}\right) \ge 0, \\ \boldsymbol{d}_L \le \boldsymbol{d} \le \boldsymbol{d}_U, \quad j = 1, 2, \ldots, n_y \end{cases} \end{aligned} \tag{53}$$

Step 3.5 **Check stopping criterion** The stopping criterion is defined as Eq. (54), and ε can be calculated by

Eq. (55). If it is satisfied, then stop and go to Step 3.6, otherwise set $k=k+1$ and go to Step 3.2. In this paper, the allowable value is taken as 1×10^{-3}.

$$\varepsilon \le \Delta\varepsilon, \tag{54}$$

$$\varepsilon = \sum_{i=1}^{n_x}\left|\frac{\boldsymbol{d}^{(k+1)}(i)-\boldsymbol{d}^{(k)}(i)}{\boldsymbol{d}^{(k+1)}(i)}\right|, \tag{55}$$

Step 3.6 **End the algorithm of the proposed RBDO method** The last design $\boldsymbol{d}^{(k+1)}$ is considered as the final result of the proposed RBDO method, which is accurate enough for the original RBDO problem

The proposed method not only guarantees the classification accuracy of the constructed surrogate models for the constraint function and the approximation accuracy of the constructed surrogate model for the objective function, but also inherits the accuracy of important sampling and the high efficiency of SORA for solving RBDO problem. Compared with other conventional RBDO methods based on the MPPs or gradient, its computational efficiency and accuracy can be increased greatly.

4 Numerical examples

4.1 A nonlinear 2D test problem

To better illustrate the proposed method, a two-dimensional example is tested, and it has been used as the verification example in the literatures [32, 42, 44, 45, 59]. This RBDO problem contains three constraints, the second one is highly nonlinear and others are weakly nonlinear. There are two random variables which obey the normal distribution $x_i \sim N\left(d_i, 0.3^2\right)$, $i=1,2$. The target reliability index is $\beta_j^t = 3, j=1,2,3$.

The formulation for RBDO can be given by Eq. (56).

$$\begin{aligned}
&\text{Find } \boldsymbol{d} = \left[d_1, d_2\right]\\
&\min f(\boldsymbol{d}) = -\frac{\left(d_1+d_2-10\right)^2}{30} - \frac{\left(d_1-d_2+10\right)^2}{120}\\
&\text{s.t.}\begin{cases} P\left[G_j(\boldsymbol{x})\le 0|\boldsymbol{d}\right] \le \boldsymbol{\Phi}\left(-\beta_j^t\right), \quad \beta_j^t = 3, \quad j=1,2,3\\ 0\le d_1\le 10, 0\le d_2\le 10,\\ x_i\sim N\left(d_i,0.3^2\right), \quad i=1,2\end{cases}\\
&\text{where} \quad G_1(\boldsymbol{x}) = x_1^2x_2/20-1,\\
&\qquad G_2(\boldsymbol{x}) = 1-\left[(Y-6)^2+(Y-6)^3-0.6(Y-6)^4+Z\right],\\
&\qquad G_3(\boldsymbol{x}) = 80/\left(x_1^2+8x_2+5\right)-1,\\
&\qquad Y = 0.9063x_1+0.4226x_2\\
&\qquad Z = 0.4226x_1-0.9063x_2
\end{aligned} \tag{56}$$

To better illustrate the proposed method, these main processes of constructing surrogate models of constraint functions, constructing a surrogate model of the objective function and solving RBDO by IS-based modified SORA are separately shown in Figs. 3, 4 and 5. Figure 3a shows the initial surrogate models for constraint functions with the initial DoE $\boldsymbol{S}$ which contains 9 samples. As the number of samples in DoE $\boldsymbol{S}$ is constantly increased, the surrogate models for constraint functions gradually converge. The surrogate models for the first, the second and the third constraint function are considered stable at the 11th iteration (Fig. 3b), at the 27th iteration (Fig. 3c) and at the 40th iteration (Fig. 3d), respectively. Subsequently, the surrogate models for constraint functions obtained at the 40th iteration are used to generate Markov chains and set up candidate sample set $\boldsymbol{Y}_M$, as shown in Fig. 4a. Then, the DoE $\boldsymbol{S}_c$ is constructed as presented in Fig. 4b, and the surrogate model of objective function is constructed. After surrogate models of the constraint functions and objective function are determined, these surrogate models are used to replace the original constraint functions and objective function. Hereafter, the IS-based modified SORA method is executed to solve the optimal solution. Figure 5 shows the variation of the optimal solution with optimization process.

The probabilistic constraints at the optimal solution are evaluated by the Monte Carlo simulation (MCS) with ten million samples and listed in Table 1 in comparison with those methods by previous publications. The reliability optimization results are captured based on the methods listed in Table 1, and the optimal design variables, optimal value of objective function, and the total number of constraint function calls and objective function calls are also summarized. N_{total}, N_{con} and N_{obj} represent the total evaluation number of constraint function calls and objective function calls, the evaluation number of constraint function calls and the evaluation number of objective function calls in the original RBDO problem, respectively.

It can be seen that almost all reliability optimization methods listed in Table 1 converge to a similar optimum result. The optimum result of the AC-SLA is less than that of the other methods including the proposed method, but it is much more than the proposed method in the evaluation number of constraint function calls and objective function calls. In addition, the optimum result of AC-SLA does not satisfy the reliability constraint with $\beta^t \ge 3$ for the first constraint function. Therefore, the proposed method is more efficient and accurate than the other methods.

4.2 Optimal design of a roof truss

A roof truss subject to the uniform loads is to be designed, and it has been studied in the literatures [1, 39, 43, 56]. As shown in Fig. 6, top members and compression bars are

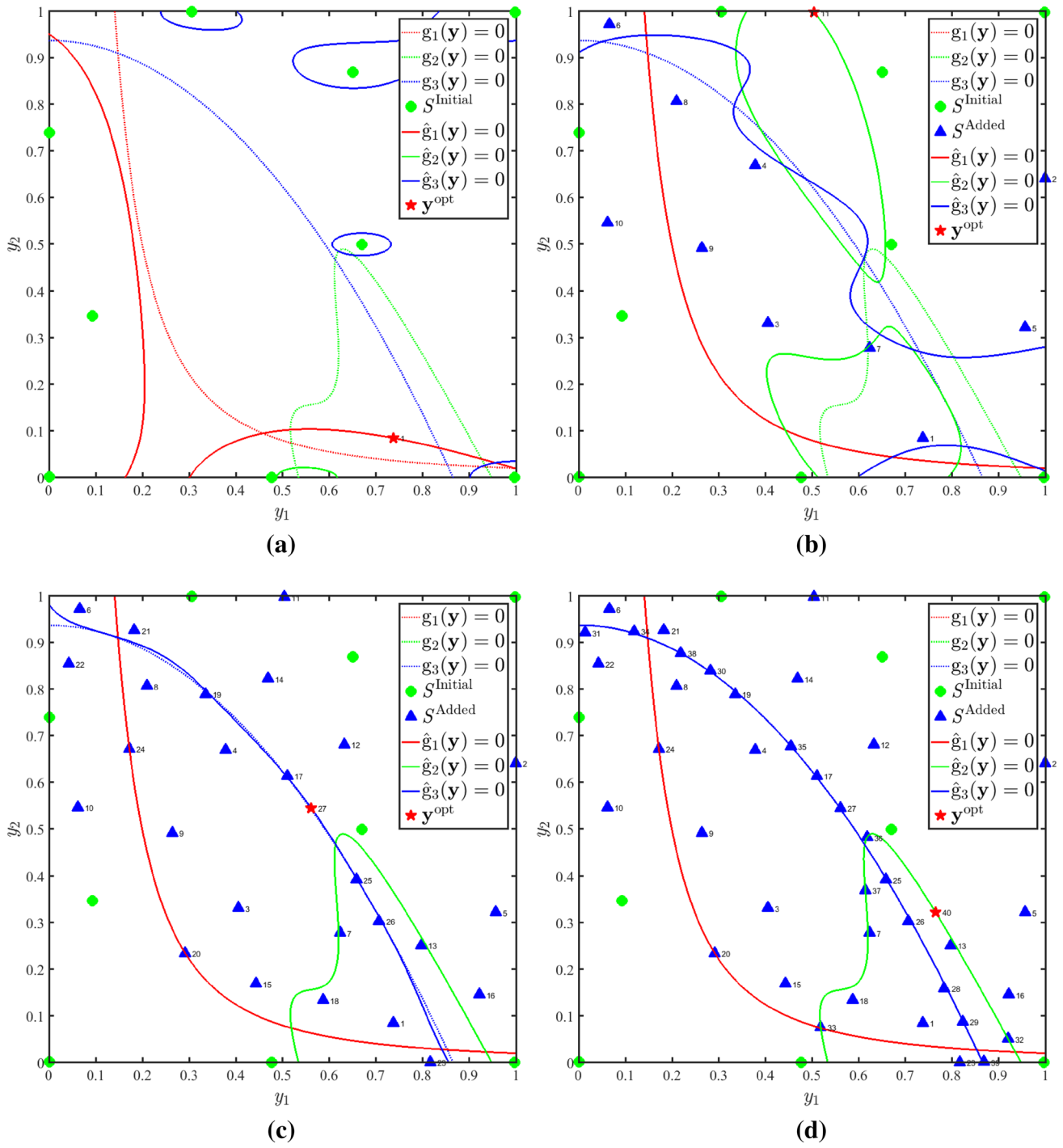

Fig. 3 Constructing surrogate models of constraint functions for example 1. **a** The initial surrogate models; **b** The surrogate models at the 11th iteration; **c** The surrogate models at the 27th iteration; **d** The surrogate models at the 40th iteration or the final surrogate models

made by reinforced concrete, and bottom members and tension bars are made by steel. Table 2 presents parameters of random variables. The vertical deflection at the peak of structure (node C) is limited to 0.03 m, and the problem is given as follows:

$$\begin{aligned} &\text{Find } \boldsymbol{d} = [A_S, A_C] \\ &\min f(\boldsymbol{d}) = 20224A_S + 364A_C \\ &\text{s.t.} \begin{cases} P\left[G(\boldsymbol{x}) = 0.03 - \left(\frac{ql^2}{2}\right)\left(\frac{3.81}{A_C E_C} + \frac{1.13}{A_S E_S}\right) \le 0 | \boldsymbol{d}\right] \le \Phi(-\beta^t) \\ 0.0006 \le A_S \le 0.0012, 0.018 \le A_C \le 0.063, \quad \beta^t = 3 \end{cases} \end{aligned} \tag{57}$$

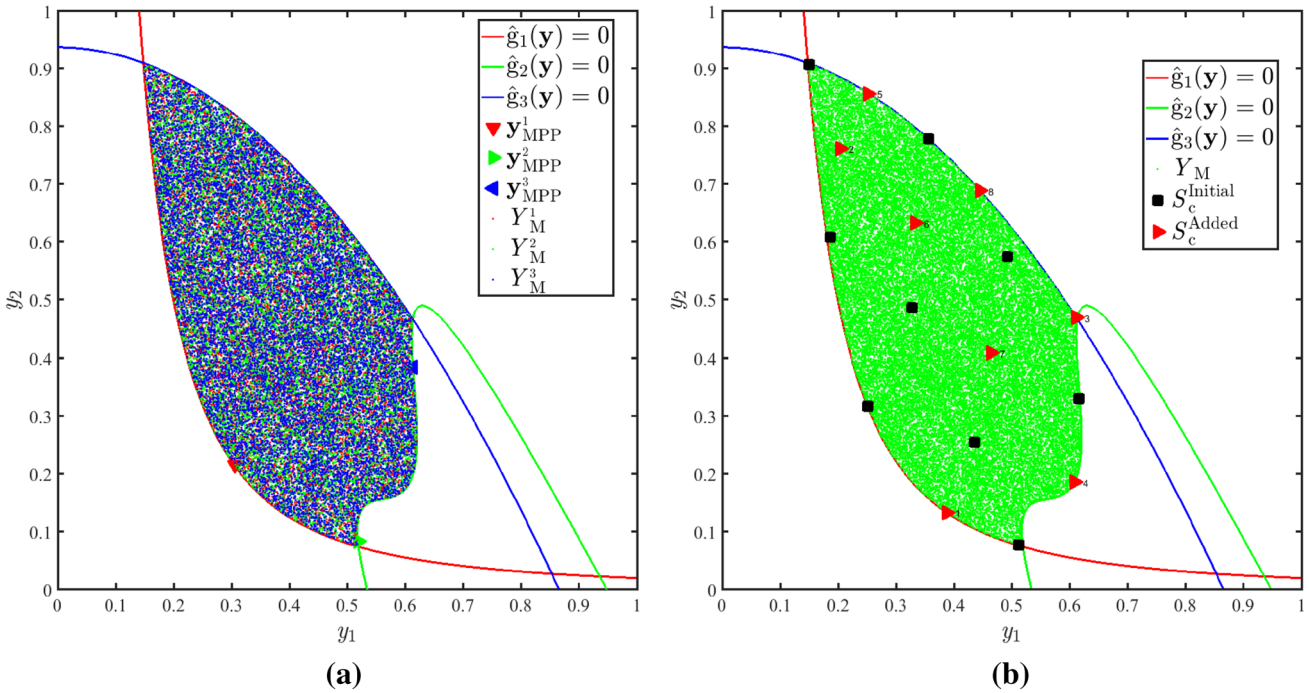

Fig. 4 Constructing surrogate model of objective function for example 1. **a** Generate Markov chains and set up candidate sample set; **b** The final DoE for constructing surrogate model of objective function

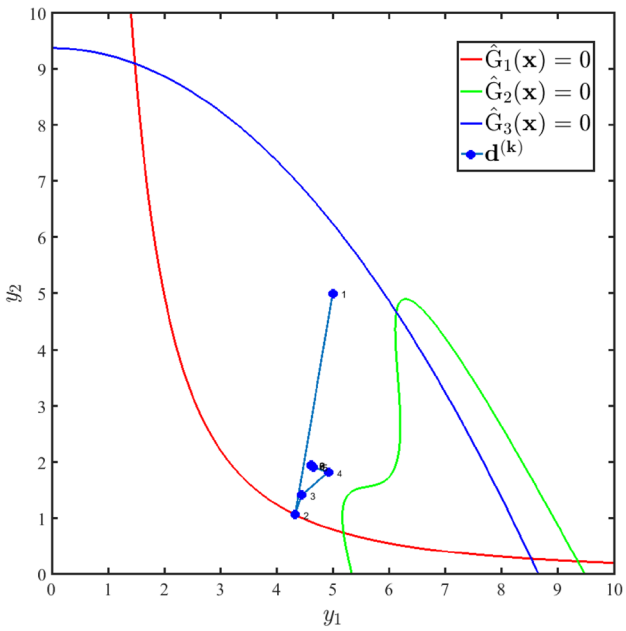

Fig. 5 The variation of the optimal solution with optimization process of the phase 3 for example 1

This example involves four normal random variables and two normal random design variables i.e. A_S, A_C. The distribution parameters of the random variables are listed in Table 2.

Similar to example 1, the optimal results with the proposed method and those methods by previous publications are listed in Table 3. It can be seen that the HCC, RMV, MMV, AAMV, DMV, HDMV and ESMV method converge to the similar unreliable optimal results, which does not satisfy the reliability constraint with $\beta^t \geq 3$; while the DCC, RBDO + WAM and the proposed method converges to the reliable optimal results with $\beta^t \geq 3$. The result of RBDO + WAM is better than that of DCC and proposed method on the objective function, but it is less efficient than all other methods for this example. As presented in Table 3, the total number of constraint function calls and objective function calls of the proposed method is much less than that of other methods for obtaining a reliable and nearly optimal result. So, the proposed method is more efficient than the other methods listed in Table 3.

Table 1 The reliability optimization results for example 1

Methods	Design variables	Objective	$N_{\text{total}}(N_{\text{con}}+N_{\text{obj}})$	β_1^{MCS}	β_2^{MCS}	β_3^{MCS}
PMA [32]	(4.569, 1.956)	− 1.728	1754(1690 + 64)	2.98	3.02	–
SLSV [32]	(4.575, 1.954)	− 1.729	839(617 + 222)	2.99	3.05	–
SORA [32]	(4.559, 1.964)	− 1.725	679(577 + 102)	2.97	3.02	–
SLSVCG [32]	(4.557, 1.964)	− 1.726	371(276 + 95)	2.97	3.06	–
RMV [42]	(4.558, 1.965)	− 1.725	762	2.95	3.20	+∞
HCC [42]	(4.558, 1.964)	− 1.725	312	2.95	3.21	+∞
AC-SLA [42]	(4.606, 1.935)	− 1.737	162	2.98	3.03	+∞
ACC [28]	(4.558, 1.965)	− 1.725	861	2.95	–	+∞
ASSA [44]	(4.558, 1.965)	− 1.725	729	2.96	–	+∞
HCMV [45]	(4.558, 1.965)	− 1.725	699	2.96	–	+∞
DCC [59]	(4.558, 1.965)	− 1.725	567	2.95	–	+∞
The proposed method	(4.623, 1.946)	− 1.732	66(49 + 17)	3.01	3.01	+∞

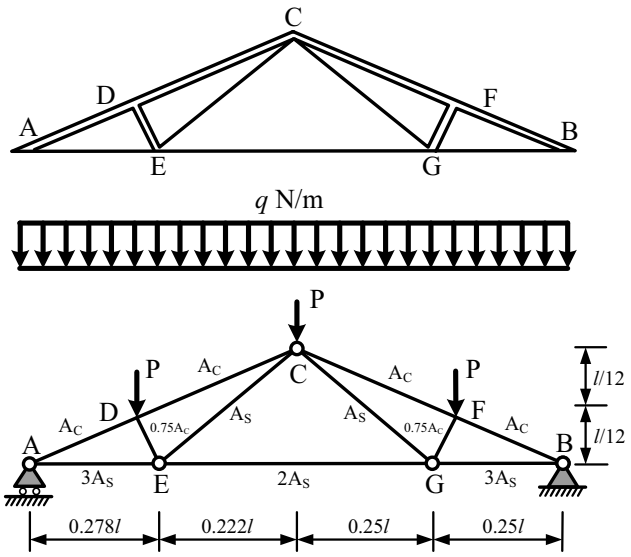

Fig. 6 Loads, reinforced concrete and steel bares of the roof truss for example 2

4.3 Welded beam problem

As shown in Fig. 7, this example considered is the welded beam problem, which has been used as a benchmark to compare with some previous publications by many scholars such as Yi et al. [30], Chen et al. [33] and Keshtegar et al. [37, 40, 45, 60]. The problem has five probabilistic constraints involving four random variables. The probabilistic constraints are related to limitation of shear stress, bending stress, buckling, and displacement. The objective of this problem is to minimize the welding cost. All random variables are statistically independent and follow normal distribution. The fixed system parameters z and c of the welded beam problem are listed in Table 4.

The RBDO model of the welded beam problem can be formulated as:

Table 2 Basic random variables of the roof truss for example 2

Random variable	Description	Distribution	Mean	Standard deviation
$q(N/m)$	Load	Normal	20,000	1400
$l(m)$	Length of truss	Normal	12	0.12
$A_S(m^2)$	Sectional area of steel bars	Normal	Design variable	5.892×10^{-5}
$A_C(m^2)$	Sectional area of reinforced concrete bars	Normal	Design variable	4.8×10^{-3}
$E_S(Pa)$	Elastic modulus of steel	Normal	1×10^{11}	6×10^{9}
$E_c(Pa)$	Elastic modulus of reinforced concrete	Normal	2×10^{10}	1.2×10^{9}

Table 3 The reliability optimization results for example 2

Methods	Design variables	Objective	$N_{\text{total}}\left(N_{\text{con}}+N_{\text{obj}}\right)$	β^{MCS}
HCC [39]	$(1.1202\times10^{-3}, 3.640\times10^{-2})$	35.90	6929	2.86
RMV [39]	$(1.0886\times10^{-3}, 3.836\times10^{-2})$	35.98	11,012	2.90
MMV [39]	$(1.0915\times10^{-3}, 3.820\times10^{-2})$	35.98	22,836	2.92
AAMV [39]	$(1.0808\times10^{-3}, 3.880\times10^{-2})$	35.98	5764	2.91
DMV [39]	$(1.0752\times10^{-3}, 3.896\times10^{-2})$	35.99	3141	2.91
HDMV [39]	$(1.0886\times10^{-3}, 3.836\times10^{-2})$	35.98	1405	2.92
ESMV [43]	$(1.0676\times10^{-3}, 4.004\times10^{-2})$	36.16	2102	2.98
DCC [59]	$(1.1008\times10^{-3}, 3.865\times10^{-2})$	36.33	1102	3.00
RBDO + WAM [1]	$(1.0680\times10^{-3}, 4.046\times10^{-2})$	36.24	40,000	3.01
The proposed method	$(1.0879\times10^{-3}, 3.944\times10^{-2})$	36.36	126(110 + 16)	3.02

$$
\begin{aligned}
&\text{Find } \boldsymbol{d} = \left[d_1, d_2, d_3, d_4\right]\\
&\min f(\boldsymbol{d}) = c_1 d_1^2 d_2 + c_2 d_3 d_4\left(z_2 + d_2\right)\\
&\text{s.t.}\begin{cases} P\left[G_j(\boldsymbol{x}) \le 0 | \boldsymbol{d}\right] \le \Phi\left(-\beta_j^t\right), \quad \beta_j^t = 3, \quad j = 1, 2, \ldots, 5\\ 3.175 \le d_1 \le 10, x_1 \sim N\left(d_1, 0.1693^2\right)\\ 150 \le d_2 \le 254, x_2 \sim N\left(d_2, 0.1693^2\right)\\ 200 \le d_3 \le 220, x_3 \sim N\left(d_3, 0.0107^2\right)\\ 3.175 \le d_4 \le 10, x_4 \sim N\left(d_4, 0.0107^2\right)\end{cases}\\
&\text{where} \quad G_1(\boldsymbol{x}) = 1 - \tau(\boldsymbol{x})/z_6, G_2(\boldsymbol{x}) = 1 - \sigma(\boldsymbol{x})/z_7,\\
&\qquad G_3(\boldsymbol{x}) = 1 - x_1/x_4, G_4(\boldsymbol{x}) = 1 - \delta(\boldsymbol{x})/z_5,\\
&\qquad G_5(\boldsymbol{x}) = P_c(\boldsymbol{x})/z_1 - 1,\\
&\qquad \tau(\boldsymbol{x}) = \left[t(\boldsymbol{x})^2 + \frac{2t(\boldsymbol{x})tt(\boldsymbol{x})x_2}{2R(\boldsymbol{x})} + tt(\boldsymbol{x})^2\right]^{0.5},\\
&\qquad t(\boldsymbol{x}) = \frac{z_1}{\sqrt{2}x_1x_2}, tt(\boldsymbol{x}) = \frac{M(\boldsymbol{x})R(\boldsymbol{x})}{J(\boldsymbol{x})},\\
&\qquad M(\boldsymbol{x}) = z_1\left(z_2 + 0.5x_2\right), R(\boldsymbol{x}) = \sqrt{\left[x_2^2 + \left(x_1 + x_3\right)^2\right]/4},\\
&\qquad J(\boldsymbol{x}) = \sqrt{2}x_1x_2\left[x_2^2/12 + \left(x_1 + x_3\right)^2/4\right],\\
&\qquad \sigma(\boldsymbol{x}) = \frac{6z_1z_2}{x_3^2x_4}, \delta(\boldsymbol{x}) = \frac{4z_1z_2^3}{z_3x_3^3x_4},\\
&\qquad P_c(\boldsymbol{x}) = \frac{4.013x_3x_4^3\sqrt{z_3z_4}}{6z_2^2}\left(1 - \frac{x_3}{4z_2}\sqrt{\frac{z_3}{z_4}}\right)
\end{aligned}
\tag{58}
$$

The RBDO results of this problem with the proposed method and previous methods are summarized in Table 5. It is seen that the objective function values of the optimal results obtained by the proposed method agree relatively well with those of other previous publications. Among these methods listed in Table 5, the SORA, ASORA, ADA, ESM, HCMV and the proposed method are relatively efficient in terms of the total number of constraint function calls and objective function calls. It can be seen that the proposed method is the best one in terms of the total number and the objective function value. So, the proposed method is more efficient than other methods to be employed for solving the RBDO problem of the welded beam.

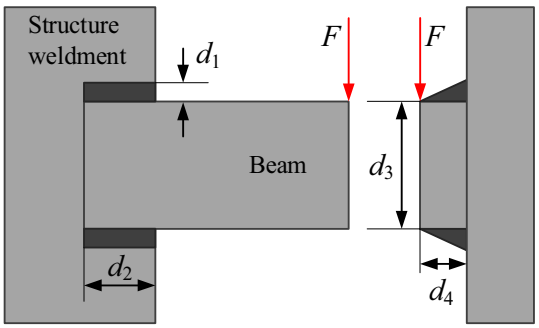

Fig. 7 The welded beam structure for example 3

Table 4 System parameters of the welded beam problem for example 3

z_1	2.6688×10^4 (N)	z_6	9.377×10^1 (MPa)
z_2	3.556×10^2 (mm)	z_7	2.0685×10^2 (MPa)
z_3	2.0685×10^5 (MPa)	c_1	6.74135×10^{-5} (\$/mm^3)
z_4	8.274×10^4 (MPa)	c_2	2.93585×10^{-6} (\$/mm^3)
z_5	6.35 (mm)		

4.4 Speed reducer RBDO problem

A speed reducer illustrated in Fig. 8 is used to rotate the engine and propeller with efficient velocity in light plane, and it is commonly studied by researchers as a benchmark example [2, 26, 30, 33, 36, 42, 45–47]. This RBDO problem has 11 probabilistic constraints involving 7 random variables. The probabilistic constraints are related to physical quantities such as bending stress, contact stress, longitudinal displacement, stress of the shaft, and geometry constraints. The random design variables are gear width (d_1), gear module (d_2), the number of pinion teeth (d_3), distance between bearings (d_4, d_5), and diameter of each shaft (d_6, d_7).

All random variables are statistically independent and have normal distributions. The objective function is to

Table 5 The reliability optimization results for example 3

Methods	Design variables	Objective	$N_{\text{total}}(N_{\text{con}}+N_{\text{obj}})$	β_1^{MCS}	β_2^{MCS}	β_3^{MCS}	β_4^{MCS}	β_5^{MCS}
RIA [30]	(5.730, 200.91, 210.60, 6.239)	2.591	9613(9590+23)	3.01	3.01	3.00	+∞	3.01
PMA [30]	(5.728, 200.97, 210.71, 6.238)	2.592	1122(1105+117)	3.01	3.53	3.01	+∞	2.98
SLSV [30]	(5.728, 200.99, 210.72, 6.238)	2.592	715(650+65)	3.01	3.58	3.01	+∞	2.98
SORA [30]	(5.731, 200.93, 210.64, 6.242)	2.592	198(125+73)	3.01	3.58	3.01	+∞	2.98
ASORA [30]	(5.731, 200.93, 210.64, 6.242)	2.592	167(90+77)	3.01	3.58	3.01	+∞	2.98
SORA–ICDE [31]	(5.730, 201.00, 210.63, 6.240)	2.593	2119	3.01	3.29	3.00	+∞	3.12
ADA [33]	(5.730, 200.91, 210.60, 6.239)	2.591	479(404+75)	3.00	3.00	3.00	+∞	3.00
AMV [37]	(5.730, 200.90, 210.60, 6.239)	2.591	1365(1350+15)	–	–	–	–	–
HMV [37]	(5.730, 200.90, 210.60, 6.239)	2.591	1365(1350+15)	–	–	–	–	–
MCC [37]	(5.730, 200.90, 210.60, 6.239)	2.591	12,185(12,170+15)	–	–	–	–	–
MMV [37]	(5.730, 200.90, 210.60, 6.239)	2.591	1144(1135+9)	–	–	–	–	–
ESM [40]	(5.730, 200.90, 210.60, 6.239)	2.591	371(360+11)	–	–	–	–	–
HCMV [45]	(5.730, 200.90, 210.60, 6.239)	2.591	854	–	–	–	–	–
SCG [60]	(5.730, 200.90, 210.60, 6.239)	2.591	987(975+12)	–	–	–	–	–
The proposed method	(5.730, 200.90, 210.60, 6.239)	2.591	91(46+45)	3.00	3.02	3.00	+∞	3.01

minimize the weight. The RBDO formulation of the speed reducer can be expressed as

$$
\begin{aligned}
&\text{Find} d = \left[d_1, d_2, d_3, d_4, d_5, d_6, d_7\right]\\
&\min f(d) = 0.7854 d_1 d_2^2 \left(3.3333 d_3^2 + 14.9334 d_3 - 43.0934\right)\\
&\qquad - 1.508 d_1 \left(d_6^2 + d_7^2\right) + 7.477\left(d_6^3 + d_7^3\right)\\
&\qquad + 0.7854\left(d_4 d_6^2 + d_5 d_7^2\right)\\
&\text{s.t.}\begin{cases} P\left[G_j(x) \le 0 | d\right] \le \Phi\left(-\beta_j^t\right), \quad \beta_j^t = 3, \quad j = 1, 2, \ldots, 11\\ 2.6 \le d_1 \le 3.6, 0.7 \le d_2 \le 0.8, \quad 17 \le d_3 \le 28,\\ 7.3 \le d_4 \le 8.3, 7.3 \le d_5 \le 8.3, 2.9 \le d_6 \le 3.9,\\ 5.0 \le d_7 \le 5.5, \quad x_i \sim N\left(d_i, 0.005^2\right), \quad i = 1, 2, \ldots, 7 \end{cases}\\
&\text{where} \quad G_1(x) = 1 - \frac{27}{x_1 x_2^2 x_3}, \quad G_2(x) = 1 - \frac{397.5}{x_1 x_2^2 x_3^2},\\
&\qquad G_3(x) = 1 - \frac{1.93 x_4^3}{x_2 x_3 x_6^4}, \quad G_4(x) = 1 - \frac{1.93 x_5^3}{x_2 x_3 x_7^4},\\
&\qquad G_5(x) = 1 - \frac{\sqrt{\left[745 x_4 / \left(x_2 x_3\right)\right]^2 + 16.9 \times 10^6}}{110 x_6^3},\\
&\qquad G_6(x) = 1 - \frac{\sqrt{\left[745 x_5 / \left(x_2 x_3\right)\right]^2 + 157.5 \times 10^6}}{85 x_7^3},\\
&\qquad G_7(x) = 1 - x_2 x_3 / 40, \quad G_8(x) = x_1 / \left(5 x_2\right) - 1,\\
&\qquad G_9(x) = 1 - x_1 / \left(12 x_2\right), \quad G_{10}(x) = 1 - \frac{1.5 x_6 + 1.9}{x_4},\\
&\qquad G_{11}(x) = 1 - \frac{1.1 x_7 + 1.9}{x_5}
\end{aligned}
\tag{59}
$$

The optimization results are summarized in Table 6. The optimal results of all the approaches are almost identical to those in the Ref. [36]. The probabilistic constraints at the optimal solutions of these methods listed in Table 6 are evaluated by the Monte Carlo simulation (MCS) with ten million samples, respectively. It can be seen that almost all the probabilistic constraints satisfy the target reliability at the optimum of each method. Among them, the LAS, IBS, and the proposed method are more efficient with fewer than 100 in terms of the total number of constraint function calls and objective function calls. Although the SSRBO method is also efficient in the RBDO problem, the sixth probabilistic constraint at the optimal solution with it does not satisfy the target reliability. Therefore, the efficiency of the proposed method is verified.

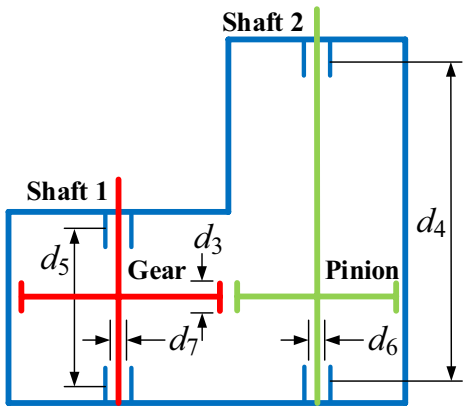

Fig. 8 The speed reducer for example 4

5 Conclusions

In this paper, an efficient method combining adaptive surrogate model and importance sampling-based modified SORA method is proposed for reliability-based design optimization.

Table 6 The reliability optimization results for example 4

Methods	Design variables	Objective	$N_{\text{total}}(N_{\text{con}}+N_{\text{obj}})$	β_5^{MCS}	β_6^{MCS}	β_8^{MCS}	β_{11}^{MCS}	$\beta_{\text{others}}^{\text{MCS}}$
RIA [36]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	4328	3.00	3.00	3.00	3.00	$+\infty$
PMA [36]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	1344	3.00	3.00	3.00	3.00	$+\infty$
SLA [36]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	736	3.00	3.00	3.00	3.00	$+\infty$
ALM [36]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	824	3.00	3.00	3.00	3.00	$+\infty$
AHA [36]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	480	3.00	3.00	3.00	3.00	$+\infty$
SLSV [33]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	1127(1078 + 49)	3.00	3.00	3.00	3.00	$+\infty$
ADA [33]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	581(541 + 40)	3.00	3.00	3.00	3.00	$+\infty$
SORA [30]	(3.580, 0.7, 17, 7.3, 7.764, 3.366, 5.302)	3040.02	521(500 + 21)	3.15	3.05	3.07	4.21	$+\infty$
ASORA [30]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	418(396 + 22)	3.18	3.08	3.07	4.21	$+\infty$
RMV [42]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.86	1472	–	–	–	–	–
HCC [42]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.86	1208	–	–	–	–	–
AC-SLA [42]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.85	484	–	–	–	–	–
HCMV [45]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.60	961	–	–	–	–	–
SQP–WSM [2]	(3.577, 0.7, 17, 7.3, 7.761, 3.365, 5.302)	3038.89	25,032	3.00	3.02	3.00	3.90	$+\infty$
PSO-WSM [2]	(3.576, 0.7, 17, 7.3, 7.760, 3.365, 5.302)	3038.80	27,250	3.01	3.01	3.00	3.77	$+\infty$
LAS [46]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.62	92(46 + 46)	2.98	3.01	2.99	2.99	$+\infty$
IBS [47]	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.66	100(50 + 50)	3.02	3.00	3.00	3.00	$+\infty$
SSRBO [26]	(3.576, 0.7, 17, 7.3, 7.754, 3.366, 5.301)	3038.53	78(39 + 39)	3.13	2.95	3.01	3.09	$+\infty$
The proposed method	(3.577, 0.7, 17, 7.3, 7.754, 3.365, 5.302)	3038.61	72(35 + 37)	3.00	3.01	3.01	3.00	$+\infty$

In the proposed method, this optimization process is cleverly divided into three phases: 1) constructing surrogate models of the constraint functions, 2) constructing surrogate model of the objective function and 3) solving RBDO by IS-based modified SORA. The proposed method is adopted to solve four classical RBDO examples, and its accuracy and efficiency are verified by comparisons with previous methods.

The accuracy and efficiency of the proposed method depend on the accuracy and efficiency in constructing surrogate models phase and the optimization solution phase. So, there are two main directions in our efforts for practical engineering in the future. The first is how to further improve the accuracy and efficiency of constructing surrogate models in a more efficient way or with a more efficient surrogate model, such as radial basis function (RBF) and Support Vector Machine (SVM). The second is to employ a more efficient analytical method in the third phase of the proposed method, such as the ASORA method, the AHA method, the AC-SLA method, etc.

Acknowledgements This study was supported in part by the Fundamental Research Funds for the Central Universities Grant No. 3102015 BJ (II) JL01 and the National Natural Science Foundation of China Grant No. 51675428.

References

1. Rashki Mohsen, Miri Mahmoud, Moghaddam Mehdi Azhdary (2014) A simulation-based method for reliability based design optimization problems with highly nonlinear constraints. Autom Constr 47:24–36
2. Hamzehkolaei Naser Safaeian, Miri Mahmoud, Rashki Mohsen (2016) An enhanced simulation-based design method coupled with meta-heuristic search algorithm for accurate reliability-based design optimization. Eng Comput 32:477–495
3. Du X, Chen W (2004) Sequential optimization and reliability assessment method for efficient probabilistic design. J Mech Des 126(2):225–233
4. Zhao J, Chan AHC, Roberts C, Madelin KB (2007) Reliability evaluation and optimisation of imperfect inspections for a component with multi-defects. Reliab Eng Syst Saf 92:65–73
5. Songqing Shan G, Wang Gary (2008) Reliable design space and complete single-loop reliability-based design optimization. Reliab Eng Syst Saf 93:1218–1230
6. Yang T, Hsieh YH (2013) Reliability-based design optimization with cooperation between support vector machine and particle swarm optimization. Eng Comput 29:151–163
7. Pradlwarter HJ, Schuëller GI (2010) Local domain Monte Carlo simulation. Struct Saf 32:275–280
8. Choi SK, Grandhi RV, Canfield RA (2007) Reliability-based structural design. Springer, London
9. Song Shufang Lu, Zhenzhou Qiao Hongwei (2009) Subset simulation for structural reliability sensitivity analysis. Reliab Eng Syst Saf 94:658–665
10. Au SK, Beck J (2001) Estimation of small failure probabilities in high dimensions by subset simulation. Probab Eng Mech 16(4):263–277
11. Angelis MD, Patelli E, Beer M (2015) Advanced line sampling for efficient robust reliability analysis. Struct Saf 52:170–182
12. Chen Chyi-Tsong, Chen Mu-Ho, Horng Wei-Tze (2014) A cell evolution method for reliability-based design optimization. Appl Soft Comput 15:67–79
13. Zhu Ping, Shi Lei, Yang Ren-Jye, Lin Shih-Po (2015) A new sampling-based RBDO method via score function with reweighting

scheme and application to vehicle designs. Appl Math Model 39:4243–4256
14. Rashki M, Miri M, Azhdary Moghaddam M (2012) A new efficient simulation method to approximate the probability of failure and most probable point. Struct Saf 39:22–29
15. Rashki M, Miri M, Azhdary Moghaddam M (2014) Closure to "A new efficient simulation method to approximate the probability of failure and most probable point". Struct Saf 46:15–16
16. Okasha Nader M (2016) An improved weighted average simulation approach for solving reliability-based analysis and design optimization problems. Struct Saf 60:47–55
17. Hamzehkolaei Naser Safaeian, Miri Mahmoud, Rashki Mohsen (2018) New simulation-based frameworks for multi-objective reliability-based design optimization of structures. Appl Math Model 62:1–20
18. Du X, Chen W (2001) A most probable point-based method for efficient uncertainty analysis. J Des Manuf Autom 4(1):47–66
19. Hasofer AM, Lind NC (1974) Exact and invariant second-moment code format. J Eng Mech Div 100(1):111–121
20. Lee I, Choi KK, Du L, Gorsich D (2008) Inverse analysis method using MPP-based dimension reduction for reliability-based design optimization of nonlinear and multi-dimensional systems. Comput Methods Appl Mech Eng 198(1):14–27
21. Grandhi RV, Wang L (1998) Reliability-based structural optimization using improved two-point adaptive nonlinear approximations. Finite Elem Anal Des 29:35–48
22. Tu J, Choi KK, Park YH (1999) A new study on reliability-based design optimization. J Mech Des 121:557–564
23. 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
24. Tsompanakis Y, Lagaros ND, Papadrakakis M (2010) Structural design optimization considering uncertainties. Taylor and Francis, London
25. Youn BD, Choi KK (2004) An investigation of nonlinearity of reliability-based design optimization approaches. J Mech Des 126(3):403–411
26. Li Xu, Gong Chunlin, Liangxian Gu, Jing Zhao, Fang Hai, Gao Ruichao (2019) A reliability-based optimization method using sequential surrogate model and Monte Carlo simulation. Struct Multidiscip Optim 59(2):439–460
27. Liang J, Mourelatos ZP, Nikolaidis E (2007) A single-loop approach for system reliability-based design optimization. J Mech Des 129(12):1215–1224
28. Cheng G, Xu L, Jiang L (2006) A sequential approximate programming strategy for reliability-based structural optimization. Comput Struct 84(21):1353–1367
29. Li Fan, Teresa Wu, Mengqi Hu, Dong Jin (2010) An accurate penalty-based approach for reliability-based design optimization. Res Eng Des 21:87–98
30. 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
31. Ho-Huu V, Nguyen-Thoi T, Le-Anh L, Nguyen-Trang T (2016) An effective reliability-based improved constrained differential evolution for reliability-based design optimization of truss structures. Adv Eng Softw 92:48–56
32. Jeong Seong-Beom, Park Gyung-Jin (2017) Single loop single vector approach using the conjugate gradient in reliability based design optimization. Struct Multidiscip Optim 55:1329–1344
33. Chen Zhenzhong, Qiu Haobo, Gao Liang, Liu Su, Li Peigen (2013) An adaptive decoupling approach for reliability-based design optimization. Comput Struct 117:58–66
34. Jiang Chen, Qiu Haobo, Gao Liang, Cai Xiwen, Li Peigen (2017) An adaptive hybrid single-loop method for reliability-based design optimization using iterative control strategy. Struct Multidiscip Optim 56:1271–1286
35. Chen Zhenzhong, Li Xiaoke, Chen Ge, Gao Liang, Qiu Haobo, Wang Shengze (2018) A probabilistic feasible region approach for reliability-based design optimization. Struct Multidiscip Optim 57:359–372
36. Li Gang, Meng Zeng, Hao Hu (2015) An adaptive hybrid approach for reliability-based design optimization. Struct Multidiscip Optim 51:1051–1065
37. Keshtegar Behrooz (2017) A modified mean value of performance measure approach for reliability-based design optimization. Arab J Sci Eng 42:1093–1101
38. Keshtegar Behrooz, Hao Peng, Meng Zeng (2017) A self-adaptive modified chaos control method for reliability-based design optimization. Struct Multidiscip Optim 55:63–75
39. Keshtegara Behrooz, Hao Peng (2018) A hybrid descent mean value for accurate and efficient performance measure approach of reliability-based design optimization. Comput Methods Appl Mech Eng 336:237–259
40. Keshtegara Behrooz, Hao Peng (2018) Enhanced single-loop method for efficient reliability-based design optimization with complex constraints. Struct Multidiscip Optim 57:1731–1747
41. Hao Peng, Wang Yutian, Ma Rui, Liu Hongliang, Wang Bo, Li Gang (2019) A new reliability-based design optimization framework using isogeometric analysis. Comput Methods Appl Mech Eng 345:476–501
42. Meng Zeng, Keshtegar Behrooz (2019) Adaptive conjugate single-loop method for efficient reliability-based design and topology optimization. Comput Methods Appl Mech Eng 344:95–119
43. Keshtegar Behrooz, Hao Peng (2018) Enriched self-adjusted performance measure approach for reliability-based design optimization of complex engineering problems. Appl Math Model 57:37–51
44. Hao Peng, Ma Rui, Wang Yutian, Feng Shaowei, Wang Bo, Li Gang, Xing Hanzheng, Yang Fan (2019) An augmented step size adjustment method for the performance measure approach: toward general structural reliability-based design optimization. Struct Saf 80:32–45
45. Zhu Shun-Peng, Keshtegar Behrooz, Trung Nguyen-Thoi, Yaseen Zaher Mundher, Bui Dieu Tien (2019) Reliability-based structural design optimization: hybridized conjugate mean value approach. Eng Comput. https://doi.org/10.1007/s00366-019-00829-7
46. Chen Zhenzhong, Qiu Haobo, Gao Liang, Li Xiaoke, Li Peigen (2014) A local adaptive sampling method for reliability-based design optimization using Kriging model. Struct Multidiscip Optim 49:401–416
47. Chen Zhenzhong, Peng Siping, Li Xiaoke, Qiu Haobo, Xiong Huadi, Gao Liang, Li Peigen (2015) An important boundary sampling method for reliability-based design optimization using kriging model. Struct Multidiscip Optim 52:55–70
48. Li Xiaoke, Qiu Haobo, Chen Zhenzhong, Gao Liang, Shao Xinyu (2016) A local kriging approximation method using MPP for reliability-based design optimization. Comput Struct 162:102–115
49. Shi Lei, Lin Shih-Po (2016) A new RBDO method using adaptive response surface and first-order score function for crashworthiness design. Reliab Eng Syst Saf 156:125–133
50. Lehký D, Slowik O, Novák D (2018) Reliability-based design: artificial neural networks and double-loop reliability-based optimization approaches. Adv Eng Softw 117:123–135
51. Echard B, Gayton N, Lemaire M (2011) AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. Struct Saf 33(2):145–154
52. Lophaven SN, Nielsen HB, Sondergaard J (2002) DACE, a matlab Kriging toolbox, version 2.0. Technical Report. IMM-TR-2002-12; Technical University of Denmark

53. Lv Z, Lu Z, Wang P (2015) A new learning function for Kriging and its applications to solve reliability problems in engineering. Comput Math Appl 70:1182–1197
54. Sun Zhili, Wang Jian, Li Rui, Tong Cao (2017) LIF: a new Kriging learning function and its application to structural reliability analysis. Reliab Eng Syst Saf 157:152–165
55. Kunling Song, Zhang Yugang Yu, Xinshui Song Bifeng (2019) A new sequential surrogate method for reliability analysis and its applications in engineering. IEEE Access 7:60555–60571
56. Papaioannou Iason, Betz Wolfgang, Zwirglmaier Kilian, Straub Daniel (2015) MCMC algorithms for Subset Simulation. Probab Eng Mech 41:89–103
57. Zhao Hailong, Yue Zhufeng, Liu Yongshou, Gao Zongzhan, Zhang Yishang (2015) An efficient reliability method combining adaptive importance sampling and Kriging metamodel. Appl Math Model 39:1853–1866
58. Kocis L, Whiten WJ (1997) Computational investigations of low-discrepancy sequences. ACM Trans Math Softw 23(2):266–294
59. Keshtegar Behrooz, Chakraborty Souvik (2018) Dynamical accelerated performance measure approach for efficient reliability-based design optimization with highly nonlinear probabilistic constraints. Reliab Eng Syst Saf 178:69–83
60. Keshtegar Behrooz, Baharom Shahrizan, EI-Shafie Ahmed (2018) Self-adaptive conjugate method for a robust and efficient performance measure approach for reliability-based design optimization. Eng Comput 34:187–202