Constrained Bayesian Optimization Algorithms for Estimating Design Points in Structural Reliability Analysis

Jingwen Song[a,*], Yifan Cui[b], Pengfei Wei[c], Marcos A. Valdebenito[d], Weihong Zhang[a]

[a]*School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China*
[b]*School of Mechanics, Civil Engineering and Architecture, Northwestern Polytechnical University, Xi'an 710072, China*
[c]*School of Power and Energy, Northwestern Polytechnical University, Xi'an 710072, China*
[d]*Chair for Reliability Engineering, TU Dortmund University, Leonhard-Euler Strasse 5, 44227, Dortmund, Germany*

Abstract

Estimating the design points with high accuracy is a historical and key issue for many reliability analysis and reliability-based design optimization methods. Indeed, it is still a challenge especially when the limit state functions (LSFs) show highly nonlinear behaviors, and/or the reliability index is large, and/or the gradients of LSF are not available. To fill the above gap, two acquisition functions incorporating both the objective function and constraints are devised, and based on which, a Constrained Bayesian Optimization (ConBayOpt) method is firstly developed for actively learning the design points with high accuracy and global convergence. Further, an improved algorithm, called Constrained Bayesian Subset Optimization (ConBaySubOpt) is devised for adaptively learning the design points far away from the origin of the standard normal space. Similar to subset simulation, the ConBaySubOpt algorithm automatically produces a set of intermediate failure surfaces and feasible regions for approaching the true design point, but does not require Markov Chain Monte Carlo simulation for conditional sampling. The efficiency, accuracy and wide applicability of the proposed methods are demonstrated with two test examples and three engineering examples.

Keywords: Bayesian Optimization; Design Point; Acquisition Function; Gaussian Process Regression; Feasible Regions

1. Introduction

Reliability analysis and Reliability-Based Design Optimization (RBDO) under the probabilistic framework have been recognized as the key for designing engineering structures with high reliability under uncertainties, and many methods have been developed for these purposes. For reliability analysis, the numerical methods include approximate analytical methods (e.g., first-order reliability method, FORM, and second-order reliability method, SORM) [1, 2], response surface methods [3], advanced Monte Carlo simulation (MCS) methods (e.g., importance sampling [4, 5], subset simulation and its variants [6, 7, 8], as well as line sampling [9, 10]), active learning combining (advanced) MCS and surrogate models [11, 12, 13, 14, 15, 16],

*Corresponding author at School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China
Email address: jingwensong@nwpu.edu.cn (Jingwen Song)

and probability conservation based methods [17, 18]. Some of these methods have been further developed for solving the RBDO problem (e.g., Ref. [2, 19, 20]), under double-loop or decoupling frameworks. Despite the above developments, devising methods with higher performance for all kinds of structural reliability problems is an ongoing issue.

The design point has played a key role in structural reliability analysis. It is defined as the point in the failure domain having the highest probability density value, and the region around a design point usually contributes the most to the probability of failure in problems involving a low or middle number of dimensions [21, 22, 23]. It is the above reason that makes the design point a very important concept in structural reliability theory, and many of the above-mentioned methods are devised based on accurate estimation of the design point. For example, both the SORM and FORM methods are based on Taylor series expansion at the design point (see e.g., Ref. [24]); the important sampling density can be designed by moving the center of density to the design point (see e.g., Refs. [25, 26]); line sampling is based on generating a set of line samples along the important direction, identified by the design point (see, e.g., Ref. [27]); and the response surface methods are based on actively updating the surrogate model for approximating the limit state function around the design point with acceptable accuracy. The concept of design point has also seen a wide application in many RBDO methods, for example, an equivalent design point concept is introduced in Ref. [28] for transforming the time-variant RBDO problem into an equivalent time-invariant one; the design point is combined with Kriging in Refs. [29, 30] for solving the RBDO problem, etc. Besides, the concept of design point has also been extended for reliability analysis and RBDO under mixed uncertainties (see, e.g., Ref. [31, 32]) and quantification of uncertainties in structural prognostics [33]. In summary, the design point is a very important concept in structural reliability engineering, and development of effective methods for estimating it with wide applicability and high efficiency is then a timely issue.

From a mathematical perspective, estimating the design point is a typical constrained optimization problem. In the past half century, many methods, mostly based on gradients, have been developed. The first milestone on this direction is Ref. [34], where a line search scheme is designed for approaching the design point based on gradients in the standard Gaussian space. This method is later extended to accommodate the non-Gaussian variables in Ref. [35], and the developed method is well-known as "Hasofer-Lind-Rackwitz-Flessler (HLRF)". Till now, this strategy has been widely used in many areas, such as for solving RBDO [36] and slope reliability analysis [37]. Some other improvements, such as the utilization of a quadratic searching strategy [38], has also been developed following this line, but all these methods are gradient-based. A gradient-free method has also been developed based on interpolation and adaptive response surface in Ref. [39], and received much attention in both engineering and academic communities. More recently, a stochastic simulation method based on uniform sampling and taking the probability density value as weight has been proposed for estimating the probability of failure and the design point [40], but it still requires several hundreds or even several thousands function calls for convergence, and is thus not applicable for expensive-

to-evaluate simulators. A probabilistic model based on cumulative distribution function has been established, and combined with sequential quadrature optimization for estimating the design point in the original space of input variables [41], but the performance relies on the utilized optimization algorithm. Besides, some methods based on meta-heuristic global optimization algorithms, e.g., Ref. [42], have also been introduced for calculating the design point, but these methods commonly require much higher number of function calls, thus are time-consuming. Overall, estimating the design point is still a challenge for problems, where the limit state function is highly nonlinear and multi-modal, and/or the gradient information is not available, and or the design point is far away from the distribution center of the inputs variables, and developing effective methods for filling this gap is an urgent demand [43].

Bayesian numerical analysis is a cutting-edge branch of engineering computation, which aims at formulating a numerical analysis problem as a statistic inference problem and then solved with Bayesian inference [44]. Till now, Bayesian numerical methods have been developed for most numerical tasks such as optimization [45], reliability analysis [46, 47], and Ordinary Differential Equation (ODE) solution [48]. Bayesian optimization is a set of gradient-free methods aiming at solving optimization problems with expensive-to-evaluate objective functions with global convergence [45, 49, 50]. Due to these appealing features, two Bayesian optimization procedures are developed for efficiently estimating the design point. Two acquisition functions adapted from the classical Expected Improvement (EI) are firstly devised, with the incorporation of both the objective and constraints, for transforming the constrained optimization problem into a non-constraint one. With either of these two acquisition functions as an engine, the first Bayesian optimization algorithm is developed for estimating the design point in those cases where this point is located relatively close to the origin of the standard normal space. Further, a more elaborated Bayesian algorithm is devised for searching the design point when it lies far away from the origin of the standard normal space. This algorithm is based on automatically introducing several intermediate failure surfaces and feasible regions for approaching the true design point in an adaptive way. The proposed methods are of wide applicability for highly nonlinear problems with very small failure probability, and are gradient-free.

The remaining of this work is organized as follows. The problem statement as well as the review of the Gaussian Process Regression (GPR) utilized for Bayesian optimization are presented in section 2, followed by the two developed Bayesian optimization algorithms for searching design point in section 3. Several benchmark examples are studied in section 4 for illustrating and demonstrating the proposed methods. Section 5 gives conclusions.

2. Research Materials

2.1. Problem Statement

The structural reliability problem is commonly formulated by a limit state function (LSF) $y = g(\boldsymbol{x})$ and a random input vector $\boldsymbol{x}$ of dimension n, with $y < 0$ indicating failure, and the randomness of $\boldsymbol{x}$ being described by the joint probability density function (PDF) $\phi_n(\boldsymbol{x})$. For simplicity, the input variables are assumed to follow independent standard Gaussian distribution throughout this work, and for the dependent and/or non-Gaussian cases, the Nataf transformation or Rosenblatt transformation can be applied to transform the input variables into independent standard Gaussian variables, and one can refer to Ref. [51] for details. With the above assumption, the failure domain is defined as $F = \{\boldsymbol{x} : g(\boldsymbol{x}) < 0\}$, and the its indicator function is denoted as $I_F(\boldsymbol{x})$, which equals to one if $\boldsymbol{x} \in F$ and otherwise zero. Given the above definition, the probability of failure p_f is defined as:

$$p_f = \int_F \phi_n(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \int_{\mathbb{R}^n} I_F(\boldsymbol{x})\, \phi_n(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. \tag{1}$$

Evaluating the probability of failure is a challenging task especially when the LSF is non-linear and the value of the probability of failure is very small (typically less than 10^{-3}). In the past half century, many methods, such as FORM, SORM and IS, have been developed for addressing the above challenging problem based on the so-called design point. Mathematically, the design point is defined as the point in the failure domain F which has the highest PDF value, and in the case of independent standard Gaussian inputs, the design point is indeed the point on the failure boundary $g(\boldsymbol{x}) = 0$ with the highest PDF value, i.e.,

$$\boldsymbol{x}_{\mathrm{D}} = \arg \max_{g(\boldsymbol{x})=0} \phi_n(\boldsymbol{x}). \tag{2}$$

From a geometrical perspective, the design point can also be formulated as the point in the failure boundary with the smallest distance to the origin, i.e.,

$$\boldsymbol{x}_{\mathrm{D}} = \arg \max_{g(\boldsymbol{x})=0} \frac{1}{\|\boldsymbol{x}\|} \tag{3}$$

, where $\|\cdot\|$ refers to the Euclidian distance. One notes that this distance is commonly called reliability index, and is denoted as $\beta = \|\boldsymbol{x}\|$, thus the cost function can also be defined as $-\|\boldsymbol{x}\|$. It has been noticed that utilizing whether Eq. (2) or Eq. (3) as the cost function makes no big difference on the training process or performance of the algorithms, but may affect the pre-specification of the algorithm parameters. For simplicity, Eq. (3) is adopted in this work. One notes the formulation in Eq. (3) is valid as long as the value of failure probability is not too large, for example $p_f \leq 0.1$.

The available methods developed for addressing the above optimization problem are mostly gradient-

based, and less effective when the optimization problem is multi-modal and/or the gradients are not available or are expensive to evaluate. The above limitations make the design point based reliability analysis methods not applicable in many engineering applications. Besides, for LSF with multi-modal behavior, the above-mentioned methods may lack global convergence. This motivates us to develop a gradient-free and efficient global optimization algorithm for addressing the above limitations. The developed methods are based on Bayesian optimization, where the LSF is approximated by a Gaussian Process Regression (GPR) model, which is briefly reviewed in the following subsection.

2.2. Gaussian Process Regression

Following the prior assumption of a Bayesian inference procedure, without knowing any information of the deterministic LSF $g(\boldsymbol{x})$, it is assumed to follow a Gaussian process $\hat{g}(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}'))$, where $m(\boldsymbol{x})$ indicates the prior mean function which can be assumed to be of any appropriate form such as zero, constant and polynomials, and $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ refers to the prior covariance function quantifying the correlation of $\hat{g}(\boldsymbol{x})$ at any two different sites $\boldsymbol{x}$ and $\boldsymbol{x}'$. Many kinds of kernel functions, such as polynomial kernel, squared exponential kernel and Matérn kernel, can be utilized, but the optimal kernel function for a specific problem depends on the behavior of the LSF. One can refer to Ref. [52] for guidance on kernel selection. Given a set of training data $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$, where $\mathcal{X}$ is a $(N \times n)$-dimensional sample matrix with the i-th row being the i-th sample of $\boldsymbol{x}$, and $\mathcal{Y}$ refers to a N-dimensional column vector with the i-th element being $y^{(i)} = g(\boldsymbol{x}^{(i)})$, it is known that vector $\mathcal{Y}$ follows a N-dimensional Gaussian distribution with mean vector $\boldsymbol{m}(\mathcal{X}) = \left(m\left(\boldsymbol{x}^{(1)}\right), m\left(\boldsymbol{x}^{(2)}\right), \cdots m\left(\boldsymbol{x}^{(N)}\right)\right)^{\top}$ and covariance matrix $\mathcal{K}$ whose (i,j)-th element is $\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right)$. The likelihood function can then be formulated as the joint PDF of this Gaussian vector, and the hyper-parameters involved in $m(\boldsymbol{x})$ and $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ can be computed by maximum likelihood estimation.

With the above concept, a posterior GPR model can be inferred by conditioning on the observed data $\mathcal{D}$ as $\hat{g}_{\mathcal{D}}(\boldsymbol{x}) \sim \mathcal{GP}(\mu_g(\boldsymbol{x}), c_g(\boldsymbol{x}, \boldsymbol{x}'))$, where the posterior mean function $\mu_g(\boldsymbol{x})$ and the posterior covariance function $c_g(\boldsymbol{x}, \boldsymbol{x}')$ are formulated as:

$$\mu_g(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \boldsymbol{\kappa}(\mathcal{X}, \boldsymbol{x})^{\top} \mathcal{K}^{-1}(\mathcal{Y} - \mu(\mathcal{X})) \tag{4a}$$

$$c_g(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\kappa}(\mathcal{X}, \boldsymbol{x})^{\top} \mathcal{K}^{-1} \boldsymbol{\kappa}(\mathcal{X}, \boldsymbol{x}) \tag{4b}$$

, where $\boldsymbol{\kappa}(\mathcal{X}, \boldsymbol{x})$ refers to a N-dimensional column vector with the i-th element being $\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}\right)$. The posterior variance is computed as $\sigma_g^2(\boldsymbol{x}) = c_g(\boldsymbol{x}, \boldsymbol{x})$. Given an unobserved point $\boldsymbol{x}$, although its LSF value $g(\boldsymbol{x})$ can not be precisely known, the Gaussian distribution with mean $\mu_g(\boldsymbol{x})$ and $\sigma_g^2(\boldsymbol{x})$ can be used for summarizing the available knowledge on this deterministic value, and provides fruitful information for active learning of the MPP, as discussed in the remaining parts of this work. It needs to be clarified that both the prior and posterior Gaussian assumption is made on the value of the LSF $g(\boldsymbol{x})$, instead of the input vector

$\boldsymbol{x}$, thus the GPR model does not restricts the distribution type associated with $\boldsymbol{x}$.

3. The Proposed Methods

In this section, two gradient-free Bayesian optimization algorithms will be developed for computing the design point near the origin and those far away from the origin, respectively. The performance of a Bayesian optimization algorithm is determined by the utilized acquisition function to a large extent [49], thus devising new acquisition functions has long been a focus. For an optimization algorithm, where "exploration" refers to the ability of exploring new regions and thus it serves to escape from a local optimum, whereas "exploitation" indicates the ability of searching local optimum in the neighborhood of the current solution, thus determines the convergence rate once the vicinity of the true optimum is found. Thus, a good acquisition function should have a balance between exploration and exploitation at different stages of searching. Many acquisition functions, such as the probability of improvement function [53], the Expected Improvement (EI) function and its variants [54, 55], the knowledge-gradient function [56], the entropy search function [57] and the predictive entropy search function [58], have been developed, where the EI functions are the most popular ones as they commonly admit closed-form expressions, and provide reasonable balance between exploration and exploitation. Motivated by this fact, the idea of EI function is utilized in this section to devise new acquisition functions for searching the design points.

Most developments in Bayesian optimization are applicable to problems with rectangular feasible regions and without constraints. There are commonly two strategies for extending those methods to problems with various types of constraints, where the first is to develop an acquisition function with the incorporation of the constraints. Such an approach has been termed as constrained Bayesian optimization (ConBayOpt) (see e.g., Ref. [59, 60] for examples). The second strategy is to transform the constrained problems into unconstrained ones using the augmented Lagrangian relaxation [61, 62]. In this section, the first strategy is adopted as the second commonly requires coping with non-stationary GPR models [63].

3.1. Constrained Bayesian Optimization for design point near origin

Let ϵ denote a small positive value. For incorporating the equality constraint $g(\boldsymbol{x})$, the following deterministic function is defined:

$$h\left(\boldsymbol{x}\right) = \frac{1}{\left\|\boldsymbol{x}\right\|} I\left[\left|g\left(\boldsymbol{x}\right)\right| \leqslant \epsilon\right] \tag{5}$$

, where $I\left[\cdot\right]$ denotes the indicator function which equals to one if the argument holds and, otherwise, zero. The above h-function is interpreted as follows. The point $\boldsymbol{x}$ with the maximum h-function value is the one which is located in the narrow strip $-\epsilon \leqslant g\left(\boldsymbol{x}\right) \leqslant \epsilon$ (i.e., on the failure surface), and meanwhile the nearest to the origin. This point would indeed tend to the design point as ϵ approaches zero. Therefore,

with $h(\boldsymbol{x})$, the constrained optimization problem in Eq.(3) for estimating the design point is reformulated as the following unconstrained optimization problem:

$$\boldsymbol{x}_{\mathrm{D}} = \arg\max_{\boldsymbol{x}\in\mathbb{R}^n} h\left(\boldsymbol{x}\right) \tag{6}$$

, which can then be solved with a Bayesian optimization procedure. However, Bayesian optimization cannot be directly applied as modeling the non-negative and non-smooth objective function $h(\boldsymbol{x})$ with a GPR model may result in a poor performance on convergence.

We develop a more reasonable scheme to fill the above gap. Assume now a GPR model $\hat{g}_{\mathcal{D}}\left(\boldsymbol{x}\right)$ is trained for approximating the LSF $g\left(\boldsymbol{x}\right)$ with an initial training data set $\mathcal{D}$. It is then clear that $\hat{h}_{\mathcal{D}}(\boldsymbol{x})$, defined by replacing $g(\boldsymbol{x})$ in Eq. (5) with $\hat{g}_{\mathcal{D}}(\boldsymbol{x})$, is a random process having two possible outcomes at each site of $\boldsymbol{x}$. It takes a value of either $1/\|\boldsymbol{x}\|$ with probability $p(\boldsymbol{x})$ or 0 with probability $1-p(\boldsymbol{x})$, i.e.,

$$\begin{cases} \Pr\left(\hat{h}_{\mathcal{D}}\left(\boldsymbol{x}\right)=\frac{1}{\|\boldsymbol{x}\|}\right)=p\left(\boldsymbol{x}\right) \\ \Pr\left(\hat{h}_{\mathcal{D}}\left(\boldsymbol{x}\right)=0\right)=1-p\left(\boldsymbol{x}\right) \end{cases} \tag{7}$$

, where $\Pr\left(\cdot\right)$ refers to the probability measure, $p\left(\boldsymbol{x}\right)=\Phi\left(\frac{\epsilon-\mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right)-\Phi\left(\frac{-\epsilon-\mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right)$, and $\Phi\left(\cdot\right)$ indicates the Cumulative Distribution Function (CDF) of the univariate standard normal distribution. The posterior mean $\mu_h\left(\boldsymbol{x}\right)$ of $\hat{h}_{\mathcal{D}}(\boldsymbol{x})$ is then derived as:

$$\mu_h\left(\boldsymbol{x}\right)=\frac{1}{\|\boldsymbol{x}\|}\mathbb{E}_{\mathcal{D}}\left[I\left[\left|\hat{g}_{\mathcal{D}}\left(\boldsymbol{x}\right)\right|\leqslant\epsilon\right]\right]=\frac{1}{\|\boldsymbol{x}\|}p\left(\boldsymbol{x}\right) \tag{8}$$

, where $\mathbb{E}_{\mathcal{D}}\left[\cdot\right]$ indicates the expectation operator taken over the GPR model $\hat{g}_{\mathcal{D}}\left(\boldsymbol{x}\right)$. A reference point can be computed by solving:

$$\boldsymbol{x}^*=\arg\max_{\boldsymbol{x}\in\mathbb{R}^n}\mu_h\left(\boldsymbol{x}\right). \tag{9}$$

As $\mu_h\left(\boldsymbol{x}\right)$ admits a closed-form expression, Eq. (9) can be efficiently solved by evolutionary optimization algorithms, and in this work, the Particle Swarm Optimization (PSO) algorithm is suggested.

After solving Eq. (9), the next step is to decide whether the GPR model should be improved with an additional evaluation of the LSF. For this purpose, it is necessary to define an acquisition function following the EI based Bayesian optimization procedure [45]. The first EI function is defined as:

$$\mathcal{A}^{\mathrm{EI1}}\left(\boldsymbol{x}\right)=\mathbb{E}_{\mathcal{D}}\left[\max\left(\hat{h}_{\mathcal{D}}\left(\boldsymbol{x}\right)-\mu_h\left(\boldsymbol{x}^*\right),0\right)\right]. \tag{10}$$

The value of $\mathcal{A}^{\mathrm{EI1}}\left(\boldsymbol{x}\right)$ quantifies the expected increment of $h(\boldsymbol{x})$ at the point $\boldsymbol{x}$ compared to that at the current best guess $\boldsymbol{x}^*$. By observing and adding the point with highest EI value to the training data to

update the GPR model, it is expected to improve the knowledge on the global maximum point to the greatest extent. Based on the explicit distribution of $\hat{h}_{\mathcal{D}}(\boldsymbol{x})$ given in Eq. (7), the closed-form expression of $\mathcal{A}^{\mathrm{EI}}(\boldsymbol{x})$ can be derived as:

$$\mathcal{A}^{\mathrm{EI1}}(\boldsymbol{x}) = p(\boldsymbol{x}) \max\left(\frac{1}{\|\boldsymbol{x}\|} - \frac{1}{\|\boldsymbol{x}^*\|} p(\boldsymbol{x}^*), 0\right). \tag{11}$$

The EI function defined in Eq. (10) evaluates the expected improvement with the posterior mean prediction at the reference point $\boldsymbol{x}^*$ as basis, and thus the prediction uncertainty at the reference point is omitted in the acquisition. To alleviate this, the second EI function is defined as:

$$\mathcal{A}^{\mathrm{EI2}}(\boldsymbol{x}) = \mathbb{E}_{\mathcal{D}}\left[\max\left(\hat{h}_{\mathcal{D}}(\boldsymbol{x}) - \hat{h}_{\mathcal{D}}(\boldsymbol{x}^*), 0\right)\right] \tag{12}$$

, of which the closed-form expression is derived based on the joint probability distribution of $\hat{h}_{\mathcal{D}}(\boldsymbol{x})$ and $\hat{h}_{\mathcal{D}}(\boldsymbol{x}^*)$ as:

$$\mathcal{A}^{\mathrm{EI2}}(\boldsymbol{x}) = \frac{1}{\|\boldsymbol{x}\|} p(\boldsymbol{x})(1 - p(\boldsymbol{x}^*)) + \max\left(\frac{1}{\|\boldsymbol{x}\|} - \frac{1}{\|\boldsymbol{x}^*\|}, 0\right) p(\boldsymbol{x}) p(\boldsymbol{x}^*). \tag{13}$$

The second EI function can then be interpreted as the expected improvement of objective function value at $\boldsymbol{x}$ compared to that at $\boldsymbol{x}^*$ with the consideration of prediction uncertainties at both $\boldsymbol{x}$ and $\boldsymbol{x}^*$. As both EI functions have closed-form expression and it is efficient to compute them, any evolutionary optimization algorithm can be used for identifying its maximum, and still, the PSO algorithm is suggested.

For illustrating the two EI functions, we train a GPR model, and the two EI functions introduced in Eqs. (11) and (13) are derived, with their heat maps shown in Figure 1, together with the true MPP as well as the true failure surface and the failure surface specified with the posterior mean of the GPR model. It is seen that, for this example, the two EI functions show very similar but not exactly the same feature. It seems that difference only exists around the intersection point of the failure boundary identified by $\mu_g(\boldsymbol{x}) = 0$ and the true failure boundary $g(\boldsymbol{x}) = 0$. The small difference is mainly caused by the small prediction uncertainty at the reference point. Indeed, the most notable difference between the two EI functions is that the second one does not allow specifying the reference point as the next training point, as the value of it at the reference point equals to zero. It is seen that the maximum points of both EI functions are located between the true design point and the reference design point specified by the posterior mean, indicating a correct pointing toward the true design point.

The stopping condition is defined as that the normalized expected improvement at a specific step is less than a small tolerance Δ^{EI}, i.e.,

$$\mathcal{A}^{\mathrm{EI}}(\boldsymbol{x}^+) < \Delta^{\mathrm{EI}} \tag{14}$$

, where $\boldsymbol{x}^+ = \arg\max_{\boldsymbol{x}\in\mathbb{R}^n} \mathcal{A}^{\mathrm{EI}}(\boldsymbol{x})$, and $\mathcal{A}^{\mathrm{EI}}(\boldsymbol{x})$ refers to the EI function defined by either Eqs. (11) or

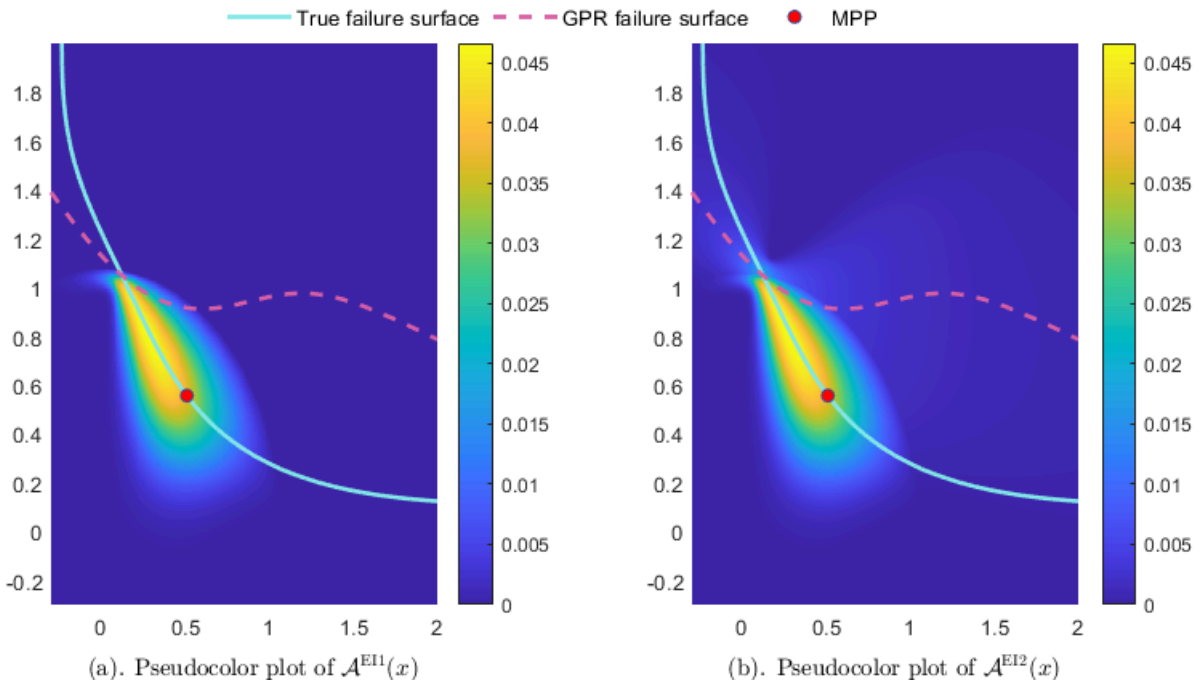


Figure 1: Pseudocolor plot of the acquisition function $\mathcal{A}^{\mathrm{EI}}(\boldsymbol{x})$ defined by Eq. (10) for a two-dimensional example, together with the design point, and the failure surfaces defined by $g(\boldsymbol{x}) = 0$ and $\mu_g(\boldsymbol{x}) = 0$.

(13).

With the above developments ready, the pseudo-code of the ConBayOpt algorithm for solving the design point is then summarized in Algorithm 1. Three algorithm parameters, i.e., the initial training sample size N_0, the error tolerance ϵ used in the EI acquisition function and the stopping threshold Δ^{EI}, needs to be specified. Based on experience, N_0 can be set to be a value higher than $n+1$, and for low-dimensional problems, at least 10 initial training points are required. Generally, different sampling schemes, such as Latin-hypercube sampling, Sobol' low-discrepancy sequence and the Halton sequence, can be used for producing the initial training points. One notes that the feasible region does not necessarily need to be properly covered by the initial training points, but initial training points with better space-filling is definitely helpful. The value of ϵ is based on the variation of the response value. The smaller ϵ is, the closer is the resultant design point to the true failure boundary, but it also results in non-smoothness of the acquisition function, making it challenging to compute the next design point, and if applicable, requires higher number of LSF calls for convergence. The value of the stopping threshold Δ^{EI} should be determined based on the distance of the design point to the origin, a larger distance requires a smaller value of Δ^{EI}. For the same problem, smaller value of Δ^{EI} results in higher number of simulator calls, but also produces results with higher accuracy. A trade-off between accuracy and efficiency should be made. Generally, it can be set to be a value

between $10^{-5} \sim 10^{-8}$. It is also noted that, during the early training stage when the number of training points is small, the posterior variance of the GPR model may not be adequate for evaluating the prediction uncertainty, and thus fake convergence may happen. To guarantee the robustness of the algorithm, a delayed judging strategy can be utilized, which means to end the algorithm only when the stopping condition is satisfied for twice in succession. This delayed judging strategy is found to be effective for avoiding the fake convergence, benefiting from the fact the posterior features of the GPR model can be adjusted to a great extent when a new (optimal) training point is added, especially at the early training stage where the GPR model conveys large prediction uncertainty.

Algorithm 1 is recommended when the distance between the design point and the origin is not large (typically less than 2.5), where design point must happen within the feasible region $x_i \in [-3, 3]$. Thus, the two optimization procedures in Algorithm 1 can be solved within this feasible region. For design point far away from the origin, this algorithm is less effective especially when the LSF is highly nonlinear, making it difficult for the GPR model to reach the true failure boundary. For this challenging case, a subset optimization procedure will be devised in the following subsection.

Algorithm 1: Constrained Bayesian Optimization (ConBayOpt) for design point near origin

Input: N_0, ϵ, Δ^{EI}, $g(\boldsymbol{x})$.

Output: $\boldsymbol{x}_{\mathrm{D}}$, $\mathcal{D}$, $\hat{g}_{\mathcal{D}}(\boldsymbol{x})$, N_{call}

StopFlag = False;

Generate $\mathcal{X}$ of size N_0 by e.g., Latin-hypercube sampling (LHS), and compute the corresponding LSF values $\mathcal{Y}$ to produce the initial training data set $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$, let $N_{\mathrm{call}} = N_0$ and $i = 0$;

while StopFlag=False **do**

  Let $i = i + 1$, train a GPR model $\hat{g}_{\mathcal{D}}(\boldsymbol{x})$ based on $\mathcal{D}$;

  Compute the i-th reference point $\boldsymbol{x}^{*(i)}$ by solving Eq. (9) with e.g., the PSO algorithm;

  Compute $\boldsymbol{x}^{+}$ by maximizing $\mathcal{A}^{\mathrm{EI}}(\boldsymbol{x})$;

  if *Eq. (14) holds* **then**

    Let $\boldsymbol{x}_{\mathrm{D}} = \boldsymbol{x}^{*(i)}$, and StopFlag=True;

  else

    Compute $y^{+} = g(\boldsymbol{x}^{+})$, add this point to $\mathcal{D}$, and let $N_{\mathrm{call}} = N_{\mathrm{call}} + 1$;

  end

end

3.2. Constrained Bayesian Subset Optimization for distant design point

The basic rationale of the Constrained Bayesian Subset Optimization (ConBaySubOpt) procedure for estimating the distant design point is schematically shown in Figure 2. The idea is borrowed from the subset simulation originally developed for estimating small failure probability [64], but our method avoids

sampling from the conditional density, which is commonly challenging and accomplished with Markov Chain Monte Carlo (MCMC). Instead of directly searching the distant design point $\boldsymbol{x}_{\mathrm{D}}$, a set of intermediate failure boundaries are introduced with both the intermediate values b_l and the corresponding intermediate design points $\boldsymbol{x}_{\mathrm{D}}^{(l)}$ actively learned. Denote the intermediate failure domains as $F_l = \{\boldsymbol{x} : g(\boldsymbol{x}) < b_l\}$, where $l = 1, 2, \cdots, m$ and $b_1 > b_2 > \cdots > b_m = 0$, it is then concluded $F_1 \supset F_2 \supset \cdots \supset F_m$.

The method starts with actively learning the value b_1 and the corresponding design point $\boldsymbol{x}_{\mathrm{D}}^{(1)}$ within the first feasible region centered at the origin (denoted as $\boldsymbol{x}_{\mathrm{D}}^{(0)}$) and bounded with a hyper-rectangle of length 2ξ, where b_1 is actively specified based on the principle that the distance β_1 of $\boldsymbol{x}_{\mathrm{D}}^{(1)}$ to $\boldsymbol{x}_{\mathrm{D}}^{(0)}$ is smaller than a pre-specified value. With the above setting, the design point of the first failure boundary must be located within the first feasible region, and thus can be learned with Algorithm 1. After the first design point $\boldsymbol{x}_{\mathrm{D}}^{(1)}$ is readily learned with pre-specified error tolerance, the second feasible region is built by moving the center of the first feasible region to $\boldsymbol{x}_{\mathrm{D}}^{(1)}$. The value of b_2 and the design point $\boldsymbol{x}_{\mathrm{D}}^{(2)}$ are then adaptively learned within the second feasible boundary. As $F_2 \subset F_1$ and b_2 is specified such that the distance β_{12} of $\boldsymbol{x}_{\mathrm{D}}^{(2)}$ to $\boldsymbol{x}_{\mathrm{D}}^{(1)}$ does not exceed ξ, the second design point should exist within the second feasible region, and can also be actively learned with Algorithm 1. The above process is repeated until that for one intermediate boundary, say the third one as shown in Figure 2, the value of b_m is smaller than zero, and is set to be zero. Within the corresponding feasible region centered at $\boldsymbol{x}_{\mathrm{D}}^{(m-1)}$, the true design point $\boldsymbol{x}_{\mathrm{D}}^{(m)}$ is actively learned with Algorithm 1, and its distance to the origin is computed as β, as shown in Figure 2.

Compared with Algorithm 1, two issues need to be specifically treated for ConBaySubOpt, i.e., the specification of the failure threshold value b_l and the acquisition function for each intermediate failure boundary. The value of b_l is specified with MCS. To specify the value of b_l, a set of N samples of $\boldsymbol{x}$ need to be generated following the independent Gaussian distribution with unit variance and centered at $\boldsymbol{x}_{\mathrm{D}}^{(l-1)}$. The posterior mean predictions at these N points are then computed, and the resultant values are then sorted in descending order. The value of b_l is then specified as the $\lceil p_0 N \rceil$-th prediction value to ensure that $100(1-p_0)$ percent samples belong to the failure domain defined by $\mu_g(\boldsymbol{x}) \leqslant b_l$. The value of b_l is also actively updated following the update of the GPR model, and as long as it is specified as a value smaller than zero, it is set as zero.

For learning the l-th design point $\boldsymbol{x}_{\mathrm{D}}^{(l)}$, the EI acquisition function defined by Eq. (10) and Eq. (12) can still be used by modifying the failure threshold from 0 as b_l. Then the $\boldsymbol{x}_{\mathrm{D}}^{(l)}$ is searched within the l-th feasible region. However, an important information, i.e., $\boldsymbol{x}_{\mathrm{D}}^{(l)} \in F_{l-1}$, has been omitted. Incorporating this constraints in the acquisition function is expected to further improve the robustness and reduce the required number of LSF calls. Motivated by the above fact, the following stochastic process model is introduced:

$$\hat{h}_{\mathcal{D}}^{(l)}(\boldsymbol{x}) = \frac{1}{\|\boldsymbol{x}\|} I\left[|\hat{g}_{\mathcal{D}}(\boldsymbol{x}) - b_l| \leqslant \epsilon \cap \hat{g}_{\mathcal{D}}(\boldsymbol{x}) \leqslant \alpha\sigma_g(\boldsymbol{x}) + b_{l-1}\right] \tag{15}$$

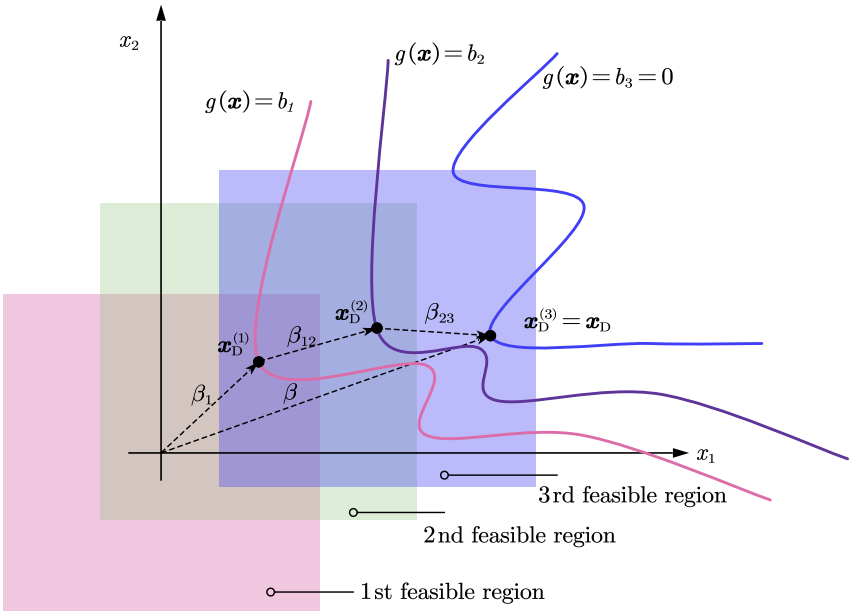


Figure 2: Schematic illustration of the ConBaySubOpt for inferring the distant design point.

, where the term $\alpha\sigma_g(\boldsymbol{x})$ is introduced as a soft bound making the constraint $\boldsymbol{x} \in F_{l-1}$ soft. This additional term provides the acquisition function with sufficient flexibility for balancing exploration and exploitation. Specifically, higher value of α refers to more inclination to exploration, which helps to exploring new regions. However, overemphasizing exploration performance will slow down the convergence rate once the globally optimal region has been identified. Thus, a trade-off needs to be made between exploration and exploitation by specifying the value of α. Based on experience, α can be set as a value between 0.5 and 1. The posterior mean of the $\hat{h}_{\mathcal{D}}^{(l)}(\boldsymbol{x})$ is then derived as:

$$\mu_h^{(l)}(\boldsymbol{x}) = \frac{1}{\|\boldsymbol{x}\|} p_l(\boldsymbol{x}) \tag{16}$$

, where the probability bound function $p_l(\boldsymbol{x})$ is formulated as:

$$p_l(\boldsymbol{x}) = \Phi\left(\frac{\min\left(\epsilon + b_l, \alpha\sigma_g(\boldsymbol{x}) + b_{l-1}\right) - \mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right) - \Phi\left(\frac{b_l - \epsilon - \mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right). \tag{17}$$

In Eq. (17), for $l = 1$, b_0 is set to be a large value approaching infinity, such that for the design point $\boldsymbol{x}_{\mathrm{D}}^{(1)}$ on the first failure boundary, the stochastic process $\hat{h}_{\mathcal{D}}^{(l)}(\boldsymbol{x})$ tends to $\hat{h}_{\mathcal{D}}(\boldsymbol{x})$ used for defining $\mathcal{A}^{\mathrm{EI}}(\boldsymbol{x})$ in Eq. (10).

The reference point $\boldsymbol{x}_l^*$ is computed as the global maximum point of $\mu_h^{(l)}(\boldsymbol{x})$ within the l-th feasible region using, e.g., the PSO algorithm. Then, the acquisition function adapted from Eq. (10) for inferring

the design point of the l-th failure boundary is defined as:

$$\mathcal{A}_l^{\mathrm{EI1}}(\boldsymbol{x}) = \mathbb{E}_{\mathcal{D}}\left[\max\left(\hat{h}_{\mathcal{D}}^{(l)}(\boldsymbol{x}) - \mu_h^{(l)}(\boldsymbol{x}_l^*), 0\right)\right]. \tag{18}$$

Similarly, $\mathcal{A}_l^{\mathrm{EI1}}(\boldsymbol{x})$ measures the expected improvement of the objective function at $\boldsymbol{x}$ compared to its expected value at $\boldsymbol{x}_l^*$, and thus the next training point can be specified as the global maximum point of $\mathcal{A}_l^{\mathrm{EI1}}(\boldsymbol{x})$. The closed-form expression of $\mathcal{A}_l^{\mathrm{EI1}}(\boldsymbol{x})$ is derived as:

$$\mathcal{A}_l^{\mathrm{EI1}}(\boldsymbol{x}) = \max\left(\frac{1}{\|\boldsymbol{x}\|} - \frac{1}{\|\boldsymbol{x}_l^*\|}p_l(\boldsymbol{x}_l^*), 0\right) p_l(\boldsymbol{x}). \tag{19}$$

One can also easily devise the EI function $\mathcal{A}_l^{\mathrm{EI2}}(\boldsymbol{x})$ for the l-th intermediate failure boundary by modifying Eq. (12) following the similar scheme, and we don't repeat the details. The next point for enriching the GPR model is determined by maximizing the cheap-to-evaluate EI function in Eq. (19) within the l-th hyper-rectangular feasible region $\left[\boldsymbol{x}_{\mathrm{MPP}}^{(l-1)} - \boldsymbol{\xi}, \boldsymbol{x}_{\mathrm{MPP}}^{(l-1)} + \boldsymbol{\xi}\right]$.

Based on the above developments in this subsection, the pseudocode of the ConBaySubOpt method is summarized in Algorithm 2. Except for the parameters N_0, ϵ and Δ^{EI}, as required for Algorithm 1, there are three more parameters, i.e., p_0, N and ξ. The values of N_0, ϵ and Δ^{EI} can be set to be the same as those in Algorithm 1. The probability p_0 is used for setting the distance of the design point $\boldsymbol{x}_{\mathrm{D}}^{(l)}$ on the l-th level to $\boldsymbol{x}_{\mathrm{D}}^{(l-1)}$ of the $(l-1)$-th level, and it should be specified jointly with ξ, which determines the size of each feasible region. According to numerical validations, Algorithm 1 is applicable for the case that the distance of the design point to the origin is less than 2.5. Thus, ξ is suggested to be $2 \sim 4$. Accordingly, p_0 is suggested to be a value between $0.01 \sim 0.05$ such that the design point $\boldsymbol{x}_{\mathrm{D}}^{(l)}$ is located within the feasible region with $\boldsymbol{x}_{\mathrm{D}}^{(l-1)}$ as center. The smaller p_0 is, the less intermediate failure surfaces will be introduced, results in higher rate of convergence, but if it is smaller than 10^{-2}, the algorithm may be unsteady for highly nonlinear problems due to the difficulty of approaching $\boldsymbol{x}_{\mathrm{D}}^{(l)}$ from $\boldsymbol{x}_{\mathrm{D}}^{(l-1)}$. The sample size N for each feasible region is introduced for numerically specifying the failure threshold value b_l using MCS estimator, and it is recommended to be $100/p_0$. There are two stopping conditions for Algorithm 2, where the first one is responsible for the determining whether estimating each $\boldsymbol{x}_{\mathrm{D}}^{(l)}$ has been converged, and the one given by Eq. (14) is utilized, as the same in Algorithm 1, and the second one is used for judging if the true design point on the failure boundary $g(\boldsymbol{x}) = 0$ has been correctly identified, and it is implied by $b_l = 0$.

Algorithm 2: Constrained Bayesian Subset Optimization (ConBaySubOpt) for distant design point

Input: N_0, ϵ, α, Δ^{EI}, $g(\boldsymbol{x})$, p_0, N, ξ

Output: $b_1 \sim b_m$, $\boldsymbol{x}_{\mathrm{D}}^{(1)} \sim \boldsymbol{x}_{\mathrm{D}}^{(m)}$,$N_{\mathrm{call}}$, $\mathcal{D}$, $\hat{g}_{\mathcal{D}}(\boldsymbol{x})$

Initialize $l = 1$, $\boldsymbol{x}_{\mathrm{D}}^{(0)} = \mathbf{0}$, StopFlag1=False and b_0 as a large value approaching infinity;

Generate the initial training data $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ of size N_0 with $\mathcal{X}$ produced by LHS and $\mathcal{Y} = g(\mathcal{X})$ generated by calling the LSF, let $N_{\mathrm{call}} = N_0$;

while *StopFlag1=False* **do**

  Initilize StopFlag2=False and $i = 0$;

  Produce N samples $\boldsymbol{x}^{(k)}$ $(k = 1, 2, \cdots, N)$ by sampling from Gaussian distribution with mean $\boldsymbol{x}_{\mathrm{MPP}}^{(l-1)}$ and identity covariance matrix using, e.g., LHS;

  while *StopFlag2=False* **do**

    Let $i = i + 1$ and train the GPR model $\hat{g}_{\mathcal{D}}(\boldsymbol{x})$;

    Compute the posterior mean prediction for each $\boldsymbol{x}^{(k)}$, sort these values in ascending order, and update b_l as the $\lfloor p_0 N \rfloor$-th value;

    if $b_l < 0$ **then** Let $b_l = 0$;

    Compute the reference point $\boldsymbol{x}^{*(i)}$ by maiximizing the posterior mean of $\hat{h}_{\mathcal{D}}^{(l)}(\boldsymbol{x})$ with PSO in the feasible region $\left[\boldsymbol{x}_{\mathrm{D}}^{(l-1)} - \boldsymbol{\xi}, \boldsymbol{x}_{\mathrm{D}}^{(l-1)} + \boldsymbol{\xi}\right]$;

    Calculate $\boldsymbol{x}^{+}$ by maximizing $\mathcal{A}_l^{\mathrm{EI}}(\boldsymbol{x})$ within the support $\left[\boldsymbol{x}_{\mathrm{D}}^{(l-1)} - \boldsymbol{\xi}, \boldsymbol{x}_{\mathrm{D}}^{(l-1)} + \boldsymbol{\xi}\right]$;

    if $\mathcal{A}_l^{\mathrm{EI}}(\boldsymbol{x}^{+}) < \Delta^{\mathrm{EI}}$ **then**

      Let $\boldsymbol{x}_{\mathrm{D}}^{(l)} = \boldsymbol{x}^{*(i)}$, and StopFlag2=True

    else

      Calculate $y^{+} = g(\boldsymbol{x}^{+})$, add this point to $\mathcal{D}$, and let $N_{\mathrm{call}} = N_{\mathrm{call}} + 1$

    end

  end

  if $b_l = 0$ **then**

    Let $m = l$, and StopFlag2=True

  else

    Let $l = l + 1$

  end

end

4. Benchmark studies

In this section, we introduce two test examples and three engineering systems, with input dimensions varying from two to seventeen, and reliability index varying from about 0.4 to 4.3, for demonstrating the

effectiveness of the proposed methods. The "particleswarm" function in Matlab with default parameter setting is utilized for computing the global maximum point of the EI acquisition function in each iteration. One notes that, as the EI functions have closed-form expressions, searching its maximum point with the PSO algorithm does not need to call the LSFs, and thus is computationally cheap.

4.1. A Two-dimensional Numerical Example

We first introduce a two-dimensional numerical example, which is highly nonlinear and has many local optima, for explaining the training details and demonstrating the performance of both algorithms. The LSF is formulated as:

$$g\left(x_1, x_2\right) = \left(x_1 - 1\right)^3 + \left(x_2 - 2\right)^2 + x_1 \sin\left(2\pi x_2\right) \cos\left(2\pi x_1\right) + a \tag{20}$$

, where x_1 and x_2 are random input variables, and a is a deterministic parameter determining the magnitude of the reliability index β. We consider three cases. For case 1, a is set to be 0 and 20 to demonstrate Algorithm 1 (ConBayOpt) for estimating design point with small β, and for case 2, a is set to be 80 and 150 to prove the effectiveness of Algorithm 2 (ConBaySubOpt) for estimating distant design point. For both case 1 and case 2, x_1 and x_2 are assumed to be independent and standard normal variables. The aim of case 3 is to prove the suitability of the proposed methods to problems with non-Gaussian and correlated variables. Thus, for case 3, x_1 and x_2 are assumed to be correlated variables uniformly distributed in the circle bounded by $x_1^2 + x_2^2 = 9$, and a is set to be 60.

For implementing ConBayOpt for case 1, the algorithm parameters are set to be $\epsilon = 0.1$, $\Delta^{\mathrm{EI}} = 10^{-6}$ and $N_0 = 12$. The 12 initial training points are generated with LHS within the support of $[-2, 2]$ for each input. For improving the robustness, the algorithm ends when the stopping condition in Eq. (14) is satisfied for two consecutive times. The training details for $a = 0$, including the initial training points, the actively designed training points, the failure boundary $\mu_g(\boldsymbol{x}) = 0$ identified with the final GPR model, and the identified design point, are schematically shown in Figure 3, together with the true failure boundary $g(\boldsymbol{x}) = 0$ and the density contour for illustrating the accuracy of results. As can be seen, the true failure boundary shows multi-modal behavior, and this can be a challenge for using a gradient-based method. It is shown that, the ConBayOpt algorithm actively produces 6 more points when the stopping condition is satisfied. Thus the total number of LSF calls is 18. As the true design point is defined as the point, on the failure boundary, having the highest density value, it can be concluded by checking the density contour and the failure boundaries in Figure 3 that, the estimated design point is of high accuracy, and the resultant GPR model approximates the LSF with high accuracy around the design point.

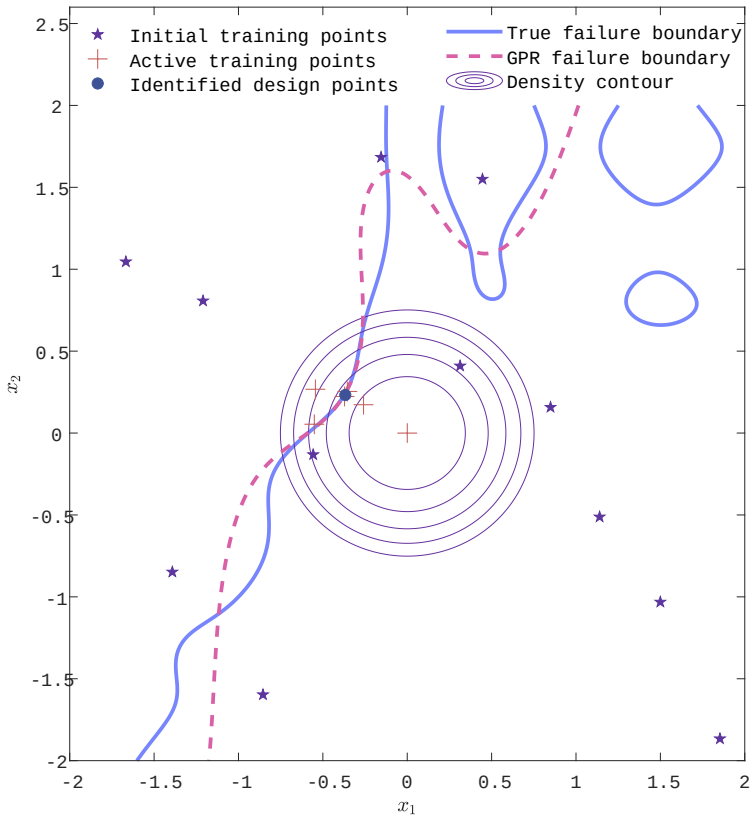


Figure 3: Training details and result of the design point generated for the two-dimensional numerical example with $a = 0$.

For case with $a = 20$, all the algorithm parameters are set to be the same as those for $a = 0$, and the training details are visually shown in Figure 4. As can be seen,this design point is farther away from the origin than that of the case with $a = 0$, but the reliability index β is still smaller than 2.5, thus the ConBayOpt algorithm still applies. Still, there are many local optima on the failure boundary, making the estimation of the design point challenging. Initialized with 12 training points, 10 more training points are produced with the EI acquisition function before touching the stopping condition for two successive times. Comparing the density contour and the failure boundaries, it can be found in Figure 4 that the design point is correctly identified. The total number of LSF calls is $12 + 10 = 22$, which is larger than that for the case with $a = 0$, but is still acceptable for engineering application.

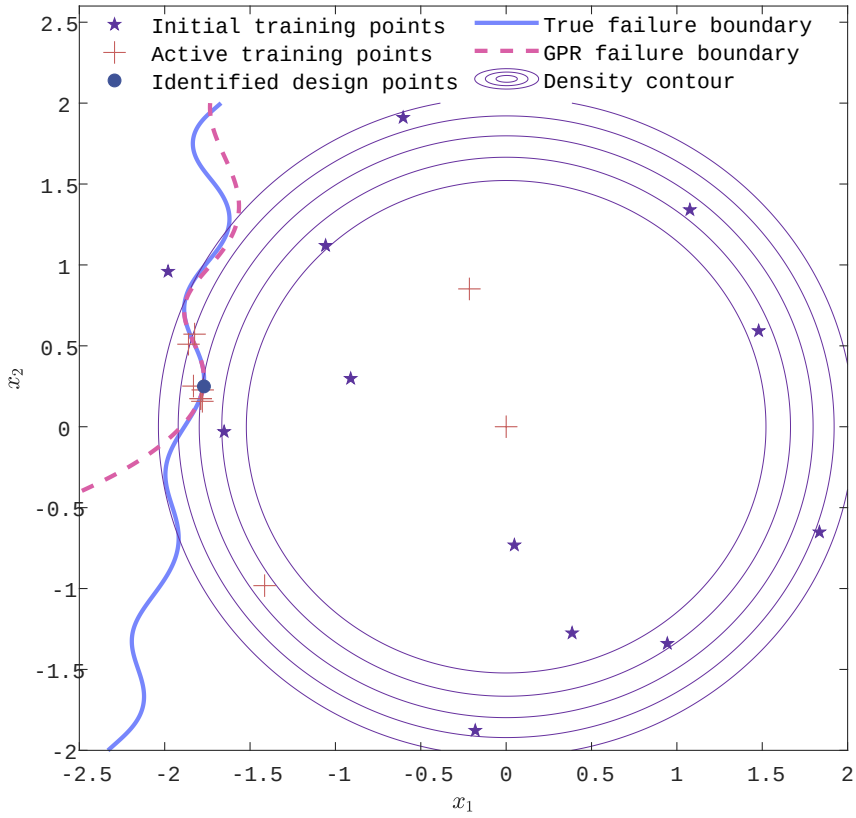


Figure 4: Training details and result of the MPP generated for the two-dimensional numerical example with $a = 20$.

For case 2 with $a = 80$ and $a = 150$, the distances of design points to the origin are both higher than 2, thus the ConBayOpt algorithm is not applicable, and the ConBaySubOpt Algorithm is implemented for this case. The parameters ϵ, Δ^{EI} and N_0 are set to be the same as for the above two cases, and the extra parameters are set as $p_0 = 0.02$, $\alpha = 1$, $\xi = 2$ and $N = 10^4$. Initialized with the 12 training points, the ConBaySubOpt algorithm automatically produced one intermediate failure boundary with $b = 58.44$ for approaching the true failure surface where the design point is located. The algorithm produces 11 more training points for identifying the intermediate failure surface and estimating the corresponding design point $\boldsymbol{x}_{\mathrm{D}}^{(1)}$ with acceptable accuracy, as shown in Figure 5. It is then, with $\boldsymbol{x}_{\mathrm{D}}^{(1)}$ as center of the second feasible region, 9 more training points are actively produced for learning the true failure design point $\boldsymbol{x}_{\mathrm{D}}^{(2)}$ with pre-specified error tolerance. Checking the density contour and the true failure surface in Figure 5, it can be found that design point $\boldsymbol{x}_{\mathrm{D}}^{(2)}$ on the true failure surface is correctly identified. The total number of LSF calls is then $12 + 11 + 9 = 32$, indicating that the ConBaySubOpt is highly efficient for estimating this distant design point.

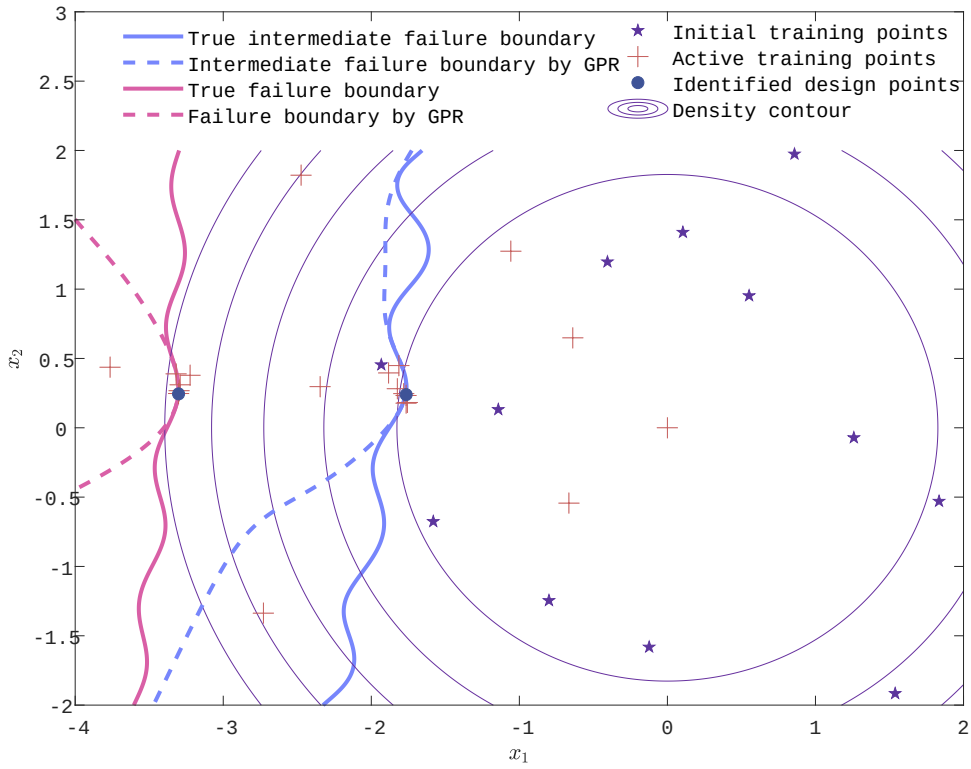


Figure 5: Training details of Algorithm 2 (ConBaySubOpt) for the two-dimensional numerical example with $a = 80$.

To further explore the potential of the ConBaySubOpt algorithm for estimating distant design point, we consider a more challenging case with $a = 150$. All the parameters are set to be the same as those for $a = 80$. The training details are schematically displayed in Figure 6. As can be seen, two intermediate failure surfaces are adaptively produced with the corresponding design points $\boldsymbol{x}_{\mathrm{D}}^{(1)}$ and $\boldsymbol{x}_{\mathrm{D}}^{(2)}$ being correctly identified with acceptable accuracy. With the second design point $\boldsymbol{x}_{\mathrm{D}}^{(2)}$ as center of the feasible region, the design point on the true failure surface $g(\boldsymbol{x}) = 0$ has also been correctly identified with high accuracy. The numbers of actively added points for learning the three design points $\boldsymbol{x}_{\mathrm{D}}^{(1)} \sim \boldsymbol{x}_{\mathrm{D}}^{(3)}$ are 8, 10 and 5 respectively, thus the total number of LSF calls is $12 + 8 + 10 + 5 = 35$, which is slightly larger than that for the case with $a = 80$, although one more intermediate failure surface has been produced.

Comparing Figure 3 with Figure 6, it is seen that, for the case with small reliability index β, the initial training points are distributed around the true design point, and thus the true failure boundary can be easily reached by active learning GPR model initialized with the initial training points; however, for the case with distant design point, the failure boundary is far from the origin, and despite equipped with the EI acquisition functions, the ConBayOpt algorithm finds difficulty in approaching the true design point especially when the LSF shows high non-linearity. Fortunately, with the adaptive and automatic

introduction of the intermediate failure surfaces and the corresponding feasible regions, it is then practical to identify the design point even it is very far from the origin. Besides, it is obvious that, for the case with design point near to the origin, ConBaySubOpt degrades into ConBayOpt, thus the users do not need to know the magnitude of the reliability index in advance. If the users know exactly the reliability index β is smaller than 2.5, then ConBayOpt can be utilized; otherwise, ConBaySubOpt should be utilized.

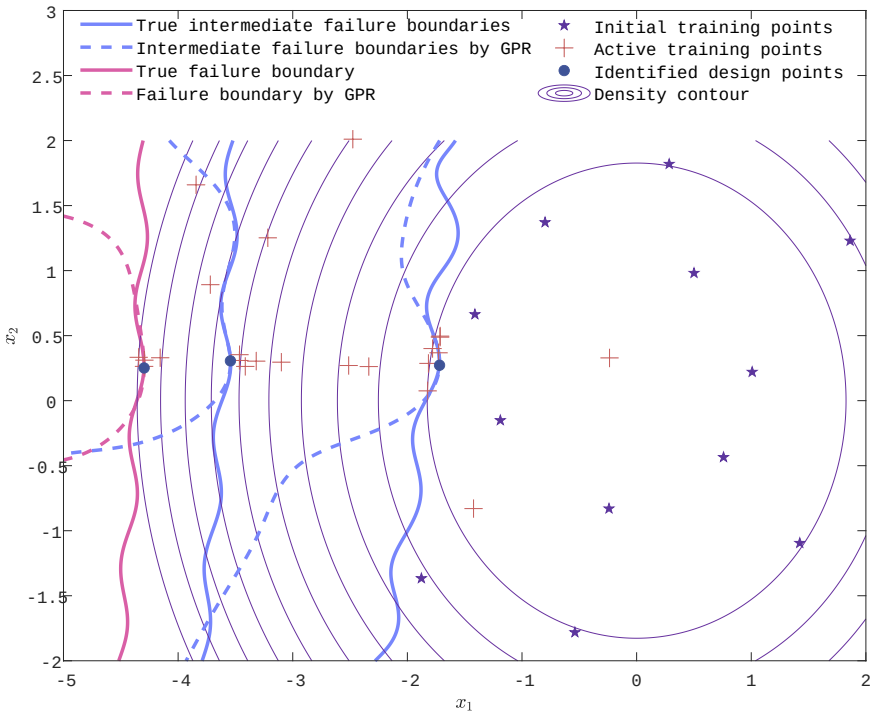


Figure 6: Training details of Algorithm 2 (ConBaySuOpt) for the two-dimensional numerical example with $a = 150$.

We then implement ConBaySubOpt algorithm for case 3 where the input variables are correlated and uniformly distributed in the circle with radius being 3. For this case, we need to transform the input variables from the original uniform space (denoted as X-space) into the standard normal space (referred to as U-space) using the Rosenblatt transformation [51]. The transformation is formulated in closed form as:

$$\begin{cases} u_1 = \Phi^{-1}\left(\frac{x_1+3}{6}\right) \\ u_2 = \Phi^{-1}\left(\frac{x_2+\sqrt{9-x_1^2}}{2\sqrt{9-x_1^2}}\right) \end{cases} \tag{21}$$

, or inversely

$$\begin{cases} x_1 = 6\Phi\left(u_1\right) - 3 \\ x_2 = 2\Phi\left(u_2\right)\sqrt{9-\left(6\Phi\left(u_1\right)-3\right)^2} - \sqrt{9-\left(6\Phi\left(u_1\right)-3\right)^2} \end{cases} . \tag{22}$$

The two variables u_1 and u_2 generated with the above transformation follow independent standard normal distribution. Substituting Eq. (22) into the g-function, we can reformulate the g-function in the U-space, and then the ConBaySubOpt algorithm is implemented with the same setting for parameters as case 2. The training details and the results are schematically shown in Figure 7. One notes that the algorithm is implemented in the U-space, and then the results are mapped to the X-space. As can be seen, both intermediate and true failure surfaces show distinct behavior in the U-space and X-space, and this is mainly caused by the nonlinear Rosenblatt transformation. Despite this difference, by introducing an intermediate failure surface, the design point is correctly identified by the algorithm with a total number of 34 g-function calls. This indicates that the proposed method is applied to problems with correlated and non-Gaussian random input variables.

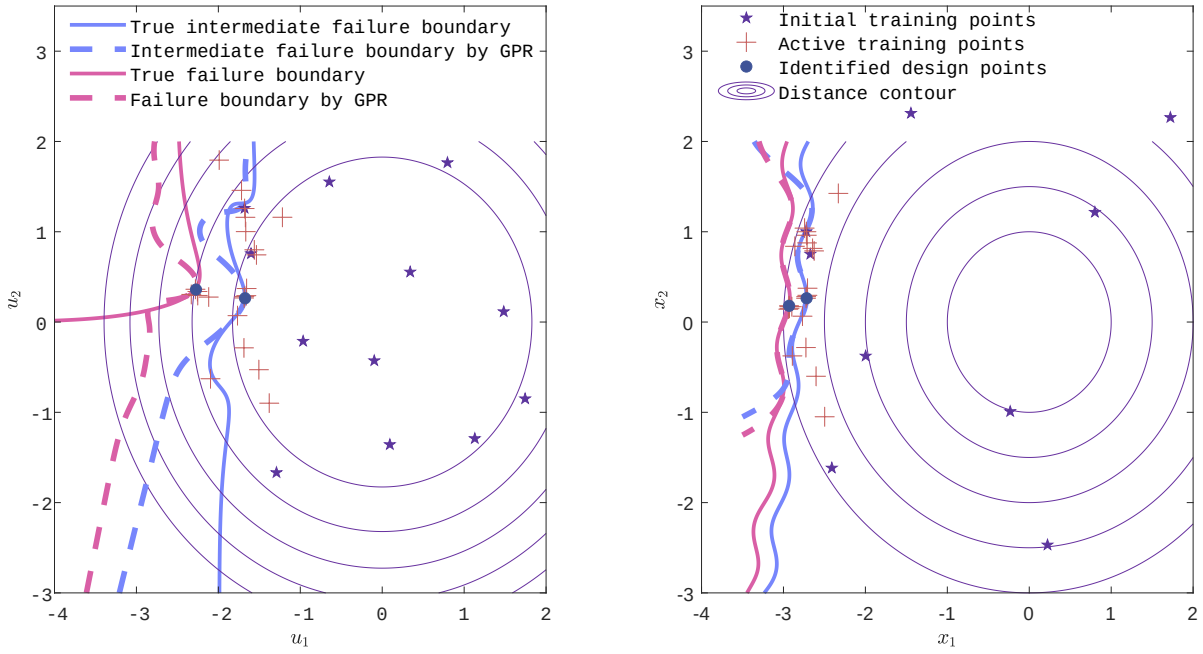


Figure 7: Training details of Algorithm 2 (ConBaySubOpt) for case 3 of the two-dimensional numerical example, where the left panel shows the training details in U-space, and the right one shows those in X-space. One notes that the legends in the figure apply to both sub-figures.

The HLRF method (see Refs. [34, 35]) has also been tested for this example, and results show that only for the first case with $a = 0$, the HLRF method produces the correct estimation of design point, but with many more LSF calls, and for the other cases, HLRF converges to local optima. Thus, the results of HLRF are not presented. Besides, all the above results shown above are generated with the first EI function $\mathcal{A}^{\mathrm{EI1}}(\boldsymbol{x})$. As the second one produces similar results, they are not presented for brevity.

4.2. A Non-Gaussian Numerical Example

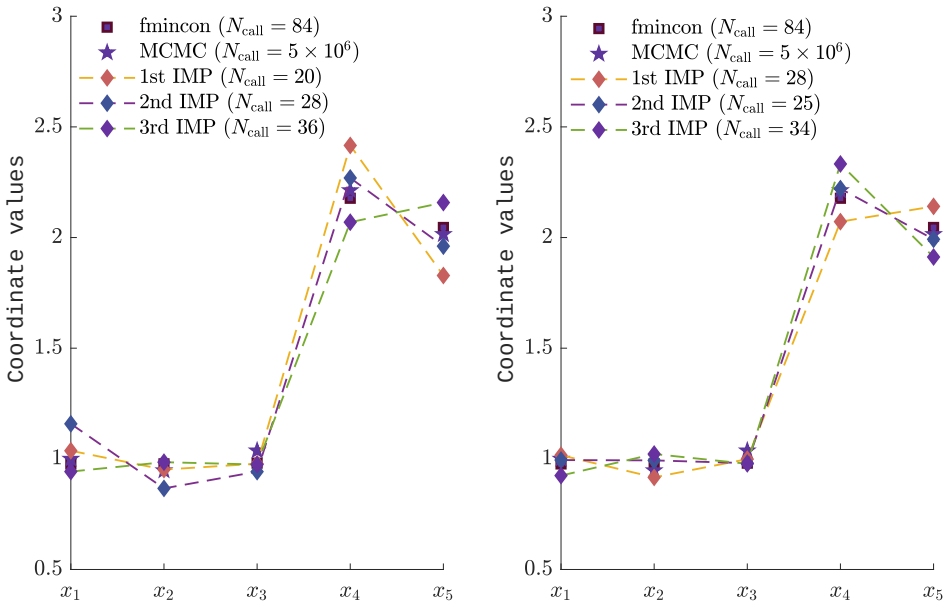
Consider a five-dimensional example with the LSF formulated as:

$$g(\boldsymbol{x}) = 0.125x_1 + 0.125x_1x_2^2 + 0.25x_3\sin(x_4) + 0.5x_5^3\cos(x_4) + a \tag{23}$$

, where a is a constant determining the magnitude of β, and all the input variables are assumed to be independent and follow lognormal distribution with zero mean and STD being 0.5. In this study, a is set as 2 for illustrating ConBayOpt, and as 50 for demonstrating ConBaySubOpt, with specific interest on their suitability to problems with multi-dimensional non-Gaussian inputs and highly-nonlinear LSF.

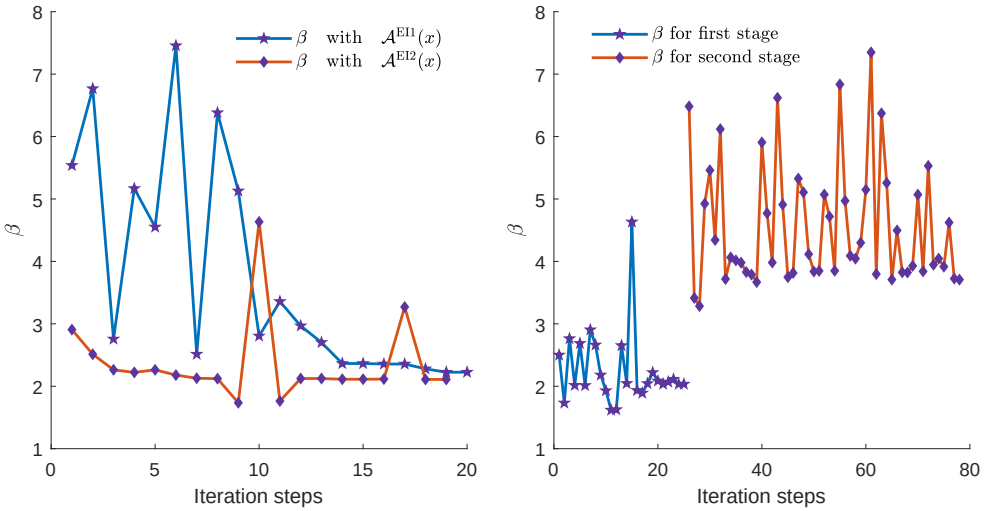
For this problem with more than two input variables, it is not possible to visually show the training details in the input space, and thus we use the reliability index β, which is defined as the distance of the design point to the origin in the standard normal space, for illustration. As HLRF does not converge for this problem, we use MCMC to generate the reference results for design point following the idea of Ref. [65]. In detail, a set of $N_{\text{call}} = 5 \times 10^6$ conditional samples are generated following the optimal importance sampling density $\phi_{\text{opt}}(\boldsymbol{x}) = I_F(\boldsymbol{x})\,\phi(\boldsymbol{x})/p_f$ using MCMC, and the one with the highest density value is specified as the reference design point. As a comparison, the Matlab function "fmincon" is also implemented for computing the design point.

To implement ConBayOpt for case $a = 2$, the parameters are set to be $N_0 = 12$, $\epsilon = 0.05$, and $\Delta^{\text{EI}} = 10^{-6}$. The ConBayOpt algorithm is then implemented three times independently, with each of the EI functions $\mathcal{A}^{\text{EI1}}(x)$ and $\mathcal{A}^{\text{EI2}}(x)$, and the results generated in the lognormal space are then reported in Figure 8, where "IMP" refers to "implementation". As can be seen, both EI functions produce design point results in good agreement with the reference solutions, and it is hard to say which EI function is better. It is also shown that the numbers of LSF calls consumed by the two EI functions are also quite similar. Thus, the two EI functions are close in terms of both efficiency and accuracy. For showing the training details of ConBayOpt, the evolution of the reliability index β for one implementation of each EI function is schematically shown in Figure 9 (a). The distinct behavior of β produced by the two acquisition functions at the first few steps are partly caused by the fact that different initial training points are utilized, and partly caused by the large prediction uncertainty at the reference uncertainties at the first few steps. The reference value of the reliability index β computed by MCMC and "fmincon" is 2.1222 and 2.1158 respectively, and thus both EI functions result in convergence of the reliability index. As the two EI functions show similar performance towards convergence, below we don't mention which one is utilized. It is also noted that all the three implementations of the ConBayOpt algorithm consume less LSF calls than the "fmincon" function, indicating the high efficiency of the proposed method.



(a). Results with $\mathcal{A}^{\text{EI1}}(x)$ (b). Results with $\mathcal{A}^{\text{EI2}}(x)$

Figure 8: Results of design point in the lognormal space for case $a = 2$ of the non-Gaussian numerical example, generated by implementing the ConBayOpt algorithm for three times with two EI functions. The reference solutions generated with MCMC and "fmincon" are also provided for comparison.

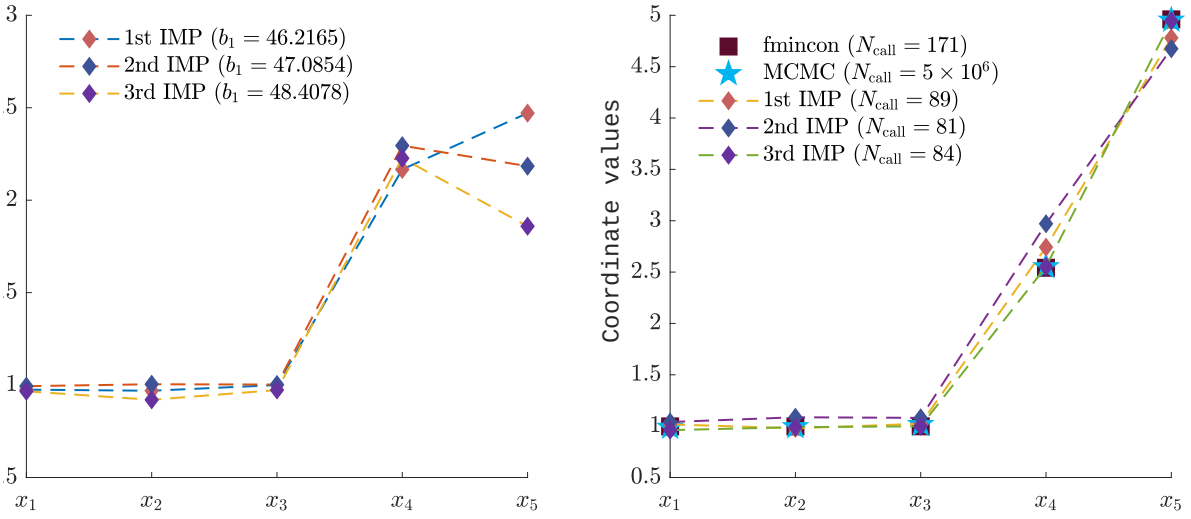


(a). β for $a = 2$ (b). β for $a = 50$

Figure 9: Evolution of the reliability index of the non-Gaussian numerical example. One notes that abrupt jump of of β on the right-hand plot is due to the movement to the second stage.

We then implement ConBaySubOpt for case $a = 50$ by setting $p_0 = 0.01$, $\xi = 4$ and all the other

parameters, except Δ^{EI}, the same as the implementations of ConBayOpt. The stopping threshold Δ^{EI} is set to be 10^{-5} for the first intermediate failure boundary, and then reduced by 90 percent successively for the subsequent (intermediate) failure boundaries. Similarly, we perform three independent runs of ConBaySubOpt for demonstrating the robustness of the algorithm. Each implementation produces one intermediate failure boundary with slightly different failure threshold value b_1, and the coordinate values of the design point for both the intermediate and the true failure boundaries are shown in Figure 10. The reference solutions are not provided for the intermediate failure boundary as the b_1 value is actively specified to be slightly different. Comparing Figure 10(a) and 10(b), it is found that the generated design points on the intermediate failure surface show slightly larger variation than those on the true failure surface across the three implementations, and this is mainly caused by the variation of the b_1 values across the three implementations. Whatever, this will not affect the quality of the results of the design point on the true value surface, as demonstrated by the comparison in Figure 10(b). As the results of all the three independent runs are in good agreement with the reference solutions produced MCMC and "'fmincon', it is concluded the ConBaySubOpt algorithm is of high accuracy and numerical robustness for this example. The total numbers of LSF calls consumed by the three independent runs are 89, 81 and 84, which are all smaller than that consumed by "fmincon", indicating the numerical efficiency of the algorithm. The evolution of the reliability index β against the iteration steps is visually shown in Figure 9(b). It is seen that algorithm automatically proceeds to the training for the second stage after that for the reliability index of the first stage being converged. The values of β produced by the three implementations are 3.7236, 3.7831 and 3.7072, and the reference values generated by MCMC and "fmincon" are 3.7078 and 3.7069. This further demonstrates the robustness and accuracy of the ConBaySubOpt algorithm. It is shown in Figure 10 that the proposed ConBaySubOpt algorithm requires much less LSF calls than "fmincon" for all the three independent implementations, indicating the high efficiency of the proposed method.



(a). Design points on intermediate failure boundary

(b). Design points on real failure boundary

Figure 10: Results of design point in the lognormal space for case $a = 50$ of the non-Gaussian numerical example generated by implementing the ConBaySubOpt algorithm three times.

4.3. A Ten-bar Frame Structure

Let's consider a ten-bar frame structure shown in Figure 11(a), with the FEM model built in ANSYS and shown in Figure 11 (b). This problem is adapted from Ref. [66], and where it was found that the HLRF algorithm cannot correctly estimate the MPP. A total number of fifteen random input variables are involved in this example, and they are the length L of horizontal and vertical bars, the section areas A_i $(i = 1, \cdots, 10)$ of the ten bars, the Young's modulus E of material, and the three point loads P_1, P_2 and P_3, as shown in Figure 11(a). All these fifteen variables are assumed to follow Gaussian distribution with mean values 1m, 0.001m^2, 100GPa, 80kN, 10kN and 10kN, respectively, and the coefficients of variation of all variables are assumed to be 2%. The limit state function is formulated as $g = 0.0033 - \Delta_y$ with Δ_y indicating the vertical displacement of node 3.

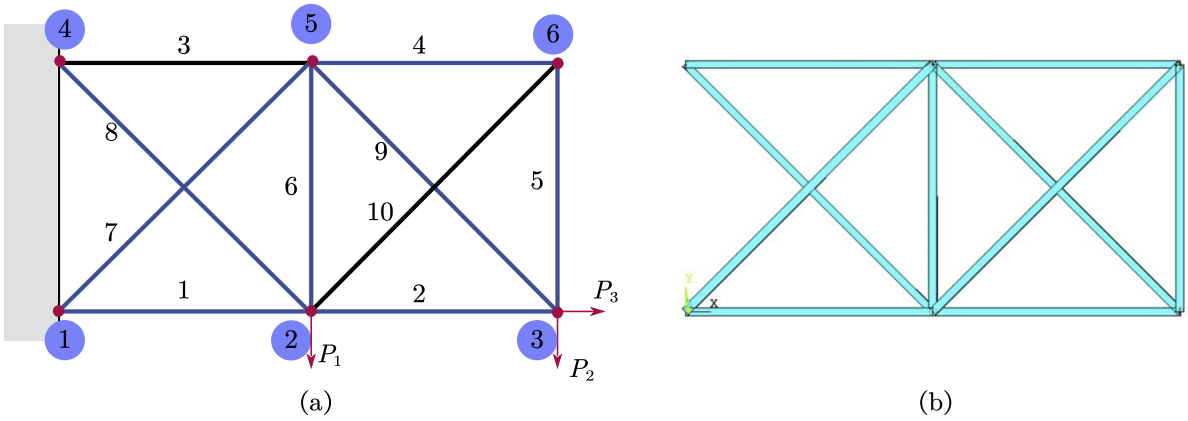
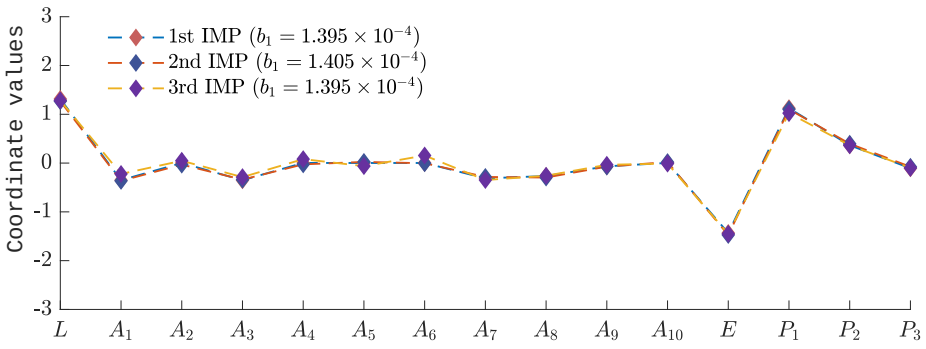
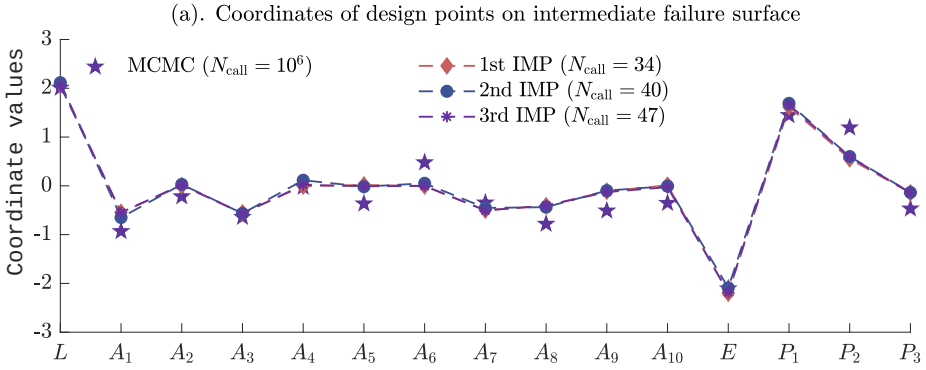


Figure 11: A ten-bar frame structure with (a) indicating the number of bars and loads, and (b) being the FE model.

The reliability index of this example is higher than 3, thus we implement ConBaySuOpt to compute the design point. The algorithm parameters are set to be $N_0 = 15$, $\epsilon = 10^{-6}$, $\alpha = 1$, $\Delta^{\mathrm{EI}} = 10^{-4}$ for the first intermediate failure boundary and then reduced by 90 percent for successive intermediate failure surface, $p_0 = 10^{-2}$ and $N = 10^4$. The results of three independent runs are shown in Figure 12, together with the reference solution computed by MCMC with 10^6 LSF calls. All the three independent runs automatically produce one intermediate failure boundary with the design points shown in Figure 12. As can be seen, the intermediate failure threshold values produced by the three independent runs are very close, resulting in close positions of design points. The results of the design points in standard normal space on the true failure boundary are shown in Figure 12. It is seen that the results of all the three independent runs are generally in good agreement with the reference solutions, indicating the robustness and accuracy of the ConBaySubOpt algorithm for this example. The values of the reliability index β estimated by the three implementations are 3.6305, 3.6392 and 3.6304, which are all slightly smaller than the reference result $\beta = 3.8735$. The small difference between the results of ConBaySubOpt and MCMC is caused by the constraint $|g(\boldsymbol{x})| \leqslant \epsilon$ as the final results are almost located on the boundary $g(\boldsymbol{x}) = \epsilon$, instead of $g(\boldsymbol{x}) = 0$. Reducing the value of ϵ can make the results closer to the reference solution, but it may also results in unsteadiness or at least more LSF calls for convergence as the EI functions can be more exploitative. The total number of LSF calls consumed by the three independent runs are 34, 40 and 47, respectively, as shown in Figure 12, demonstrating the high efficiency of the ConBaySuOpt algorithm.



(a). Coordinates of design points on intermediate failure surface



(b). Coordinates of design points at true boundary

Figure 12: Positions of design points in standard Gaussian space for the ten-bar structure.

4.4. Hydrodynamic lubrication reliability of a journal bearing

The failure of an aero-engine pump is mostly due to the wear of its several friction pairs, where the journal sliding bearing pair commonly is one of the most critical parts. For the journal bearing working with high speed, the hydrodynamic lubrication provides support to the shaft by producing oil film with pressure. However, as the pressure of the film exceed a threshold related to the material of the shaft, failure may occur. Therefore, we develop a computational model for simulating the hydrodynamic lubrication, and for predicting the thickness and pressure of the film by solving a set of PDEs consisting of Reynolds equation, elastic deformation equation, energy equation and temperature-viscosity equation. For simplicity, we don't give details of these equations and the numerical solver. The predicted pressure distribution along the shaft and circumferential directions predicted at nominal point are shown in Figure 13, where $p_{\max}$ refers to the peak of the film pressure, which depends on six random input parameters, i.e., the width B of bearing, the radius R of the bearing, the radial gap C, the rotation speed ω, the oil viscosity η, and the eccentricity ratio r. Thus, the LSF of this problem is formulated as:

$$g = p_m - p_{\max}\left(B, R, C, \omega, \eta, r\right) \tag{24}$$

, where p_m refers to the maximal allowable threshold pressure, which is determined by the material property, and assumed to be 0.75 MPa. The six input variables are assumed to be independent and follow truncated normal distribution with distribution parameters shown in Table 1.

Table 1: Distribution parameters of the journal bearing

Variables	Distribution type	Means	C.O.Vs
Width B (mm)	Normal	25	0.005
Radius R (mm)	Normal	12.5	0.005
Radial gap C (mm)	Normal	0.02	0.005
Rotate speed ω $(\mathrm{r} \cdot \mathrm{min}^{-1})$	Normal	8000	0.03
Viscosity η $(\mathrm{Pa} \cdot \mathrm{s})$	Normal	9.66e-4	0.3
Eccentricity ratio r	Normal	0.8	0.03

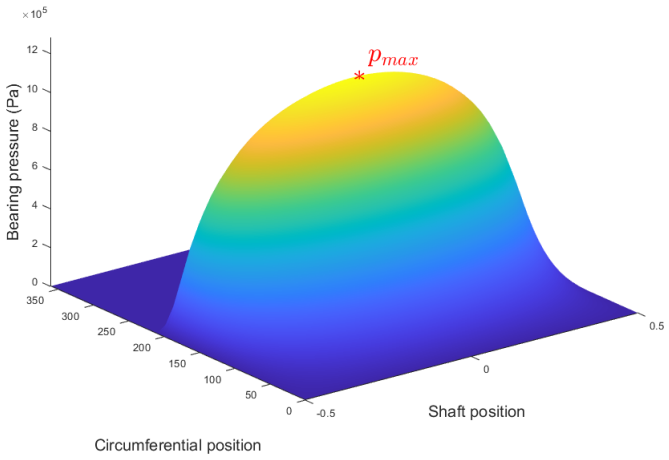
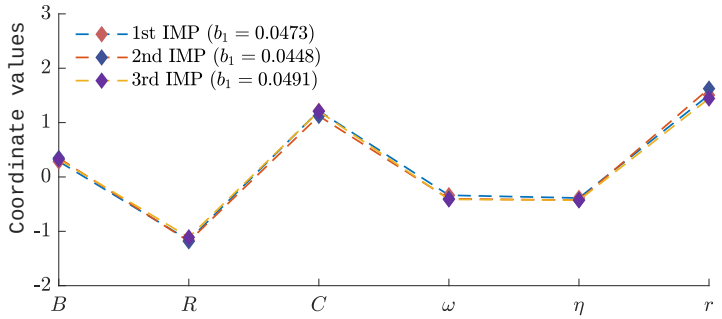


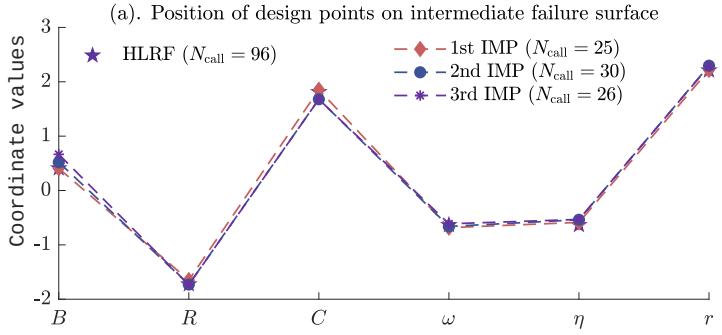
Figure 13: Distribution of film pressure against the shaft and circumferential positions for the journal bearing.

As the reliability index β of this example is higher than 2.5, the ConBaySubOpt algorithm is implemented. The algorithm parameters are set as follows: $N_0 = 12$, $\epsilon = 5 \times 10^{-4}$, $\alpha = 1$, $p_0 = 10^{-2}$, $N = 10^4$, $\xi = 4$ and $\Delta^{\mathrm{EI}} = 10^{-4}$ for the first layer and then reduced by 90 percent for each successive layer. FORM can provide reliable results for this example, thus is served as reference method. The ConBaySubOpt algorithm is implemented with three independent runs, and each of which automatically produce one intermediate failure surface, and the results of design points for the two failure surfaces are reported in Figure 14, together with the reference solutions for comparison. As can be seen, the design points on the first failure boundary are in good accordance across the three implementations as the resultant values of b_1 are very close. From Figure 14(b), it is seen that the design points on the true failure surface are very close, and also match well with the reference solutions produced with HLRF. The evolution of the reliability index β produced by the third run is shown in Figure 15, and from which the convergence of the implementation can be convincingly confirmed. The total numbers of LSF calls consumed by these three implementation are 25, 30 and 26 respectively,

which are smaller than that of HLRF, indicating the high efficiency of ConBaySubOpt.



(a). Position of design points on intermediate failure surface



(b). Positions of design points at true boundary

Figure 14: Results of design points in the standard Gaussian space for the journal bearing example.

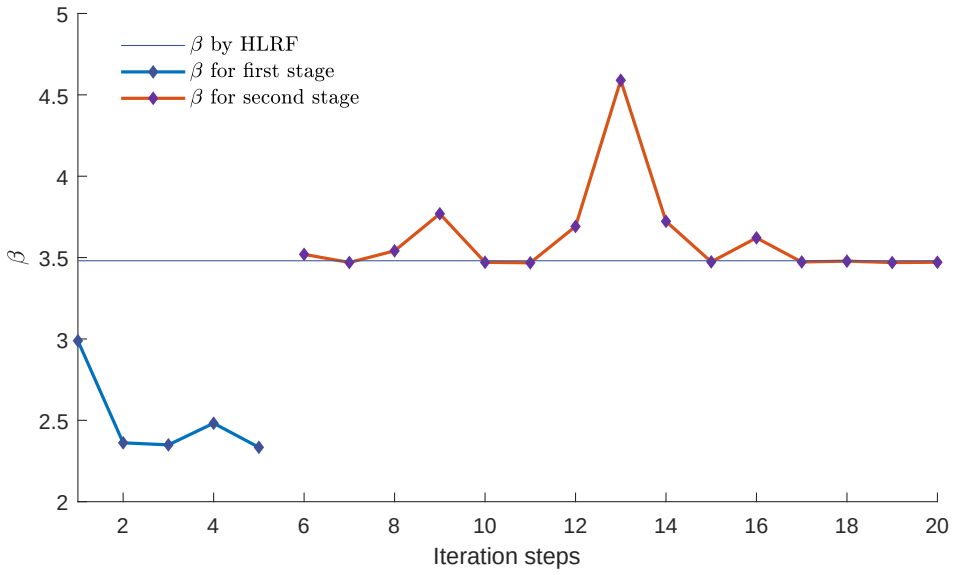


Figure 15: Evolution of reliability index β by ConBaySubOpt for the journal bearing example.

4.5. Water distribution network

Let us consider the reliability analysis of a water distribution network adapted from Ref. [67]. The layout of this network is shown in Figure 16. There are seventeen nodes with the first node having a prescribed head of 15 m, and all the other nodes have a demand of $0.005\text{m}^3/\text{s}$. The length of the pipe between Node 8 and Node 16 is 100 m, and that of all the other pipes is assumed to be 50 m. Further, the diameters of all pipes are assumed to be 0.11 m, and the material relative roughness is assumed to be 0.046 mm. The water temperature is 15°C. We consider the prescribed head at Node 1 and the demands of all the other sixteen nodes as random input variables following lognormal distribution. The mean values for the prescribed head at Node 1 and the demands of all the other sixteen nodes are assumed to be 15 m and $0.005\text{m}^3/\text{s}$ respectively, and the coefficients of variation of all the seventeen input variables are assumed to be 10%. The minimum allowable head is assumed to be 4 m, and the failure occurs when it is lower than this threshold. We use the water distribution system hydraulic and water quality analysis toolkit "EPANET" to implement this model.

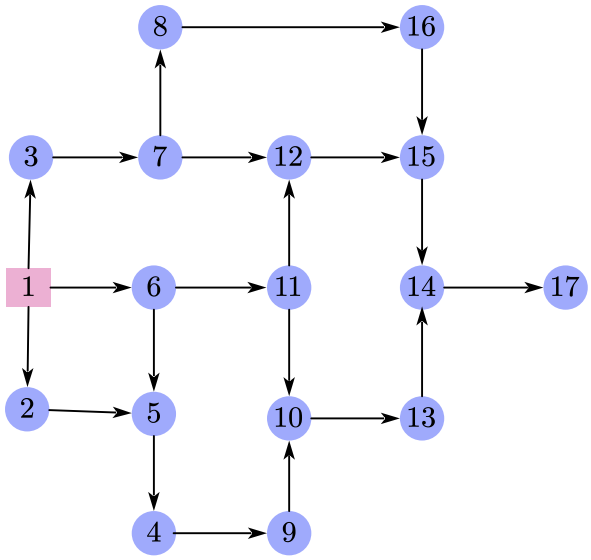


Figure 16: A water distribution network.

The ConBaySubOpt algorithm is initialized with the following parameter setting: $N_0 = 20$, $\epsilon = 0.02$, $\alpha = 1$, $p_0 = 10^{-2}$, $\zeta = 4$, and $\Delta^{\text{EI}} = 10^{-4}$ for the first layer and then reduced by 90% for each successive layer. With the above setting, the ConBaySubOpt algorithm is implemented for three times, and the corresponding results are reported in Table 2. For comparison, the reliability index computed by MCS with 10^5 samples is also reported for comparison. As for MCS, the coefficient of variation for the estimate of the failure probability is 2.1%. Thus, it is believed that the estimate of β computed by MCS is of high accuracy. For each of the three implementations of ConBaySubOpt, one intermediate failure surface was automatically produced, and the corresponding values of b_1 being 0.7160, 0.6937 and 0.7025 respectively.

The HLRF algorithm fails to find the design point for this example. From Table 2, it can be concluded that the results generated by the three implementations of the ConBaySubOpt algorithm are consistent, and the results of β computed by these three implementations are also in good agreement with the reference solution by MCS, indicating the high accuracy and robustness of the proposed ConBaySubOpt algorithm for this example. The total numbers of LSF calls of these three implementations are 37, 36, and 53, indicating the high efficiency of the proposed method.

Table 2: Results of design points for the water distribution network

Node	1st IMP	2nd IMP	3rd IMP	MCS
1	-2.7542	-2.7722	-2.7679	-
2	0.0628	0.0618	0.0662	-
3	0.0953	0.0725	0.0959	-
4	0.1778	0.1976	0.1644	-
5	0.0793	0.1210	0.1294	-
6	0.1424	0.1039	0.0969	-
7	0.2258	0.1616	0.1906	-
8	0.2767	0.2476	0.2349	-
9	0.2117	0.1781	0.1999	-
10	0.2152	0.2157	0.2417	-
11	0.1871	0.2465	0.2033	-
12	0.2080	0.1670	0.2321	-
13	0.2435	0.2486	0.2698	-
14	0.3592	0.2875	0.3096	-
15	0.2638	0.2900	0.2574	-
16	0.2490	0.2774	0.2738	-
17	0.4407	0.3895	0.3636	-
β	2.9100	2.9104	2.9078	2.8618
N_{call}	37	36	53	10^6

4.6. Final remarks

With all the five numerical and engineering examples examined in this section, it is now clear that the proposed methods are effective for searching the design point no matter it is far or near to the origin. However, there are still some specific aspects to be discussed. The first is on the global convergence of the proposed methods to LSFs with multi-modal behaviors. The Bayesian optimization algorithms, also

called Efficient Global Optimization (EGO) algorithms [54], as revealed by their name, are widely accepted as a set of optimization algorithms with global convergence for non-constrained optimization problem, and specifically, the global convergence of the EI function has been theoretically proved (see Ref. [68, 69]). It was concluded that, for a non-constrained optimization aiming at minimizing $g(\boldsymbol{x})$, if $g(\boldsymbol{x})$ is approximated with a GPR model with kernel $\kappa(\boldsymbol{x}, \boldsymbol{x}')$, the EI function is known to converge to the minimum of $g(\boldsymbol{x})$ if it is in the reproducing kernel Hilbert space (RKHS) defined by $\kappa(\boldsymbol{x}, \boldsymbol{x}')$. Thus, if $g(\boldsymbol{x})$ is a smooth function, the algorithm must converge to its minimum if the squared exponential kernel is utilized. The EI functions defined in this work are modified from the classical EI function to formulate the constrained optimization problem as a non-constrained one. Although it is not rigorously proved, it is believed that, as the LSF is in the RKHS equipped with the kernel $\kappa(\boldsymbol{x}, \boldsymbol{x}')$, the proposed methods should converge to the global optimum. A theoretical proof to this conclusion is beyond the scope of this work, thus we do not suggest absolutely positive arguments, and leave the theoretical proof to future work as a technical note.

One notes that, following the theory of the GPR model, although a Gaussian process assumption is made for approximating the LSF, it does not mean that the method only applies to the problems with $\boldsymbol{x}$ following normal or lognormal distribution. Indeed, the Gaussian process assumption is not imposed on the input variables, but on the deterministic-but-unknown value of the LSF at each location, and it does not restrict the distribution type of $\boldsymbol{x}$. One can refer to, e.g., Ref. [70], for example of GPR model applied to e.g., uniformly distributed input variables. For some active learning algorithms, such as those based on the knowledge-gradient acquisition function and the entropy search acquisition function [45], searching the global maximum/minimum of the acquisition functions may itself be more time-consuming than a FEM analysis. This is mainly due to the fact that these acquisition functions do not have closed-form expressions, and each call of them requires expensive numerical computation. However, the EI acquisition functions devised in this work all have closed-form expressions, and searching their global maximums is computationally cheap. For example, using a personal laptop with core i7 CPU, for the first numerical example, searching the global maximum of the EI function takes about $0.4 \sim 2$ seconds, and for the journal bearing example, each search consumes about $0.7 \sim 3$ seconds. This is also why we devised the acquisition functions based on EI function, instead of the knowledge-gradient function or the entropy search function [45], although they were stated to provide better trade-off between exploitation and exploration.

As revealed by the results of benchmark studies, two parameters, i.e, the error tolerance ϵ and the stopping threshold Δ^{EI}, differ obviously for different examples. As has been explained in subsection 3.1, the value of ϵ determines the smoothness of the acquisition function and also the distance of the resultant design point to the failure surface. Based on the experience of solving all the five test examples, it can be specified as a value between 0.5% and 3% of the STD of y, where the STD of y can be roughly estimated based on the N_0 training points before initializing the algorithms. This is also the reason why ϵ is set to be quite different values for different examples. The stopping threshold Δ^{EI} provides a balance between accuracy

and efficiency. A large value of Δ^{EI} may result in non-convergence of the result, and a very small value may consumes more LSF calls than expected. Based on the experience of investigating the five examples, it is found that setting Δ^{EI} as a value between 10^{-5} and 10^{-8} usually provides good balance between accuracy and efficiency. From the training process of the ConBaySubOpt algorithm for these examples, it is also found that, if multiple intermediate failure surfaces are required for approaching the distance design point, setting Δ^{EI} as 10^{-4} for the first intermediate failure surface and then reducing it by, e.g., 90 percents, for each subsequent intermediate (or true) failure surface, can provide a better trade-off in most cases. The initial training sample size N_0 can be set as $\max(12, n+1)$, and ξ is suggested to be 4. The values of p_0 and α are set to be 10^{-2} and 1 respectively for all examples. All the results of the five examples are generated following the above principles for parameter setting.

One should also note that, although the idea of subset simulation is utilized in the ConBaySubOpt algorithm for actively producing feasible regions, it does not inherit the local convergence phenomenon of the subset simulation, as the feasible region for searching the design point consists of both regions within and outside of the intermediate failure domain, and the term $\alpha\sigma_g(\boldsymbol{x})$ introduced in Eq. (15) puts more emphasis on global exploration. However, the proposed methods do have some limitations. First, for a rare extreme case where the true design point is in the opposite directions of the intermediate design points, the ConBaySubOpt algorithm may fail to find the global optimum, but this can be fixed by reformulating the method under the polar coordinate, and will be treated in future work. Second, it is not recommended to use the method for high-dimensional problems (typically, $n > 30$). This is caused by the limitation of the GPR model in high dimension. As Euclidean distance is utilized for defining the kernel function, the covariance with the same length of Euclidean distance may get less information for higher dimension, making the prediction performance of the GPR model poorer in high dimension. This can be alleviated by, e.g., introducing a built-in dimension reduction scheme for the GPR model [71], but this will make the training process time-consuming. Another limitation of the proposed methods is that they are not applicable for searching multiple design points. Indeed, the EI functions defined by both Eqs. (10) and (12) can only inform the global design point with the highest density value. However, this limitation can be eliminated by devising more informative acquisition functions and introducing multi-modal optimization algorithms for searching multiple optima of the new acquisition functions.

5. Conclusions and prospects

This work has developed two Bayesian optimization algorithms, i.e., ConBayOpt and ConBaySubOpt, to efficiently estimate the design points for scenarios where classical methods, such as HLFR, may not be applicable, due to the multi-modal property and the unavailability of gradient information of the LSF. The core development is the EI acquisition functions developed for constrained optimization, which allows

searching the design point in an active learning scheme. The ConBayOpt algorithm is applicable for the cases where the reliability index is small (typically less than 2.5) and the LSF is of high nonlinearity, while the ConBaySubOpt algorithm is devised, based on the idea of introducing intermediate failure boundaries, for estimating distant design point where the reliability index is large. Although ConBaySubOpt borrows idea from the subset simulation, it does not require the MCMC for creating conditional samples associated with the intermediate failure thresholds b_l $(l = 1, \cdots, m-1)$. Both algorithms are gradient-free, and applicable to problems, presented in this work, with multi-modal and highly nonlinear LSFs, thus provide great potential for improving all the design point based reliability analysis methods and, eventually, reliability-based optimization methods. For real-world applications, the users do not need to know the magnitude of the reliability index in advance, as the ConBaySubOpt algorithm automatically degrades into ConBayOpt if the reliability index is small.

The above superiority of the developed algorithms has been demonstrated by the results of the benchmark studies, but there are still space for improving. For example, dimension reduction techniques need to be embedded to the algorithms for addressing high-dimensional problems; extension of the developed algorithms to solve multiple design points (if exist) at once is also a promising direction, etc.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under grant number 12202358, 72171194 and 12220101002, and the Sino-German Mobility Programme under grant number M-0175 (2021-2023).

References

[1] B. Low, FORM, SORM, and spatial modeling in geotechnical engineering, Structural Safety 49 (2014) 56–64.
[2] G. J.-W. Hou, P. A. Newman, A most probable point-based method for reliability analysis, sensitivity analysis and design optimization, Tech. rep.
[3] D.-Q. Li, D. Zheng, Z.-J. Cao, X.-S. Tang, K.-K. Phoon, Response surface methods for slope reliability analysis: review and comparison, Engineering Geology 203 (2016) 3–14.
[4] I. Papaioannou, C. Papadimitriou, D. Straub, Sequential importance sampling for structural reliability analysis, Structural Safety 62 (2016) 66–75.
[5] A. Tabandeh, G. Jia, P. Gardoni, A review and assessment of importance sampling methods for reliability analysis, Structural Safety 97 (2022) 102216.
[6] I. Papaioannou, W. Betz, K. Zwirglmaier, D. Straub, MCMC algorithms for subset simulation, Probabilistic Engineering Mechanics 41 (2015) 89–103.
[7] S.-K. Au, J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probabilistic Engineering Mechanics 16 (4) (2001) 263–277.
[8] G. Jia, A. Tabandeh, P. Gardoni, A density extrapolation approach to estimate failure probabilities, Structural Safety 93 (2021) 102128.
[9] M. A. Valdebenito, P. Wei, J. Song, M. Beer, M. Broggi, Failure probability estimation of a class of series systems by multidomain line sampling, Reliability Engineering & System Safety 213 (2021) 107673.
[10] J. Song, P. Wei, M. Valdebenito, M. Beer, Adaptive reliability analysis for rare events evaluation with global imprecise line sampling, Computer Methods in Applied Mechanics and Engineering 372 (2020) 113344.
[11] B. Echard, N. Gayton, M. Lemaire, AK-MCS: an active learning reliability method combining kriging and monte carlo simulation, Structural Safety 33 (2) (2011) 145–154.

[12] X. Zhang, L. Wang, J. D. Sørensen, AKOIS: an adaptive kriging oriented importance sampling method for structural system reliability analysis, Structural Safety 82 (2020) 101876.
[13] J. Song, P. Wei, M. Valdebenito, M. Beer, Active learning line sampling for rare event analysis, Mechanical Systems and Signal Processing 147 (2021) 107113.
[14] H.-M. Qian, Y.-F. Li, H.-Z. Huang, Time-variant system reliability analysis method for a small failure probability problem, Reliability Engineering & System Safety 205 (2021) 107261.
[15] Y. Zhang, Y. Dong, J. Xu, An accelerated active learning kriging model with the distance-based subdomain and a new stopping criterion for reliability analysis, Reliability Engineering & System Safety 231 (2023) 109034.
[16] T. Zhou, Y. Peng, Reliability analysis using adaptive polynomial-chaos kriging and probability density evolution method, Reliability Engineering & System Safety 220 (2022) 108283.
[17] J. Li, Probability density evolution method: background, significance and recent developments, Probabilistic Engineering Mechanics 44 (2016) 111–117.
[18] G. Chen, D. Yang, A unified analysis framework of static and dynamic structural reliabilities based on direct probability integral method, Mechanical Systems and Signal Processing 158 (2021) 107783.
[19] R. H. Lopez, A. T. Beck, Reliability-based design optimization strategies based on form: a review, Journal of the Brazilian Society of Mechanical Sciences and Engineering 34 (2012) 506–514.
[20] X. Yuan, S. Liu, M. A. Valdebenito, M. G. Faes, D. J. Jerez, H. A. Jensen, M. Beer, Decoupled reliability-based optimization using markov chain monte carlo in augmented space, Advances in Engineering Software 157 (2021) 103020.
[21] L. Katafygiotis, K. Zuev, Geometric insight into the challenges of solving high-dimensional reliability problems, Probabilistic Engineering Mechanics 23 (2-3) (2008) 208–218.
[22] M. Valdebenito, H. Pradlwarter, G. Schuëller, The role of the design point for calculating failure probabilities in view of dimensionality and structural non-linearities, Structural Safety 32 (2) (2010) 101–111.
[23] K. Breitung, SORM, design points, subset simulation, and markov chain monte carlo, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 7 (4) (2021) 04021052.
[24] A. Der Kiureghian, Engineering Design Reliability Handbook, CRC Press, 2004, Ch. First- and Second-Order Reliability Methods.
[25] R. Melchers, Importance sampling in structural systems, Structural Safety 6 (1) (1989) 3–10.
[26] G. Schuëller, R. Stix, A critical appraisal of methods to determine failure probabilities, Structural Safety 4 (4) (1987) 293–309.
[27] P. Koutsourelakis, H. Pradlwarter, G. Schuëller, Reliability of structures in high dimensions, part I: Algorithms and applications, Probabilistic Engineering Mechanics 19 (4) (2004) 409–417.
[28] T. Fang, C. Jiang, Z. Huang, X. Wei, X. Han, Time-variant reliability-based design optimization using an equivalent most probable point, IEEE Transactions on Reliability 68 (1) (2018) 175–186.
[29] X. Li, H. Qiu, Z. Chen, L. Gao, X. Shao, A local Kriging approximation method using MPP for reliability-based design optimization, Computers & Structures 162 (2016) 102–115.
[30] T. Van Huynh, S. Tangaramvong, B. Do, W. Gao, S. Limkatanyu, Sequential most probable point update combining gaussian process and comprehensive learning pso for structural reliability-based design optimization, Reliability Engineering & System Safety (2023) 109164.
[31] H. Cho, K. K. Choi, J. Shin, Iterative most probable point search method for problems with a mixture of random and interval variables, Journal of Mechanical Design 142 (7) (2020) 071703.
[32] Z. Zhang, C. Jiang, G. Wang, X. Han, First and second order approximate reliability analysis methods using evidence theory, Reliability Engineering & System Safety 137 (2015) 40–49.
[33] S. Sankararaman, K. Goebel, A novel computational methodology for uncertainty quantification in prognostics using the most probable point concept, in: Annual Conference of the PHM Society, Vol. 5, 2013.
[34] A. M. Hasofer, N. C. Lind, An exact and invarient first order reliability format, Journal of Engineering Mechanics 100 (1) (1974) 111–121.
[35] R. Rackwitz, B. Flessler, Structural reliability under combined random load sequences, Computers & Structures 9 (5) (1978) 489–494.
[36] J. Lim, B. Lee, A semi-single-loop method using approximation of most probable point for reliability-based design optimization, Structural and Multidisciplinary Optimization 53 (2016) 745–757.
[37] J. Ji, C. Zhang, Y. Gao, J. Kodikara, Effect of 2d spatial variability on slope reliability: a simplified FORM analysis, Geoscience Frontiers 9 (6) (2018) 1631–1638.
[38] M. Khalessi, Y.-T. Wu, T. Torng, Most-probable-point-locus reliability method in standard normal space, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 7460, American Society of Mechanical Engineers, 1991, pp. 15–20.
[39] C. G. Bucher, U. Bourgund, A fast and efficient response surface approach for structural reliability problems, Structural Safety 7 (1) (1990) 57–66.
[40] M. Rashki, M. Miri, M. A. Moghaddam, A new efficient simulation method to approximate the probability of failure and most probable point, Structural Safety 39 (2012) 22–29.
[41] S.-P. Zhu, B. Keshtegar, S. Chakraborty, N.-T. Trung, Novel probabilistic model for searching most probable point in structural reliability analysis, Computer Methods in Applied Mechanics and Engineering 366 (2020) 113027.
[42] C. Zhong, M. Wang, C. Dang, W. Ke, S. Guo, First-order reliability method based on harris hawks optimization for high-dimensional reliability analysis, Structural and Multidisciplinary Optimization 62 (2020) 1951–1968.
[43] M. Rashki, M. G. Faes, No-free-lunch theorems for reliability analysis, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 9 (3) (2023) 04023019.

[44] P. Hennig, M. A. Osborne, H. P. Kersting, Probabilistic Numerics: Computation as Machine Learning, Cambridge University Press, 2022.
[45] P. I. Frazier, A tutorial on bayesian optimization, arXiv preprint arXiv:1807.02811 (2018).
[46] P. Wei, Y. Zheng, J. Fu, Y. Xu, W. Gao, An expected integrated error reduction function for accelerating bayesian active learning of failure probability, Reliability Engineering & System Safety 231 (2023) 108971.
[47] C. Dang, P. Wei, M. G. Faes, M. A. Valdebenito, M. Beer, Parallel adaptive bayesian quadrature for rare event estimation, Reliability Engineering & System Safety 225 (2022) 108621.
[48] F. Tronarp, S. Särkkä, P. Hennig, Bayesian ode solvers: the maximum a posteriori estimate, Statistics and Computing 31 (3) (2021) 23.
[49] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, S. Venkatesh, Bayesian optimization for adaptive experimental design: A review, IEEE Access 8 (2020) 13937–13948.
[50] X. Wang, Y. Jin, S. Schmitt, M. Olhofer, Recent advances in bayesian optimization, arXiv preprint arXiv:2206.03301 (2022).
[51] R. Lebrun, A. Dutfoy, Do Rosenblatt and Nataf isoprobabilistic transformations really differ?, Probabilistic Engineering Mechanics 24 (4) (2009) 577–584.
[52] C. E. Rasmussen, C. Williams, Gaussian processes for machine learning, vol. 1, MIT press 39 (2006) 40–43.
[53] H. J. Kushner, A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise, Journal of Basic Engineering 86 (1) (1964) 97–106.
[54] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization 13 (4) (1998) 455.
[55] D. Huang, T. T. Allen, W. I. Notz, N. Zeng, Global optimization of stochastic black-box systems via sequential kriging meta-models, Journal of Global Optimization 34 (3) (2006) 441–466.
[56] P. I. Frazier, W. B. Powell, S. Dayanik, A knowledge-gradient policy for sequential information collection, SIAM Journal on Control and Optimization 47 (5) (2008) 2410–2439.
[57] P. Hennig, C. J. Schuler, Entropy search for information-efficient global optimization., Journal of Machine Learning Research 13 (6) (2012) 1809–1837.
[58] J. M. Hernández-Lobato, M. W. Hoffman, Z. Ghahramani, Predictive entropy search for efficient global optimization of black-box functions, Advances in neural information processing systems 27 (2014).
[59] B. Letham, B. Karrer, G. Ottoni, E. Bakshy, Constrained bayesian optimization with noisy experiments, Bayesian Analysis 13 (2) (2019) 495–519.
[60] R.-R. Griffiths, J. M. Hernández-Lobato, Constrained bayesian optimization for automatic chemical design using variational autoencoders, Chemical Science 11 (2) (2020) 577–586.
[61] S. Ariafar, J. Coll-Font, D. H. Brooks, J. G. Dy, ADMMBO: Bayesian optimization with unknown constraints using ADMM., Journal of Machine Learning Research 20 (123) (2019) 1–26.
[62] V. Picheny, R. B. Gramacy, S. Wild, S. Le Digabel, Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian, Advances in neural information processing systems 29 (2016).
[63] R. B. Gramacy, G. A. Gray, S. Le Digabel, H. K. Lee, P. Ranjan, G. Wells, S. M. Wild, Modeling an augmented lagrangian for blackbox constrained optimization, Technometrics 58 (1) (2016) 1–11.
[64] S.-K. Au, J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probabilistic Engineering Mechanics 16 (4) (2001) 263–277.
[65] S. K. Au, Probabilistic failure analysis by importance sampling markov chain simulation, Journal of Engineering Mechanics 130 (3) (2004) 303–311.
[66] P. Wei, C. Tang, Y. Yang, Structural reliability and reliability sensitivity analysis of extremely rare failure events by combining sampling and surrogate model methods, Proceedings of the Institution of Mechanical Engineers, Part O: Journal of risk and reliability 233 (6) (2019) 943–957.
[67] A. J. Torii, R. H. Lopez, Reliability analysis of water distribution networks using the adaptive response surface approach, Journal of Hydraulic Engineering 138 (3) (2012) 227–236.
[68] E. Vazquez, J. Bect, Convergence properties of the expected improvement algorithm with fixed mean and covariance functions, Journal of Statistical Planning and inference 140 (11) (2010) 3088–3095.
[69] A. D. Bull, Convergence rates of efficient global optimization algorithms., Journal of Machine Learning Research 12 (10) (2011) 2879–2904.
[70] P. Wei, X. Zhang, M. Beer, Adaptive experiment design for probabilistic integration, Computer Methods in Applied Mechanics and Engineering 365 (2020) 113035.
[71] R. Tripathy, I. Bilionis, M. Gonzalez, Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation, Journal of Computational Physics 321 (2016) 191–223.