

A CROSS-ENTROPY METHOD ACCELERATED DERIVATIVE-FREE RBDO ALGORITHM

Tian Gao¹ & Jinglai Li^{2,*}

¹Department of Mathematics, Shanghai Jiao Tong University, Shanghai 200240, China

²Institute of Natural Sciences, Department of Mathematics, and MOE Key Laboratory of Scientific and Engineering Computing, Shanghai Jiao Tong University, Shanghai 200240, China

*Address all correspondence to: Jinglai Li, E-mail: jinglaili@sjtu.edu.cn

Original Manuscript Submitted: 5/26/2016; Final Draft Received: 10/5/2016

Many engineering applications require optimization of the system performance subject to reliability constraints, which are commonly referred to as the reliability based design and optimization (RBDO) problems. In this work we propose a derivative-free algorithm to solve the RBDO problems. In particular, we focus on the type of RBDO problems where the objective function is deterministic and easy to evaluate, whereas the reliability constraints involve very small failure probabilities. The algorithm consists of solving a set of subproblems, in which simple surrogate models of the reliability constraints are constructed and used in solving the subproblems. Moreover, we employ a cross-entropy (CE) method with sample reweighting to evaluate the rare failure probabilities, which constructs the surrogate for the reliability constraints by performing only a single full CE simulation in each iteration. Finally we demonstrate the performance of the proposed method with both academic and practical examples.

KEY WORDS: uncertainty quantification, stochastic optimization, stochastic sensitivity analysis, Monte Carlo methods, computational design

1. INTRODUCTION

Real-world engineering systems are inevitably subject to various uncertainties such as material properties, geometric parameters, manufacturing tolerances, and applied loadings. Optimization of the system performance under the influence of those uncertainties is an essential task of many practical engineering design problems. A particularly important problem in this setting, known as the reliability based design and optimization (RBDO), is to optimize the system performance subject to the constraint that the system reliability satisfies a prescribed requirement. Such problems have wide applications in many fields, especially structural, mechanical, and aerospace engineering [1, 2]. Developing efficient and accurate methods to solve the RBDO problems has been an active research topic in the past decades (see the reviews [1–5] and the references therein).

In a standard RBDO problem, the reliability constraint is typically formulated as that the failure probability of the system is lower than a threshold value. Existing methods for solving RBDO problems can be largely organized into three groups [4, 5]: the double loop, the single loop, and the decoupling methods. The double loop methods resemble the most natural formulation of the RBDO problems: the inner loop which estimates the failure probability is nested in the outer loop which solves the optimization problem. The single loop methods, first proposed in [6], replace the probabilistic constraint with an approximate, deterministic constraint, so that standard numerical optimization techniques directly apply. Another way to avoid the double loop computation is to separate the reliability analysis procedure and the optimization procedure, and information from the reliability analysis step is used at the optimization step when it is needed. As a result, the method does not perform full reliability analysis for each time the optimization

procedure arrives at a new point in the design space. Methods of this kind are referred to as the decoupling approaches, including [7, 8].

In this work we focus on the double loop approaches where the reliability constraints are estimated by simulating the underlying system, and so methods based on approximating the reliability constraints (see, e.g., [9]), are not in our scope. When applicable, the simulation based double loop methods can usually produce more accurate and reliable results than the other aforementioned techniques, as they do not make any simplification or approximation of the problem. More importantly, such methods only require the ability to evaluate the limit state function of the underlying system, which makes it particularly convenient for problems with black-box models. On the other hand, the major disadvantage of the simulation based double loop approach is the high computational cost associated with it. The computational burden arises from both the inner and the outer loops. Namely, the total computational cost depends on the number of reliability (failure probability) evaluations required by the optimization algorithm and the cost for performing each single failure probability evaluation. In the inner loop, the failure probability is typically computed with Monte Carlo (MC) simulations, which requires a rather large number of samples, and each sample involves a full scale simulation of the underlying system. Considerable efforts have been devoted to studies of methods to efficiently simulate the failure probability, which typically employ either of the following two strategies. The first is to develop advanced sampling methods (see, e.g., [10] for an overview) to obtain reliability estimates of the failure probability with a limited number of samples. The other one is to construct a computationally inexpensive surrogate for the underlying system model, and replace the true model with the surrogate in the MC simulation, e.g., [11–14]. The two types of methods can also be employed together to achieve better results [15, 16]. In this work we choose to use an advanced sampling method to accelerate the computation of failure probability, and also, we use an efficient optimization method that can solve the RBDO problems with a small number of reliability evaluations. A difficulty here is that, due to the use of MC simulations, it is very difficult to obtain the analytical expressions of the derivatives of the reliability constraints. Several works (e.g., [17–19]) alleviate the difficulty by performing stochastic sensitivity analysis with the so-called score functions (SF) [20]. Once the gradients are computed, the problem can be readily solved with a standard numerical constrained optimization technique. As will be illustrated later, a major issue of the method is that it requires a very large number of MC samples, and otherwise the gradients estimated may not be reliable.

On the other hand, a very attractive class of optimization techniques, known as the derivative-free (DF) trust-region (TR) methods [21, 22], have been developed to solve problems whose derivatives are difficult to obtain. Most DF-TR algorithms are designed to solve unconstrained optimization problems, while some constrained DF-TR algorithms have been developed as well (see, e.g., [22]). In particular, a recent work [23] proposes a DF-TR algorithm with provable convergence for noisy objective and/or constraint functions. Loosely speaking, the DF-TR methods consist of solving a set of TR subproblems in which surrogate models of the objective and/or the constraint functions are constructed and used in solving the subproblems. In particular, the surrogates are often constructed with regression or interpolation, namely, without using the derivatives of the functions. In a short note [24], we outline a DF-TR scheme for RBDO problems, and in this work we discuss the implementation details, especially when the failure probability is small. In particular, we consider the situation that the objective function is deterministic and in a simple form, and so one only needs to construct surrogate models for the reliability constraints. However, the computational cost poses a major challenge for using the DF-TR methods in RBDO problems, as constructing a surrogate model with regression or interpolation requires repeated evaluation of the reliability constraints, which is computationally intensive. Thus our algorithm employs a special adaptive importance sampling scheme known as the cross-entropy (CE) method, with sample reweighting, which only uses a *single full reliability evaluation* to construct the surrogates of the constraints in each TR iteration. With numerical examples, we demonstrate that the DF-TR methods can be more robust and efficient than the score-function based methods in certain problems. Finally we note that, from an alternative perspective, the proposed DF-TR method can be viewed as an extension of the works [25, 26], where the failure probability is approximated with an exponential function. An advantage of the present work is that the TR formulation provides an effective and rigorous framework to handle the surrogates of the reliability constraints.

The paper is organized as follows. We present our DF-TR algorithm for RBDO problems in Section 2. We describe the evaluation of reliability constraints in Section 3. We provide both mathematical and practical bench-mark examples to demonstrate the performance of the proposed algorithm in Section 4. Finally we offer some concluding remarks in Section 5.

2. THE DF-TR ALGORITHM FOR RBDO PROBLEMS

There are several different formulations of the RBDO problems. Among them, a very common form is to minimize a deterministic cost function subject to the constraint that the failure probability is lower than a threshold value, which is mathematically given as

$$\begin{aligned} & \min_{\mathbf{x} \in D} f(\mathbf{x}), \\ \text{s.t. } & c(\mathbf{x}) := \ln P(\mathbf{x}) - \ln \theta \leq 0, \end{aligned} \quad (1)$$

where \mathbf{x} is the design parameter, D is the design space, $f(\cdot)$ is the cost function, $P(\cdot)$ is the failure probability, and θ is the failure probability threshold. In many applied problems, the cost function is deterministic and easy to evaluate, while computing the probabilistic constraint is much more costly as it requires large scale MC simulations. We will discuss how to compute the failure probability in the next section, and here we simply regard it as a computationally intensive function.

Due to the high computational cost of $P(\mathbf{x})$ or equivalently $c(\mathbf{x})$, a natural idea to solve the problem is to construct a computationally efficient surrogate for the constraint $c(\mathbf{x})$, and then solve the optimization problem subject to the surrogate constraint. The trust-region methods provide a rigorous formulation of the surrogate based optimization. In what follows we present our TR based RBDO algorithm in detail.

Algorithm 1. The derivative-free trust-region RBDO algorithm

Require: $f(\mathbf{x})$, $c(\mathbf{x})$, ρ_0 , ρ_{\min} , ϵ^* , ω , α^* , K , M , δ , \mathbf{x}_0 .

Ensure: Solution \mathbf{x}_{opt} ;

```

1: Outer := 1; k := 0;
2: while Outer = 1 do
3:   Inner := 1;
4:   while Inner = 1 do
5:      $[s_k(\mathbf{x}), \rho_k] := \text{SurrogConstruct}(\mathbf{x}_k, \rho_k, \epsilon^*, \omega, M, \alpha^*)$ 
6:      $\mathbf{x}_{k+1} := \arg \min_{\mathbf{x} \in O(\mathbf{x}_k, \rho_k)} f(\mathbf{x})$ , s.t.  $s_k(\mathbf{x}) \leq 0$ ;
7:     if  $c(\mathbf{x}_{k+1}) < 0$  then
8:       Inner := 0;
9:        $\rho_k = \omega^+ \rho_k$ ;
10:    else
11:       $\rho_k := \omega^- \rho_k$ ;
12:    end if
13:  end while
14:  if  $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \rho_k$  or  $\|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)\| \leq \delta$  or  $\rho_k < \rho_{\min}$  then
15:    Outer := 0;
16:  else
17:     $\rho_{k+1} := \rho_k$ ;
18:     $k := k + 1$ ;
19:  end if
20: end while
21:  $\mathbf{x}_{\text{opt}} := \mathbf{x}_k$ ;

```

The TR methods are originally developed to solve unconstrained optimization problems, but have been gradually applied to constrained problems as well. The TR methods start from an initial point \mathbf{x}_0 and find a critical point by computing a series of intermediate points $\{\mathbf{x}_k\}_{k \in \mathbb{N}}$. Specifically, suppose the current point is \mathbf{x}_k ; to compute the next point, the algorithms solve a TR subproblem in which surrogates of the objective function and the constraints are constructed and used in a neighborhood of \mathbf{x}_k . This neighborhood of \mathbf{x}_k is known as the trust-region and the size of it is adjusted in a way that the surrogate models are sufficiently accurate in it. In our problem, as has been mentioned, the objective function is of simple form, and we only need to construct the surrogate for the constraint function. As a

result, in our RBDO problems, the TR subproblem at iteration k becomes

$$\begin{aligned} & \min_{\mathbf{x} \in D} f(\mathbf{x}), \\ \text{s.t. } & s_k(\mathbf{x}) \leq 0, \quad \text{and} \quad \|\mathbf{x} - \mathbf{x}_k\| \leq \rho_k, \end{aligned} \quad (2)$$

where $s_k(\mathbf{x})$ is the surrogate model of $c(\mathbf{x})$, and ρ_k is the radius of the TR of \mathbf{x}_k . In what follows we use the notation $O(\mathbf{x}_c, \rho^*) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_c\| \leq \rho^*\}$.

In the standard TR algorithms, the surrogate models are determined from the derivatives of the constraint function, which are not available in our problems. Here in the derivative-free framework, one first writes the surrogate model as a linear combination of a set of basis functions and then determines the coefficients with either regression or interpolation. There are many different choices for the surrogate models, including polynomial response surfaces, radial basis functions, etc. [22]. Here we write the surrogate in a rather generic form:

$$s(\mathbf{x}) = \sum_{l=1}^L a_l b_l(\mathbf{x}), \quad (3)$$

where $\{b_l(\mathbf{x})\}_{l=1}^L$ are a set of basis functions and $\mathbf{a} = (a_1, \dots, a_L)^T$ is the vector collecting all the coefficients.

Before discussing the construction of the surrogate models, we first present our DF-TR based method for solving the RBDO problems. We describe the complete procedure in Algorithm 1. A key step in a TR algorithm is to adjust the radius of the trust-region in each step. In this respect our algorithm follows the procedure given in [23], with certain modifications. The first modification is that the original algorithm adjusts the TR radius based on the quality of surrogates for both the objective and the constraint functions, while in our algorithm, the radius is only adjusted according to the constraint function. The trust-region subproblem Eq. (2) can be solved with any usual constrained optimization technique, and in this work we choose to use the sequential quadratic programming (SQP) method. The algorithm terminates when either of the following two conditions are satisfied: \mathbf{x}_{k+1} is an inner point of $O(\mathbf{x}_k, \rho_k)$, or the difference between $f(\mathbf{x}_k)$ and $f(\mathbf{x}_{k+1})$ is sufficiently small. We note that a key step of the algorithm is to evaluate the failure probabilities and construct a surrogate for it in each TR, which will be discussed in the next section.

3. FAILURE PROBABILITY ESTIMATION

In this section, we discuss how to evaluate the reliability constraint function $c(\mathbf{x})$. Recall that we consider the constraint in the form of $c(\mathbf{x}) = \ln P(\mathbf{x}) - \ln \theta \leq 0$, and thus the key in evaluating $c(\mathbf{x})$ is to estimate the failure probability $P(\mathbf{x})$. As is mentioned in Section 1, estimating the failure probability is computationally intensive and requires advanced sampling techniques. Among many such efficient sampling techniques, we adopt the cross-entropy (CE) method [27, 28], an adaptive importance sampling (IS) scheme, to estimate the failure probability. The CE method is a quite general IS strategy and has been under intense development since its introduction. It has been applied to many practical problems, including structural safety [29], communication systems [30, 31], etc.

3.1 MC Estimation and Importance Sampling

Here we describe the failure probability estimation problem in a general setting. We consider a probabilistic model where \mathbf{z} is a d_z -dimensional random variable that represents the uncertainty in the model and the system failure is defined with a real-valued function $g(\mathbf{z}; \mathbf{x})$ of both the random variable \mathbf{z} and the design variable \mathbf{x} , which is known as the *limit state function*. Without causing any ambiguity, we omit the argument \mathbf{x} in g for conciseness. The event of failure is defined as $g(\mathbf{z}) < 0$ and as a result the failure probability is

$$\mathbb{P}(g(\mathbf{z}) < 0) = \int_{\{\mathbf{z} \in R^{d_z} \mid g(\mathbf{z}) < 0\}} q(\mathbf{z}) d\mathbf{z} = \int_{\mathbf{z} \in R^{d_z}} \mathbb{I}_{\{g < 0\}} q(\mathbf{z}) d\mathbf{z}, \quad (4)$$

where $\mathbb{I}_{\{g < \gamma\}}(\mathbf{z})$ is an indicator function:

$$\mathbb{I}_{\{g < \gamma\}} = \begin{cases} 1 & \text{if } g(\mathbf{z}) < \gamma \\ 0 & \text{if } g(\mathbf{z}) \geq \gamma \end{cases},$$

and $q(\mathbf{z})$ is the probability density function (PDF) of \mathbf{z} . In what follows we shall omit the integration domain when it is simply R^{d_z} . This is a general definition for failure probability, which is used widely in many disciplines involving reliability analysis and risk management. P can be computed with the standard Monte Carlo estimation:

$$\hat{P}_{\text{MC}} = \frac{1}{N} \sum_{n=1}^N \mathbb{I}_{\{g < 0\}}(\mathbf{z}^{(n)}), \quad (5)$$

where samples $\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N)}$ are drawn from distribution $q(\mathbf{z})$.

In practice, however, many engineering systems require high reliability, namely, the failure probability $P \ll 1$. In this case, MC requires a large number of samples to produce a reliable estimate of P . For example, for $P \approx 10^{-6}$, MC simulation requires 10^8 samples to obtain an estimate with 10% coefficient of variation. On the other hand, in almost all practical cases, the limit state function $g(\mathbf{z})$ does not admit analytical expression and has to be evaluated through expensive computer simulations, which renders the crucial MC estimation of the failure probability prohibitive.

The technique of IS is an effective way to accelerate the standard MC simulation. The idea of IS is that, instead of sampling directly from the density $q(\mathbf{z})$ directly, one introduces a biasing distribution $p(\mathbf{z})$ and rewrites Eq. (4) as

$$P_f = \int \mathbb{I}_{\{g < 0\}}(\mathbf{z}) \frac{q(\mathbf{z})}{p(\mathbf{z})} p(\mathbf{z}) d\mathbf{z} = \int \mathbb{I}_{\{g < 0\}}(\mathbf{z}) W(\mathbf{z}) p(\mathbf{z}) d\mathbf{z}, \quad (6)$$

where $W(\mathbf{z}) = q(\mathbf{z})/p(\mathbf{z})$ is the likelihood ratio. One then has the IS estimator for P :

$$P_{\text{IS}} = \frac{1}{N} \sum_{n=1}^N \mathbb{I}_{\{g < 0\}}(\mathbf{z}^{(n)}) W(\mathbf{z}^{(n)}), \quad (7)$$

where samples $\{\mathbf{z}^{(n)}\}$ are drawn from the biasing distribution $p(\mathbf{z})$. The key in designing an effective IS method is to construct a “good” biasing distribution $p(\mathbf{z})$ so that more samples will land in the failure domain, while the contributions of the samples will be adjusted by the likelihood ratio. The successful design of the biasing distribution is, however, not a trivial task, and considerable efforts have been devoted to this issue.

3.2 The Cross-Entropy Method

There exist several different approaches for IS, some of which are problem dependent. Here we employ the cross-entropy (CE) method, which adaptively searches for a good biasing distribution without requiring any specific information of the underlying problem. For detailed overviews of the method, see [27, 28]. Here we provide a brief introduction to the CE method for our specific purposes.

It is well known that the “optimal” biasing distribution exists:

$$p^*(\mathbf{z}) = \frac{1}{P} \mathbb{I}_{\{g < 0\}}(\mathbf{z}) q(\mathbf{z}). \quad (8)$$

It will result in zero variance in Eq. (6) and therefore require only a single sample to evaluate the integral. The problem is that it depends on the unknown P_f and thus cannot be of direct use. Nevertheless, one can find a good biasing distribution by requiring that it is “close” to the optimal one. The key ingredient of the CE method is the use of cross-entropy, also known as Kullback-Leibler divergence (KLD) [27], between two probability distributions to optimally determine the biasing distribution in Eq. (6). For two distributions p_1 and p_2 , the CE, or KLD, takes the following form:

$$\mathcal{D}(p_1, p_2) = \mathbb{E}_{p_1} \left[\ln \frac{p_1(\mathbf{z})}{p_2(\mathbf{z})} \right] = \int p_1(\mathbf{z}) \ln \frac{p_1(\mathbf{z})}{p_2(\mathbf{z})} d\mathbf{z}. \quad (9)$$

It is easy to verify that $\mathcal{D}(p_1, p_2) \geq 0$, where equality is achieved when $p_1 = p_2$ a.e.

To be specific, in the CE method, the biasing distribution $p(\mathbf{z})$ in Eq. (6) is determined by requiring $p(\mathbf{z})$ to be “close” to $p^*(\mathbf{z})$, in terms of the CE distance. The CE distance from p^* to p can be written in two parts:

$$\mathcal{D}(p^*, p) = \int p^*(\mathbf{z}) \ln p^*(\mathbf{z}) d\mathbf{z} - \int p^*(\mathbf{z}) \ln p(\mathbf{z}) d\mathbf{z}. \quad (10)$$

Minimizing Eq. (10) is equivalent to maximizing the second integral on the right-hand side. In practice, one often searches for the biasing distribution from a parametrized family of distribution $\{p(\mathbf{z}; \mathbf{v})\}_{\mathbf{v} \in V}$, where V is the space of the parameter \mathbf{v} . By using Eq. (8), the optimization problem Eq. (10) becomes solving for

$$\max_{\mathbf{v}} \int \mathbb{I}_{\{g < 0\}}(\mathbf{z}) \ln(p(\mathbf{z}; \mathbf{v})) q(\mathbf{z}) d\mathbf{z}.$$

However, the same difficulty caused by the rareness of $\{g < 0\}$ in term of the distribution $q(\mathbf{z})$ persists. To circumvent the difficulty, the idea of IS is again utilized. By adopting another biasing distribution $p(\mathbf{z}; \mathbf{w})$ we obtain the following optimization problem:

$$\max_{\mathbf{v}} \int \mathbb{I}_{\{g < 0\}}(\mathbf{z}) W(\mathbf{z}; \mathbf{w}) \ln(p(\mathbf{z}; \mathbf{v})) p(\mathbf{z}; \mathbf{w}) d\mathbf{z}, \quad (11)$$

where $W(\mathbf{z}; \mathbf{w}) = q(\mathbf{z})/p(\mathbf{z}; \mathbf{w})$ is the likelihood ratio. In practice, the stochastic counterpart of Eq. (11) is usually employed. Let $\{\mathbf{z}^{(i)}\}_{i=1}^M$ be samples drawn from the distribution $p(\mathbf{z}; \mathbf{w})$; we solve

$$\hat{\mathbf{v}} = \operatorname{argmax}_{\mathbf{v} \in V} \frac{1}{N} \sum_{n=1}^N \mathbb{I}_{\{g < 0\}}(\mathbf{z}^{(n)}) W(\mathbf{z}^{(n)}; \mathbf{w}) \ln(p(\mathbf{z}^{(n)}; \mathbf{v})). \quad (12)$$

The stochastic optimization problem Eq. (12) can be solved by a multilevel algorithm, which generates a sequence of reference parameters v_k and a sequence of decreasing levels $\gamma_k > 0$ with $k = 0, 1, 2, \dots$. These sequences are then used to define intermediate failure domains $\{g(\mathbf{z}) < \gamma_k\}$ under distribution $p(\mathbf{z}; \mathbf{v}_k)$, instead of $p(\mathbf{z}; \mathbf{w})$ as in Eq. (11), whose probability is not rare. The iteration terminates when the level γ_k reaches zero, resulting in the original failure domain, and the converged v_k defines the biasing distribution $p(\mathbf{z}; \mathbf{v})$ in the IS integral Eq. (6). The complete multilevel algorithm and the convergence analysis can be found in [27]. We note that a major issue in the implementation of the CE method is to choose an appropriate parametrized family for the biasing distribution. Here we follow the work of Wang and Zhou [32], using Gaussian mixtures to represent the biasing distributions, mainly because they have high flexibility and the optimal parameter values can be efficiently computed. The detailed algorithm to update the biasing distributions is well documented in [32] and we shall not repeat it here.

3.3 Recycling the Limit State Function Evaluations

Recall that in Algorithm 1, we need to evaluate the failure probability at a number of design points in the TR to construct the surrogate function. Since each evaluation requires a full CE sampling procedure, the total computational cost can be very high. To improve the efficiency, we present a reweighting approach, which allows one to obtain the failure probability values at all design points with one full CE based reliability analysis. Following the formulations in [17], we assume that the design parameters \mathbf{x} only affect the reliability via the distribution of the random parameters \mathbf{z} , while the limit state function is independent of the design parameters, namely,

$$P(\mathbf{x}) = \int \mathbb{I}_{\{g < 0\}}(\mathbf{z}) q(\mathbf{z}; \mathbf{x}) d\mathbf{z}, \quad (13)$$

where $q(\mathbf{z}; \mathbf{x})$ is the PDF of \mathbf{z} depending on the design parameter \mathbf{x} . Suppose we have performed a CE estimation of the failure probability at the center of the TR, \mathbf{x}_c , obtaining a biasing distribution $p(\mathbf{z})$ and a set of samples from it: $\{\mathbf{z}_i, g(\mathbf{z}^{(n)}), W(\mathbf{z}^{(n)})\}$, where $W(\mathbf{z}) = q(\mathbf{z}; \mathbf{x}_c)/p(\mathbf{z})$. For any point \mathbf{x} in the TR, we can rewrite Eq. (13) as

$$P(\mathbf{x}) = \int \mathbb{I}_{\{g < 0\}}(\mathbf{z}) \frac{q(\mathbf{z}; \mathbf{x})}{q(\mathbf{z}; \mathbf{x}_c)} q(\mathbf{z}; \mathbf{x}_c) d\mathbf{z} = \int \mathbb{I}_{\{g < 0\}}(\mathbf{x}) W(\mathbf{z}) r(\mathbf{z}) p(\mathbf{z}) d\mathbf{z},$$

where $r(\mathbf{z}) = q(\mathbf{z}; \mathbf{x})/q(\mathbf{z}; \mathbf{x}_c)$. It follows immediately that $P(\mathbf{x})$ can be estimated as

$$\hat{P}(\mathbf{x}) = \sum_{n=1}^N \mathbb{I}_{\{g < 0\}}(\mathbf{z}^{(n)}) W(\mathbf{z}^{(n)}) r(\mathbf{z}^{(n)}), \tag{14}$$

i.e., by simply assigning new weights $r(\mathbf{z})$ to the samples generated in the evaluation of $P(\mathbf{x}_c)$. In this method, only the computation of $P(\mathbf{x}_c)$ involves the evaluations of the limit state function $g(\cdot)$, which is referred to as a *full reliability evaluation*. Note that, if the TR region is large, it is possible that the biasing distribution obtained at \mathbf{x}_c is not good for other locations. Thus we need to introduce step size restrictions on the TR to ensure that $p(\mathbf{z})$ is a good biasing distribution for any point \mathbf{x} in the TR. This step is included in our algorithm to construct the constraint surrogates, which is discussed in the next section.

3.4 Construction of the Surrogate Models

We now discuss the construction of the surrogates, which is a critical step in the optimization algorithm. The general form of the surrogate has been given by Eq. (3), while the basis functions are not specified. Now we choose to use the popular *quadratic polynomial* surrogates, while noting that the proposed algorithm does not depend on any particular type of surrogates. As has been mentioned, the coefficients \mathbf{a} in Eq. (3) can be determined by either interpolation or regression.

The surrogate construction procedure highlights a difference between our algorithm and the standard DF-TR methods. In the standard DF-TR algorithms, the surrogate models are required to be fully linear or fully quadratic [22]. Imposing such conditions is very difficult in RBDO problems as the failure probability is evaluated with sampling methods. Thus here we simply require that the error between the surrogate and the true constraint function is bounded in the TR: for a given $\epsilon > 0$ and a TR $O(\mathbf{x}_c, \rho)$, $|s(\mathbf{x}) - c(\mathbf{x})| \leq \epsilon$ for any $\mathbf{x} \in O(\mathbf{x}_c, \rho)$.

Now, we propose a scheme to construct the TR surrogate with a bounded error, which is described in Algorithm 2. Simply speaking, the algorithm constructs the quadratic regression and examines whether the resulting surrogate satisfies the error bound condition, and if not, the algorithm contracts the TR and repeats. Some remarks regarding the implementation details are listed in order:

1. The number of sampling points M is determined as [22]

$$M = \sqrt{d_x}(d_x + 1)(d_x + 2)/2,$$

where d_x is the dimension of \mathbf{x} .

2. In Line 12, we estimate the approximation error with the leave-one-out cross validation method. Namely, let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ and $Y = \{y_1, \dots, y_M\}$ and $y_m = c(\mathbf{x}_m)$ for $m = 1 \dots M$. Let $X_-^m = \{\mathbf{x}_1, \dots, \mathbf{x}_{m-1}, \mathbf{x}_{m+1}, \dots, \mathbf{x}_M\}$ and $Y_-^m = \{y_1, \dots, y_{m-1}, y_{m+1}, \dots, y_M\}$. Let $s^m(\mathbf{x})$ be the surrogate model based on data (X_-^m, Y_-^m) and the approximation error ϵ is estimated by

$$\epsilon = \max\{|c(\mathbf{x}_m) - s^m(\mathbf{x}_m)|\}_{m=1}^M.$$

3. As is discussed in the previous section, we need to reduce the TR size when the biasing distribution $p(\mathbf{z})$ obtained at \mathbf{x}_c is not good for some points in the TR. In particular, we compute the coefficient of variation (CV) for the failure probability computed at each point \mathbf{x} drawn in the TR:

$$\alpha = \frac{\left(\sum_{n=1}^N \mathbb{I}_{\{g < 0\}}(\mathbf{z}^{(n)}) W^2(\mathbf{z}^{(n)}) r^2(\mathbf{z}^{(n)}) - \hat{P}^2\right)^{1/2}}{\hat{P}}, \tag{15}$$

where \hat{P} is computed with Eq. (14). We then reduce the TR radius if the largest CV exceeds a threshold value.

Algorithm 2. $[s(\cdot), \rho] = \text{SurrogConstruct}(\mathbf{x}_c, \rho_{\max}, \epsilon^*, \omega, M, \alpha^*)$

```

1: let  $\rho := \rho_{\max}$ ; LOOP := 1;
2: perform CE simulation at  $\mathbf{x}_c$ , obtaining the biasing distribution  $p(\mathbf{z})$  and the set  $\{g(\mathbf{z}^{(1)}), \dots, g(\mathbf{z}^{(N)})\}$  where
    $x_1, \dots, x_N$  are drawn from  $p(\mathbf{z})$ ;
3: while LOOP = 1 do
4:   randomly generate  $M - 1$  points in  $O(\mathbf{x}, \rho)$ :  $\{\mathbf{x}_m \in O(\mathbf{x}, \rho)\}_{m=1}^{M-1}$ ;
5:   let  $\mathbf{x}_M = \mathbf{x}_c$ ;
6:   compute the CV of the failure probability estimated at each  $\mathbf{x}_m$  for  $m = 1 \dots M - 1$  with Eq. (15), and denote
   the results as  $\{\alpha_1, \dots, \alpha_{M-1}\}$ ;
7:   if  $\max\{\alpha_1, \dots, \alpha_{M-1}\} > \alpha^*$  then
8:      $\rho := \omega\rho$ ;
9:   else
10:    let  $y_m = c(\mathbf{x}_m)$  for  $m = 1 \dots M$ ;
11:    compute the surrogate  $s(\mathbf{x})$  using data set  $\{(\mathbf{x}_m, y_m)\}_{m=1}^M$ ;
12:    estimate the approximation error bound  $\epsilon$  of  $s(\cdot)$  with leave-one-out cross validation;
13:    if  $\epsilon < \epsilon^*$  then
14:      LOOP := 0;
15:    else
16:       $\rho := \omega\rho$ ;
17:    end if
18:  end if
19: end while
20: return  $s(\cdot)$  and  $\rho$ .

```

4. NUMERICAL EXAMPLES

In this section, we provide one mathematical and three practical examples to illustrate the performance of the proposed DF-TR algorithm.

4.1 A Two-Dimensional Example

We first consider a two-dimensional mathematical example:

$$\begin{aligned}
 \min_{(x \in R^+, r \in R^+)} \quad & f(x, r) = ax^2 + \frac{b}{r}, \\
 \text{s.t.} \quad & \mathbb{P}[(z_1 - z_x)^2 + (z_2 - z_x)^2 \leq z_r] \leq P_{\max},
 \end{aligned} \tag{16}$$

where z_1, z_2, z_x , and z_r are four independent Gaussian random variables. In particular, we take $z_1, z_2 \sim N(0, 1)$, $z_x \sim N(x, \sigma_x^2)$, and $z_r \sim N(r, \sigma_r^2)$. The geometric interpretation of the problem is that the event of failure is defined as the random variable (z_1, z_2) falls inside the disk $\{(z_1 - z_x)^2 + (z_2 - z_x)^2 \leq z_r\}$, and by increasing the ‘‘cost,’’ one can reduce the failure probability by either moving the center of the disk apart from the means of random variables (z_1, z_2) , or reducing the radius of the disk. In our numerical tests, we choose $a = 2$, $b = 1$, $\sigma_x = 10^{-2}$, and $\sigma_r = 10^{-3}$.

We test two different cases: $P_{\max} = 0.1$ and $P_{\max} = 10^{-6}$. We solve the problem with our DF-TR algorithm, where the failure probability is with the CE method. The algorithm parameter values are given in Table 1, and are used in all the numerical examples. In the CE method, we use the multilevel algorithm, where the biasing distribution is chosen to be Gaussian, 10^4 samples are generated in each iteration, and the 10% samples with the lowest limit state function values are used to evaluate the parameter values for the next iteration. As a comparison, we also solved the problem with a standard active set method [33] for constrained optimizations, where the gradients are computed

TABLE 1: The parameter values of the DF-TR algorithm

Parameter	ρ_0	ϵ^*	ω^-	ω^+	M	δ
Value	0.1	0.10	0.9	1.1	20	10^{-4}

with the score function (SF) method (see Appendix A). We note that, in the estimation of the gradients with the score function method, we also employ the multilevel CE method with 10^5 samples in each iteration, which renders the gradient estimation more expensive than the failure probability evaluation. Since both methods employ sampling procedure subject to random errors, to take that into account, we repeatedly solve the problem with both methods 100 times and summarize the results of both methods in Table 2. Specifically, we compare the feasibility rate, defined as the percentage of the feasible solutions obtained,* the average (over all feasible solutions) minimal cost, and the average number of function evaluations.† The results indicate that our DF-TR method is more robust (higher feasibility rate) and more efficient (fewer function evaluations) than the SF based optimization. Finally we note that, to valid the CE method, we have also performed the DF-TR optimization with standard MC (10^5 samples) for the case $P < 0.1$, and the results are very similar to those of the CE based DF-TR and so are omitted here.

4.2 Cantilever Beam

We now consider a cantilever beam problem as illustrated in Fig. 1, with width W , height T , length L , and subject to transverse load Y and horizontal load X . This is a well adopted bench-mark problem in engineering, where the system failure is defined as the maximum deflection exceeding a threshold value:

$$g = D_o - \frac{4L^3}{EWT} \sqrt{\left(\frac{Y}{T}\right)^2 + \left(\frac{X}{W}\right)^2},$$

where D_o is the deflection threshold and E is the Young’s modulus. In this example we assume the beam length L is fixed to be 100 and $D_o = 6$. The random variables are as follows: the elastic module $E \sim \mathcal{N}(29 \times 10^6, (1.45 \times 10^6)^2)$, external loads $X \sim \mathcal{N}(500, 25^2)$, and $Y \sim \mathcal{N}(500, 25^2)$, and the actual beam width $W \sim N(w, \sigma^2)$ and height $T \sim N(t, \sigma^2)$, respectively. The mean width w and the mean height t are design variables, and our goal is to minimize the construction cost:

$$f(w, t) = wt, \tag{17}$$

subject to the associated failure probability being smaller than 10^{-6} . To test how the algorithm is sensitive to the value of the variance σ^2 , we solve the problem with $\sigma = 10^{-2}$ and $\sigma = 10^{-3}$.

We solve the problem with the DF-TR method, where the algorithm parameter values are also given in Table 1, and in the CE procedure we use the same configurations as those in example 1. To illustrate the behavior of the algorithm, we plot the path from the initial guess to the converged solution for $\sigma = 10^{-3}$ in Fig. 2(a), and the function value against the number of iterations in Fig. 2(b). The figures demonstrate two main properties of the DF-TR method:

TABLE 2: Comparison of the performance of DF-TR and SF for example 1

		Feasible Rate	Minimal Cost	Full Evaluations
$P < 0.1$	SF	88%	2.10	137
	DF-TR	100%	2.15	13
$P < 10^{-6}$	SF	79%	24.97	152
	DF-TR	100%	24.94	50

*Throughout this work, we call a solution feasible if the resulting failure probability is smaller than $1.1P_{\max}$, i.e., 10% error being allowed.

†In the DR-TR method, a function evaluation is referred to as a full evaluation of the reliability constraint, and in the SF based method, it is referred to as an evaluation of the constraint or the gradient of it.

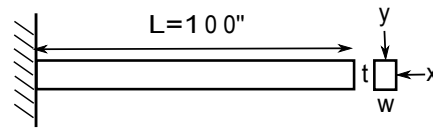


FIG. 1: The schematic diagram of the cantilever beam.

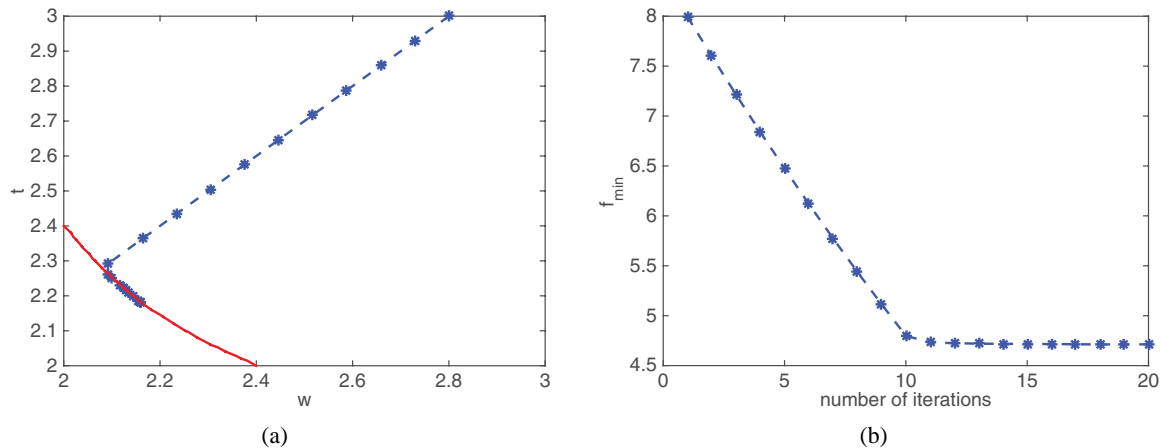


FIG. 2: (a) the path from the initial guess to the final solution (the solid line is the boundary of the feasible region, also computed with the CE method). (b) the function value plotted against the number of iterations and the dashed line shows the cost associated with the exact solution.

during the iterations, it does not violate the reliability constraint or increase the cost function value. In the SF based method, we also use 10^5 samples in each iteration to compute the gradients. We solve the problem with each method 100 times and compare their performance in Table 3. The results indicate that, for $\sigma = 10^{-2}$, both methods perform very well and are able to find feasible solutions in all trials. However, for $\sigma = 10^{-3}$, the two methods perform very differently: the DF-TR method still finds feasible solutions in all the trials, while the SF based method fails in all of them. The SF based method does not perform well in our test, because with a limited number of samples, the gradient estimates are not reliable. We note that the performance of the SF method can be improved by increasing the number of samples, which certainly results in much higher computational cost.

4.3 Vehicle Side Impact

Our third example is the vehicle side impact problem used as a bench-mark example in several works [2, 8, 18]. For side impact protection, the vehicle design must satisfy certain safety requirements which can be expressed as the reliability constraints. The goal of the problem is then to minimize the total weight, subject to 10 such reliability constraints of four types: head injury criterion (HIC) constraint, abdomen load, pubic symphysis force, viscous criteria (VC), and rib deflections (including upper, middle, and lower). In our test, we assume that the problem has 11 random parameters denoted as $\mathbf{z} = (z_1, \dots, z_{11})$, which are independent and follow Gaussian distributions. The

TABLE 3: Comparison of the performance of DF-TR and SF for example 2

		Feasible Rate	Minimal Cost	Full Evaluations
$\sigma = 10^{-2}$	SF	100%	4.73	143
	DF-TR	100%	4.71	46
$\sigma = 10^{-3}$	SF	0%	-	-
	DF-TR	100%	4.71	48

means of z_1, \dots, z_7 are the design parameters and those of z_8, \dots, z_{11} are fixed. The meaning and the specifications of the random and design parameters are given in Table 4. The mathematical formulation of this RBDO problem is described as

$$\begin{aligned}
 \min_{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U} \quad & \text{Weight}(\mathbf{x}) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7 \\
 \text{s.t.} \quad & \mathbb{P}[\text{abdomen load} > 1.0 \text{ kN}] \leq P_{\max} \\
 & \mathbb{P}[\text{upper/middle/lower VC} > 0.32 \text{ mls}] \leq P_{\max} \\
 & \mathbb{P}[\text{upper/middle/lower rib deflection} > 32 \text{ mm}] \leq P_{\max} \\
 & \mathbb{P}[\text{pubic symphysis force } F > 4.0 \text{ kN}] \leq P_{\max} \\
 & \mathbb{P}[\text{velocity of B-pillar at middle point} > 9.9 \text{ mm/ms}] \leq P_{\max} \\
 & \mathbb{P}[\text{velocity of front door at B-pillar} > 15.7 \text{ mm/ms}] \leq P_{\max}.
 \end{aligned} \tag{18}$$

The specific forms of the limit state functions are

$$\begin{aligned}
 \text{Load abdomen} &= 1.16 - 0.3717z_2z_4 - 0.00931z_2z_{10} - 0.484z_3z_9 + 0.01343z_6z_{10}, \\
 \text{Deflection rib}_u &= 28.98 + 3.818z_3 - 4.2z_1z_2 + 0.0207z_5z_{10} + 6.63z_6z_9 - 7.73z_7z_8 + 0.32z_9z_{10}, \\
 \text{Deflection rib}_m &= 33.86 + 2.95z_3 + 0.1792z_{10} - 5.057z_1z_2 - 11z_2z_8 - 0.0215z_5z_{10} - 9.98z_7z_8 \\
 &\quad + 22z_8z_9, \\
 \text{Deflection rib}_l &= 46.36 - 9.9z_2 - 12.9z_1z_8 + 0.1107z_3z_{10}, \\
 \text{VC upper} &= 0.261 - 0.0159z_1z_2 - 0.188z_1z_8 - 0.019z_2z_7 + 0.0144z_3z_5 + 0.0008757z_5z_{10} \\
 &\quad + 0.08045z_6z_9 + 0.00139z_8z_{11} + 0.00001575z_{10}z_{11}, \\
 \text{VC middle} &= 0.0214 + 0.00817z_5 - 0.131z_1z_8 - 0.0704z_1z_9 + 0.03099z_2z_6 - 0.018z_2z_7 \\
 &\quad + 0.0208z_3z_8 + 0.121z_3z_9 - 0.00364z_5z_6 + 0.0007715z_5z_{10} - 0.0005354z_6z_{10} \\
 &\quad + 0.00121z_8z_{11}, \\
 \text{VC lower} &= 0.74 - 0.61z_2 - 0.163z_3z_8 + 0.001232z_3z_{10} - 0.166z_7z_9 + 0.0227z_2^2, \\
 \text{Force pubic} &= 4.72 - 0.5z_4 - 0.19z_2z_3 - 0.0122z_4z_{10} + 0.009325z_6z_{10} + 0.00019z_{11}^2, \\
 \text{Velocity B-Pillar} &= 10.55 - 0.674z_1z_2 - 1.95z_2z_8 + 0.02054z_3z_{10} - 0.0198z_4z_{10} + 0.028z_6z_{10}, \\
 \text{Velocity Door} &= 16.45 - 0.489z_3z_7 - 0.843z_5z_6 + 0.0432z_9z_{10} - 0.0556z_9z_{11} - 0.000786z_{11}^2.
 \end{aligned} \tag{19}$$

TABLE 4: Random parameters in the vehicle side impact problem. The 8th to 11th random variables are not regarded as design variables

Random Variable	Std. Dev	Mean	\mathbf{x}^L	\mathbf{x}^U
z_1 (B-pillar inner)	0.030	x_1	0.500	1.500
z_2 (B-pillar reinforce)	0.030	x_2	0.500	1.500
z_3 (Floor side inner)	0.030	x_3	0.500	1.500
z_4 (Cross member)	0.030	x_4	0.500	1.500
z_5 (Door beam)	0.030	x_5	0.500	1.500
z_6 (Door belt line)	0.030	x_6	0.500	1.500
z_7 (Roof rail)	0.030	x_7	0.500	1.500
z_8 (Mat. B-pillar inner)	10^{-3}	0.345	-	-
z_9 (Mat. floor side inner)	10^{-3}	0.192	-	-
z_{10} (Barrier height)	10^{-3}	0	-	-
z_{11} (Barrier hitting)	10^{-3}	0	-	-

Here we set $P_{\max} = 10^{-3}$ and solve the problem with the proposed DF-TR method (with 10^4 samples in CE). The initial condition is chosen to be (0.5, 1.5, 0.5, 1.5, 1.5, 1.5, 1.5), and the algorithm terminates in 11 iterations, obtaining a solution: (0.50, 1.33, 0.50, 1.34, 1.38, 1.37, 1.41). We present more results in Table 5. In particular, in the table, we show the weight and the maximum failure probability associated with the design parameter values at the 1st, 5th, and 11th iterations, respectively. We also show the numbers of full reliability evaluations performed in the table. The results indicate that the proposed method can apply to this real-world RBDO problem, even for a rather small failure probability requirement ($<10^{-3}$). We have also applied the SF based optimization with 10^5 samples to this example, and the algorithm fails to converge to a feasible solution, which suggests that the SF based method requires significantly more computational efforts to compute a solution.

5. CONCLUSIONS

In summary, we have proposed a derivative-free trust-region double loop algorithm to solve the reliability based optimization problems. An important feature of the algorithm is that we employ a sample reweighting method so that the TR surrogate can be obtained by performing a single full reliability evaluation. With both mathematical and practical examples, we show that the method is more robust and efficient than the popular SF based method. We believe the proposed DF-TR algorithm can be useful in a wide range of practical engineering design problems.

We want to reinstate that the main purpose of this work is to present an optimization scheme that can solve the RBDO problem with a small number of full reliability evaluations. On the other hand, there are a number of methods that can efficiently evaluate the failure probability by constructing surrogates for the limit state functions, and these methods can be readily incorporated in the proposed DF-TR algorithm, to improve the overall efficiency. Developing such surrogate based DF-TR algorithm is not in the scope of the present work, but is an interesting problem that we plan to study in the future. Finally note that we are not able to provide a convergence analysis of the proposed method in this work, and we also hope to address this issue in a future work.

TABLE 5: The optimization results for the vehicle side impact example

Iterations	Weight	Max Failure Probability	Total Full Evaluations
1	30.7	3.4×10^{-4}	0
5	29.0	7.6×10^{-4}	5
11	28.4	9.0×10^{-4}	27

ACKNOWLEDGMENTS

The work was partially supported by the National Natural Science Foundation of China under grant no. 11301337, and by the National Basic Research Program (973 Program) of China under grant no. 2014CB744900.

REFERENCES

1. Enevoldsen, I. and Sørensen, J. D., Reliability-based optimization in structural engineering, *Struct. Saf.*, 15(3):169–196, 1994.
2. Yang, R. and Gu, L., Experience with approximate reliability-based optimization methods, *Struct. Multidiscip. Optim.*, 26(1-2):152–159, 2004.
3. Schuëller, G. I. and Jensen, H. A., Computational methods in optimization considering uncertainties—an overview, *Comput. Methods Appl. Mech. Eng.*, 198(1):2–13, 2008.
4. Valdebenito, M. A. and Schuëller, G. I., A survey on approaches for reliability-based optimization, *Struct. Multidiscip. Optim.*, 42(5):645–663, 2010.
5. Aoues, Y. and Chateaneuf, A., Benchmark study of numerical methods for reliability-based design optimization, *Struct. Multidiscip. Optim.*, 41(2):277–294, 2010.

6. Chen, X., Hasselman, T. K., and Neill, D. J., Reliability based structural design optimization for practical applications, In *Proc. of the 38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conf.*, pp. 2724–2732, 1997.
7. Tu, J., Choi, K. K., and Park, Y. H., Design potential method for robust system parameter design, *AIAA J.*, 39(4):667–677, 2001.
8. Du, X. and Chen, W., Sequential optimization and reliability assessment method for efficient probabilistic design, *J. Mech. Des.*, 126(2):225–233, 2004.
9. Youn, B. D. and Choi, K. K., Selecting probabilistic approaches for reliability-based design optimization, *AIAA J.*, 42(1):124–131, 2004.
10. Schuëller, G. and Pradlwarter, H., Benchmark study on reliability estimation in higher dimensions of structural systems—an overview, *Struct. Saf.*, 29(3):167–182, 2007.
11. Jin, R., Du, X., and Chen, W., The use of metamodeling techniques for optimization under uncertainty, *Struct. Multidiscip. Optim.*, 25(2):99–116, 2003.
12. Lee, T. H. and Jung, J. J., A sampling technique enhancing accuracy and efficiency of metamodel-based rbdo: Constraint boundary sampling, *Comput. Struct.*, 86(13):1463–1476, 2008.
13. Xiong, Y., Chen, W., and Tsui, K.-L., A new variable-fidelity optimization framework based on model fusion and objective-oriented sequential sampling, *J. Mech. Des.*, 130(11):111401, 2008.
14. Papadrakakis, M. and Lagaros, N. D., Reliability-based structural optimization using neural networks and Monte Carlo simulation, *Comput. Methods Appl. Mech. Eng.*, 191(32):3491–3507, 2002.
15. Li, J., Li, J., and Xiu, D., An efficient surrogate-based method for computing rare failure probability, *J. Comput. Phys.*, 230:8683–8697, 2011.
16. Wu, K. and Li, J., A surrogate accelerated multicanonical monte carlo method for uncertainty quantification, arXiv preprint, arXiv:1508.06700, 2015.
17. Lee, I., Choi, K. K., Noh, Y., Zhao, L., and Gorsich, D., Sampling-based stochastic sensitivity analysis using score functions for RBDO problems with correlated random variables, *J. Mech. Des.*, 133(2):021003, 2011.
18. Wang, Z. and Wang, P., A maximum confidence enhancement based sequential sampling scheme for simulation-based design, *J. Mech. Des.*, 136(2):021006, 2014.
19. Dubourg, V. and Sudret, B., Meta-model-based importance sampling for reliability sensitivity analysis, *Struct. Saf.*, 49:27–36, 2014.
20. Rubinstein, R. Y. and Shapiro, A., *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*, New York: John Wiley & Sons Inc, 1993.
21. Powell, M. J., UOBYQA: unconstrained optimization by quadratic approximation, *Math. Program.*, 92(3):555–582, 2002.
22. Conn, A. R., Scheinberg, K., and Vicente, L. N., *Introduction to derivative-free optimization*, vol. 8, Philadelphia: SIAM, 2009.
23. Augustin, F. and Marzouk, Y., NOWPAC: A provably convergent derivative-free nonlinear optimizer with path-augmented constraints, arXiv preprint, arXiv:1403.1931, 2014.
24. Gao, T. and Li, J., A derivative-free trust-region algorithm for reliability-based optimization, *Struct. Multidiscip. Optim.*, doi: 10.1007/s00158-016-1587-y, 2016.
25. Gasser, M. and Schuëller, G. I., Reliability-based optimization of structural systems, *Math. Methods Oper. Res.*, 46(3):287–307, 1997.
26. Jensen, H. A., Design and sensitivity analysis of dynamical systems subjected to stochastic loading, *Comput. Struct.*, 83(14):1062–1075, 2005.
27. Rubinstein, R. and Kroese, D., *The Cross-Entropy Method*, New York: Springer Science+Business Media, Inc., 2004.
28. de Boer, P.-T., Kroese, D., Mannor, S., and Rubinstein, R., A tutorial on cross-entropy method, *Ann. Oper. Res.*, 134:19–67, 2005.
29. Kurtz, N. and Song, J., Cross-entropy-based adaptive importance sampling using Gaussian mixture, *Struct. Saf.*, 42:35–44, 2013.
30. Li, J., Biondini, G., Kath, W. L., and Kogelnik, H., Anisotropic hinge model for polarization-mode dispersion in installed fibers, *Opt. Lett.*, 33(16):1924–1926, 2008.

31. Li, J., Biondini, G., Kath, W. L., and Kogelnik, H., Outage statistics in a waveplate hinge model of polarization-mode dispersion, *J. Lightwave Technol.*, 28(13):1958–1968, 2010.
32. Wang, H. and Zhou, X., A cross-entropy scheme for mixtures, *ACM Trans. Model. Comput. Simul. (TOMACS)*, 25(1):6, 2015.
33. Nocedal, J. and Wright, S., *Numerical Optimization*, New York: Springer Science & Business Media, 2006.

APPENDIX. THE SCORE FUNCTION METHOD

Here we provide a brief description of the score function method to compute the gradient of the reliability constraints. Under certain conditions [17, 20], the gradient of the failure probability with respect to the design parameters can be written as

$$\begin{aligned} \frac{\partial P}{\partial x_i} &= \frac{\partial}{\partial x_i} \int \mathbb{I}_{\{g>0\}}(\mathbf{z})q(\mathbf{z}; \mathbf{x})d\mathbf{z} = \int \mathbb{I}_{\{g>0\}}(\mathbf{z})\frac{\partial}{\partial x_i}q(\mathbf{z}; \mathbf{x})d\mathbf{z} \\ &= \int \mathbb{I}_{\{g>0\}}(\mathbf{z})\frac{\partial}{\partial x_i} \ln[q(\mathbf{z}; \mathbf{x})]q(\mathbf{z}; \mathbf{x})d\mathbf{z} = \mathbb{E}_q[\mathbb{I}_{\{g>0\}}(\mathbf{z})s(\mathbf{z}; \mathbf{x})], \end{aligned} \quad (\text{A.1})$$

where

$$s(\mathbf{z}; \mathbf{x}) = \frac{\partial}{\partial x_i} \ln(q(\mathbf{z}; \mathbf{x})),$$

is known as the score function. It should be clear that the partial derivatives Eq. (A.1) then can be estimated with a MC simulation or the CE method, just like the failure probability itself. More details of the score function method, including how to derive the score function $s(\mathbf{z}; \mathbf{x})$ for various distributions can be found in [17, 20] and the references therein, and shall not be repeated in this paper.