Journal of Computational Physics 344 (2017) 23–35

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A subset multicanonical Monte Carlo method for simulating rare failure events

Xinjuan Chen[a], Jinglai Li[b,*]

[a] Department of Mathematics, College of Science, Jimei University, Xiamen, Fujian, 361021, China
[b] School of Mathematical Sciences, Institute of Natural Sciences, and MOE Key Laboratory of Scientific and Engineering Computing, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 21 December 2016
Received in revised form 18 April 2017
Accepted 20 April 2017
Available online 25 April 2017

Keywords:
Failure probability estimation
Multicanonical Monte Carlo
Subset simulation
Uncertainty quantification

ABSTRACT

Estimating failure probabilities of engineering systems is an important problem in many engineering fields. In this work we consider such problems where the failure probability is extremely small (e.g. $\leq 10^{-10}$). In this case, standard Monte Carlo methods are not feasible due to the extraordinarily large number of samples required. To address these problems, we propose an algorithm that combines the main ideas of two very powerful failure probability estimation approaches: the subset simulation (SS) and the multicanonical Monte Carlo (MMC) methods. Unlike the standard MMC which samples in the entire domain of the input parameter in each iteration, the proposed subset MMC algorithm adaptively performs MMC simulations in a subset of the state space, which improves the sampling efficiency. With numerical examples we demonstrate that the proposed method is significantly more efficient than both of the SS and the MMC methods. Moreover, like the standard MMC, the proposed algorithm can reconstruct the complete distribution function of the parameter of interest and thus can provide more information than just the failure probabilities of the systems.

1. Introduction

Real-world engineering systems are unavoidably subject to various uncertainties such as material properties, geometric parameters, boundary conditions and applied loadings. These uncertainties may cause undesired events, in particular, system failures or malfunctions, to occur. Accurate evaluation of failure probability of a given system is essential in many engineering fields such as risk management [1], structural safety [2], reliability-based design and optimization [3], and thus is a central task of uncertainty quantification.

Conventionally, the failure probability is often computed by constructing linear or quadratic expansions of the system model around the so-called most probable point or β-point [4], which is known as the first/second order reliability method (FORM/SORM); see e.g., [5] and the references therein. It is well known that FORM/SORM may fail for systems with nonlinearity or multiple failure modes. The Monte Carlo (MC) simulation, which estimates the failure probability by repeatedly simulating the underlying system, provides an alternative to the FORM/SORM methods. The MC method does not make any reduction to the underlying models, and so it can be applied to any systems. On the other hand, it is well known that the MC method suffers from slow convergence, and can become prohibitively expensive when the system failures are rare

[*] Corresponding author.
E-mail addresses: chenxinjuan@jmu.edu.cn (X. Chen), jinglaili@sjtu.edu.cn (J. Li).

http://dx.doi.org/10.1016/j.jcp.2017.04.051
0021-9991/© 2017 Elsevier Inc. All rights reserved.

(for example, around 10^{-10}). To this end, many advanced sampling schemes have been developed to reduce the estimation variance and improve the computational efficiency. Among these schemes, the subset simulation (SS) method proposed by Au and Beck [6,7], is one of the most popular sampling strategies for estimating rare failure probabilities. Simply speaking, SS successively constructs a sequence of nested events with the very last one being the event of interest, and the probability of each event is estimated conditionally upon the previous one. Another attractive approach for estimating the failure probability is the multicanonical Monte Carlo (MMC) method [11,12], which was first developed to simulate rare events in physical systems. Later the method was used to estimate rare failure events in optical communication systems [13,14]. More recently, a surrogate accelerated MMC method was developed in [15] for uncertainty quantification applications. The main idea of the MMC method is to partition the state space of the parameter of interest (which is usually a scalar and will be referred to the performance parameter in what follows) into a set of small bins, and then iteratively construct a so-called flat-histogram distribution that can assign equal probabilities into each of the bins. Note that a major advantage of the MMC method is that it can reconstruct the entire distribution function of the parameter of interest, thus can provide with more information than just estimating the probability of a single event. Other small probability estimation methods include, the cross entropy method [8,9], and the population Monte Carlo [10], just to name a few.

In this work, we propose a new algorithm that combines the key ideas of the SS and the MMC methods. Specifically, the new algorithm also constructs a sequence of nested subdomains of the performance parameter, and then performs the MMC scheme in each subdomain. The algorithm preserves some key properties of the standard MMC algorithm, while using the subset idea to accelerate the computation. We thus refer to the proposed algorithm as the subset MMC (SMMC) method in the rest of the work. Like the MMC method, the proposed SMMC algorithm can also compute the entire distribution function of the parameter of interest. Using several examples, we compare the performance of the proposed SMMC algorithm with that of the SS and the MMC methods, and the numerical results show that the new algorithm can significantly outperform both of the original ones.

The rest of the work is organized as the following. In Section 2 we describe the mathematical formulation of the failure probability estimation problem. We then introduce the SS method in Section 3 and the MMC method in Section 4 respectively. The proposed SMMC algorithm is presented in Section 5 and several numerical examples are provided in Section 6. Finally some closing remarks will be given in Section 7.

2. Failure probability estimation

In this section, we shall describe the failure probability estimation problem in a general setting. Consider a probabilistic model where $\mathbf{x}$ is a d-dimensional random variable that represents the uncertainty in the model and the system failure is defined by a real-valued function

$$y = f(\mathbf{x}), \tag{2.1}$$

which is known as the *perform function*. For simplification, we shall assume that the state space of $\mathbf{x}$ is R^d. The event of system failure is defined as that y exceeds a certain threshold value y^*:

$$F = \{\mathbf{x} \in R^d \mid y = f(\mathbf{x}) > y^*\}, \tag{2.2}$$

and as a result the failure probability is

$$P_F = \mathbb{P}(F) = \int\limits_{\{\mathbf{x}\in R^d \mid f(\mathbf{x})>y^*\}} \pi(\mathbf{x})d\mathbf{x} = \int\limits_{\mathbf{x}\in R^d} I_F(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \tag{2.3}$$

where $I_A(\mathbf{x})$ is defined as an indicator function of a set A:

$$I_A(\mathbf{x}) = \begin{cases} 1 & \text{if}\, \mathbf{x} \in A, \\ 0 & \text{if}\, \mathbf{x} \notin A; \end{cases}$$

and $\pi(\mathbf{x})$ is the probability density function (PDF) of $\mathbf{x}$. In what follows we shall omit the integration domain when it is simply R^d. This is a general definition for failure probability, which is widely used in many disciplines involving with reliability analysis and risk management. Ideally, P_F can be computed by using the standard MC estimation:

$$P_F \approx \frac{1}{N}\sum_{j=1}^{N} I_F(\mathbf{x}_j), \tag{2.4}$$

where samples $\mathbf{x}_1, ..., \mathbf{x}_N$ are drawn from the distribution $\pi(\mathbf{x})$. However, as has been discussed in Section 1, most engineering systems require high reliability, namely the failure probability $P_F \ll 1$. In this case, MC requires a large number of samples to produce a reliable estimate of P_F. On the other hand, in almost all practical cases, the performance function $f(\mathbf{x})$ does not admit analytical expression and has to be evaluated through expensive computer simulations, which makes the MC estimation of the failure probability prohibitive. Many advanced sampling schemes have been developed to compute the failure probability P_F, and in what follows, we shall introduce two popular choices of them: the SS and the MMC methods.

3. The subset simulation method

A brief introduction of the SS method, largely following [6], will be provided in this section. Note that we shall only outline the basic idea of the SS algorithm, and readers who are interested in the implementing details are referred to [6,16,17] and the references therein.

The idea of the SS method is to decompose the rare event F into a sequence of "less-rare" nested events,

$$F = F_K \subset F_{K-1} \subset \cdots \subset F_1 \subset F_0,$$

where F_k is a more frequent event than F_{k+1} for $k = 1, \cdots, K-1$ and $F_0 = R^d$. Hence, the failure probability P_F of the event F can be computed by

$$\begin{aligned} P_F = \mathbb{P}(F) = \mathbb{P}(F_K) &= \mathbb{P}(F_1)\frac{\mathbb{P}(F_2)}{\mathbb{P}(F_1)}\frac{\mathbb{P}(F_3)}{\mathbb{P}(F_2)}\cdots\frac{\mathbb{P}(F_K)}{\mathbb{P}(F_{K-1})} \\ &= \mathbb{P}(F_1|F_0)\mathbb{P}(F_2|F_1)\cdots\mathbb{P}(F_K|F_{K-1}), \end{aligned} \tag{3.1}$$

where $\mathbb{P}(F_k|F_{k-1})$ is the conditional probability of event F_k given the occurrence of event F_{k-1}. Note that $\mathbb{P}(F_1|F_0) = \mathbb{P}(F_1)$.

Before looking deeper into the algorithm, we will set up some new notations first. Given an intermediate threshold value y_k, define $F_k = \{\mathbf{x} \in R^d \mid f(\mathbf{x}) > y_k\}$ as a corresponding intermediate event. In addition, we choose $y_0 = -\infty$ so that $F_0 = R^d$. The failure probability P_F is now evaluated in a sequential manner. In short words, starting from stage $k = 0$, the algorithm generates a number of samples $\mathbf{x}_1, \cdots, \mathbf{x}_N$ from the distribution with PDF

$$\pi_k(\mathbf{x}) = \pi(\mathbf{x}|F_k) \propto \pi(\mathbf{x}) I_{F_k}(\mathbf{x}), \tag{3.2}$$

where it should be noted that $\pi_0(\mathbf{x}) = \pi(\mathbf{x})$. It is worth noticing that drawing samples from $\pi_k(\cdot)$ is done with the Markov Chain Monte Carlo (MCMC) methods, which do not require the knowledge of the unavailable normalization constant in Eq. (3.2). Afterward, one chooses an intermediate threshold value y_{k+1} and compute the conditional probability $\mathbb{P}(F_{k+1}|F_k)$ with standard MC, getting

$$\mathbb{P}(F_{k+1}|F_k) \approx \frac{1}{N}\sum_{j=1}^{N} I_{F_{k+1}}(\mathbf{x}_j). \tag{3.3}$$

The crucial point here is to pick y_{k+1} so that the resulting conditional probability $\mathbb{P}(F_{k+1}|F_k)$ is not too small. A commonly used approach is to let y_{k+1} be the $(1-\gamma)$-th percentile of samples $\{y_1 = f(\mathbf{x}_1), \cdots, y_N = f(\mathbf{x}_N)\}$ for some not too small positive number γ (e.g., $= 0.1$). The algorithm proceeds until y_{k+1} reaches y^*. Therefore, one obtains the estimates of all the conditional probabilities $\mathbb{P}(F_1|F_0), \cdots, \mathbb{P}(F_K|F_{K-1})$ (assuming the algorithm reaches y^* at the $(K-1)$-th iteration), and substituting the results into Eq. (3.1) yields an estimate of the sought failure probability P_F. We reinstate that the complete description of SS method is well documented in several works [6,16–18].

4. The multicanonical Monte Carlo method

We will now succinctly present the scheme of the MMC method, which is another effective algorithm used to estimate small failure probabilities. Unlike the SS method, MMC solves the problem by constructing the distribution of the output parameter y. Namely, let $\pi_y(\cdot)$ be the PDF of y, and then the failure probability can be obtained by

$$P_F = \int_{y^*}^{b} \pi_y(y)dy, \tag{4.1}$$

where b is in principle the maximum value of y. In practice, however, it is often not necessary to let b be the maximum value of y, especially when y is not bounded from above. It is easy to see that, for our purposes, it is sufficient to choose b such that $\mathbb{P}(y > b) \ll \mathbb{P}(y > y^*)$. Hence, in order to find the failure probability of the system, one only needs the PDF of y. To be more precise, we only need the PDF of y in the interval $[y^*, b]$. This is not a simple task, however, because the failure region is typically located in the tail of y.

A popular strategy applied to estimate the PDF of a continuous random variable y with simulation is to approximate the PDF with histograms. Suppose we are interested in the PDF of y in the interval $\mathcal{B} = [a, b]$, and we first equally decompose $\mathcal{B}$ into m bins of width Δ, whose centers are the discrete values $\{b_1, ..., b_m\}$. We define the i-th bin as the interval $B_i = [b_i - \Delta/2, b_i + \Delta//2]$ and the probability for y to be in B_i is $P_i = \mathbb{P}\{y \in B_i\}$. Note that the width of each bin needs not to be identical in principle, and here we use identical bin width just for the simplicity of notations. In fact the choice of bins is critical to the performance of the MMC method, and in this respect, one possible guideline is that the bins should be chosen such that adjacent bins have probabilities within one order of magnitude of one other [19]. The PDF of y at point b_i then can be approximated by

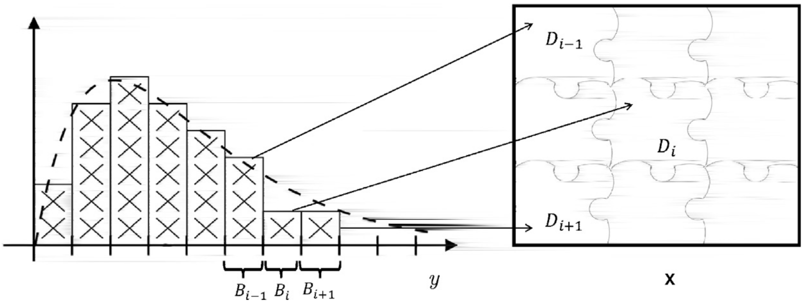

Fig. 1. Schematic illustration of the connection between B_i and D_i.

$$p(b_i) \approx P_i/\Delta,$$

if Δ is sufficiently small. This binning implicitly defines a partition of the input space X into m domains $\{D_i\}_{i=1}^m$, where

$$D_i = \{\mathbf{x} \in R^d : f(\mathbf{x}) \in B_i\}$$

is the domain in X that is mapped into the i-th bin B_i by $f(\cdot)$ (see Fig. 1). Note that, while B_i are simple intervals, the domains D_i are multidimensional regions with possibly tortuous topologies. As a result, the probability P_i can be re-written as an integral in the input space:

$$P_i = \int_{D_i} \pi(\mathbf{x})dx = \int I_{D_i}(\mathbf{x})\pi(\mathbf{x})dx = \mathbb{E}[I_{D_i}(\mathbf{x})]. \tag{4.2}$$

Now suppose that N samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are drawn from the distribution $\pi(\mathbf{x})$, possibly with MCMC, P_i can be estimated with the MC estimator:

$$\hat{P}_i^{MC} = \frac{1}{N}\sum_{j=1}^{N} I_{D_i}(\mathbf{x}_j) = \frac{N_i}{N}, \tag{4.3}$$

where N_i is the number of samples that fall in the domain D_i.

As is mentioned earlier, standard MC simulations have difficulty in reliably estimating the probabilities in the tail bins. The technique of importance sampling (IS) can be effectively used to address the issue. The principle idea of IS is to choose a biasing distribution $q(\mathbf{x})$ and rewrite Eq. (4.2) as

$$P_i = \int I_{D_i}(\mathbf{x})\left[\frac{\pi(\mathbf{x})}{q(\mathbf{x})}\right]q(\mathbf{x})d\mathbf{x} = \mathbb{E}^*[I_{D_i}(X)w(X)] \tag{4.4}$$

where $w(\mathbf{x}) = \pi(\mathbf{x})/q(\mathbf{x})$ is called the IS weight, and $\mathbb{E}^*$ indicates the expectation with respect to the biasing distribution $q(\mathbf{x})$. It follows that the IS estimator of P_i becomes

$$\hat{P}_i^{IS} = \left(\frac{N_i^*}{N}\right)\left[\frac{1}{N_i^*}\sum_{j=1}^{N} I_{D_i}(\mathbf{x}_j)w(\mathbf{x}_j)\right], \tag{4.5}$$

where the samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are now generated from the biasing distribution $q(\mathbf{x})$, and N_i^* is the number of samples falling in the region D_i.

One can easily see that the key of IS is to choose an appropriate biasing distribution $q(\mathbf{x})$ that can help to achieve the objective of the simulation. While regular IS usually aims to estimate the probability in a given region, the goal of our simulation is to have a good estimate of P_i for all $i = 1 \ldots m$, and in this respect, it is reasonable to seek a biasing distribution that assigns equal probability to each bin and zero probability for any region outside $\mathcal{D} = \cup_{i=1}^m D_i$, which implies that

$$P_1^* = P_2^* = \ldots P_m^* = 1/m, \tag{4.6a}$$

where

$$P_i^* = \int_{R^d} I_{D_i}(\mathbf{x})q(\mathbf{x})d\mathbf{x} = E^*[I_{D_i}(X)], \text{for } i = 1, \cdots, m. \tag{4.6b}$$

We refer to the biasing distribution which satisfies Eqs. (4.6) to be flat-histogram (FH). One should be noted that the FH distributions are not unique, and among them there is one which assigns a constant weight to all $\mathbf{x} \in D_i$, i.e. $w(\mathbf{x}) = w_i$ for $\mathbf{x} \in D_i$ where $w_i = P_i/P_i^*$. In this case, the biasing distribution $q(\mathbf{x})$ is called to be uniform-weight (UW).

In particular, we assume that the biasing distribution $q(\mathbf{x})$ is given in the form of

$$q(\mathbf{x}) = \begin{cases} \dfrac{\pi(\mathbf{x})}{c_\Theta \Theta(\mathbf{x})} & \mathbf{x} \in \mathcal{D}; \\ 0 & \mathbf{x} \notin \mathcal{D}, \end{cases} \tag{4.7}$$

where $\Theta(\mathbf{x}) = \Theta_i > 0$ for all $x \in D_i, i = 1, ..., m$, satisfying

$$\sum_{i=1}^{m} \Theta_i = 1, \tag{4.8}$$

and c_Θ being a normalized constant.

It is easy to show that $q(\mathbf{x})$ given in Eq. (4.7) is UW with $w_i = c_\Theta \Theta_i$ for $i = 1, \ldots, m$. Next we shall impose the constraint so that $q(\mathbf{x})$ given in Eq. (4.7) is FH. Since

$$P_i^* = \int_{D_i} q(\mathbf{x}) d\mathbf{x} = \frac{\int_{D_i} \pi(\mathbf{x}) d\mathbf{x}}{c_\Theta \Theta_i} = \frac{P_i}{c_\Theta \Theta_i}, \tag{4.9}$$

and by setting the left hand side of Eq. (4.9) to be equal to $1/m$, we obtain

$$\Theta_i = \frac{m}{c_\Theta} P_i. \tag{4.10}$$

Substituting Eq. (4.10) into Eq. (4.8) results in $c_\Theta = m\rho$, where $\rho = \sum_{i=1}^{m} P_i$, and it follows immediately that $\Theta_i = P_i/\rho$. Note that in general the probability $\rho = \mathbb{P}[y \in \mathcal{B}] \leq 1$ and is unknown in advance. A conventional solution is to take a sufficiently large interval $\mathcal{B}$ so that $\rho \approx 1$, and we adopt this choice in this work.

However, for the reason that Θ_i, $i = 1, \cdots, m$, depend on the sought after unknown P_i, the actual UW–FH distribution just derived above can not be utilized directly to achieve the goal of sampling equally in each bin.

The MMC method uses an adaptive scheme to address this issue. Briefly speaking, MMC adaptively constructs a sequence of distributions

$$q_k(\mathbf{x}) = \begin{cases} \dfrac{\pi(\mathbf{x})}{c_k \Theta_k(\mathbf{x})}, & \mathbf{x} \in \mathcal{D}; \\ 0, & \mathbf{x} \notin \mathcal{D}, \end{cases} \tag{4.11}$$

where $\Theta_k(\mathbf{x}) = \Theta_{k,i}$ for $\mathbf{x} \in D_i$, converging to the actual UW–FH distribution. Before proceeding to the MMC algorithm, we derive an alternative representation of Θ_i from Eq. (4.10):

$$\Theta_i = P_i^* w_i/\rho, \quad \text{for} \quad i = 1, \cdots, m. \tag{4.12}$$

Typically, the MMC method starts from the original PDF $q_0(\mathbf{x}) = \pi(\mathbf{x})$, where the associated parameter values are $c_0 = 1$ and $\Theta_{0,i} = 1$ for all $i = 1, \ldots, m$. In the k-th iteration, one first draws N samples $\{\mathbf{x}_j\}_{j=1}^{N}$ from the current distribution $q_k(\mathbf{x})$, and then updates $\{\Theta_{k+1,i}\}_{i=1}^{m}$ using the following formulas derived from Eq. (4.12):

$$\hat{H}_{k,i} = \frac{N_{k,i}^*}{N}, \tag{4.13a}$$

$$w_{k,i} = c_k \Theta_{k,i}, \tag{4.13b}$$

$$\Theta_{k+1,i} = \hat{H}_{k,i} w_{k,i}/\rho, \tag{4.13c}$$

where $N_{k,i}^*$ is the number of samples falling into region D_i in the k-th iteration. It should be noted that, MMC usually employs MCMC to draw samples from $q_k(\mathbf{x})$, thanks to which we do not need to estimate c_k during the iterations (i.e., just to take $c_k = 1$ in each iteration). However, the constant is needed when one wants to compute P_i for $i = 1, \cdots, m$, using the IS estimator (4.5), in the final stage. To circumvent the obstacle, we estimate P_i by

$$P_i \approx \frac{\Theta_{K,i}}{\sum_{i=1}^{m} \Theta_{K,i}} \rho, \quad \text{for} \quad i = 1, \cdots, m,$$

where K is the index of the final iteration. Formal convergence analysis, as well as possible improvements of the MMC method are not discussed in this work, and readers who are interested may consult, e.g., [20–23], and the references therein.

5. The subset MMC method

As it's been described in the previous section, the conventional MMC method uses a sufficiently large interval $\mathcal{B}$ such that $\rho = 1$, which, unfortunately, is not an efficient approach for our purposes because the failure region that we are interested in, $[y^*, b]$, is typically a small subinterval of $\mathcal{B}$. As a result, only a very small portion of the samples will be used to estimate the density in the region of interest. To address the issue, we propose a subset MMC algorithm, which combines the key ideas of the SS and the MMC methods.

Consider the case where ρ is unknown. Similar to the SS method, we now construct a sequence of nested intervals $\mathcal{B}^0 \supset \ldots \supset \mathcal{B}^J$, where $\mathcal{B}^0 = \mathcal{B}$ and $\mathcal{B}^J = [y^*, b]$ is the interval of interest. It should be easy to see that the corresponding domains in the input space are also nested. Let $\rho_j = \mathbb{P}(y \in \mathcal{B}^j)$ for $j = 1, \cdots, J$, and as it's explained before, the failure probability $P_F \approx \rho_J$.

Let the bins $B_1, \ldots, B_m$ be predetermined as the previous section and will not be changed as the algorithm proceeds. Moreover, for the sake of simplicity, we assume that the threshold value y^* coincides with the left boundary of one of the bins, i.e. $b_{m^*} - \Delta/2 = y^*$ for some integer $1 \le m^* \le m$. In this case, it will be natural to construct each interval $\mathcal{B}^j$ as a union of bins: $\mathcal{B}^j = \cup_{i=m_j}^{m} B_i$ for some integer $1 \le m_j \le m^*$. It can be easily seen that $m_1 \le m_2 \le \cdots$. Starting from $\mathcal{B}^0$ (with $\rho_0 = 1$ and $m_0 = 1$), we now perform a standard MMC within the interval $\mathcal{B}^j$ and compute the probabilities of bins from m_j to m: $P_{m_j}, \ldots, P_m$. An m_{j+1} is chosen such that $m_j \le m_{j+1} \le m$, which indicates that the choice of m_{j+1} determines the next interval $\mathcal{B}^{j+1}$ (the criterion that we use to determine m_j will be provided later). The basic thought is that we can gradually concentrate the samples toward the region of interest. It follows immediately that the associated probability ρ_{j+1} can be estimated by

$$\rho_{j+1} = \sum_{i=m_{j+1}}^{m} P_i.$$

The algorithm proceeds until $m_{j+1} = m^*$. The complete scheme is described in Algorithm 1. Some remarks regarding the implementation of the algorithm of SMMC are listed in order.

- In each iteration, m_{j+1} is determined according to the following. First, a not-too-small positive number $\alpha < 1$ (e.g. $= 0.2$) is chosen. Then we select an m_{j+1} such that (approximately) $100\alpha\%$ of the samples fall in the interval $\mathcal{B}^j = \cup_{i=m_{j+1}}^{m} B_i$.
- In line 18, the samples are drawn from $q_k(\cdot)$ using the MCMC methods. In particular, we implement a multiple chain MCMC algorithm specifically tailored for this problem. The details of the algorithm are given in Appendix A.
- The terminating condition used here is $m_j = m^*$, i.e., when the interval in which we perform MMC reaches the area of interest.

So far, although we have presented the SMMC algorithm as a variant of the standard MMC method, the key idea is also inspired by the SS method. Namely, in the SMMC algorithm, a sequence of subsets are first constructed and in each subset a MMC iteration rather than plain MC is performed in order to drive samples towards the failure region. It is also interesting to notice that, just like the standard MMC method, one of the significant advantages of the SMMC method is that, if desired, it can construct the entire PDF of y without any additional cost, as we obtain the estimates of P_i for $i = 1, \cdots, m$ during the iteration. We shall illustrate this advantage with numerical examples in Section 6.

Finally we shall provide a simple analysis of the estimator error of the SMMC algorithm. Two simplifications are made for the sake of convenience. One is that the samples drawn are independent while noting that this is certainly not the case when the samples are drawn with the MCMC method. The other assumption is that the biasing distribution is "perfectly flat" in the last iteration; namely, the biasing distribution is given by Eq. (4.7), where $\{\Theta_i\}_{i=m^*}^{m}$ are defined in Eq. (4.10) and

$$c_\Theta = (m - m^*)\rho,$$

where $(m - m^*)$ is the number of bins in the last iteration. Now for any $m^* \le i \le m$, the estimator of P_i is

$$\hat{P}_i = \sum_{j=1}^{N} I_{D_i}(\mathbf{x}_j) w(\mathbf{x}_j), \tag{5.1}$$

where the samples are drawn from distribution Eq. (4.7) with

$$w(\mathbf{x}) = (m - m^*)\rho' \Theta(\mathbf{x}). \tag{5.2}$$

Note that ρ' in Eq. (5.2) is an estimate of ρ as the actual value of ρ is unknown in our problem.

Algorithm 1 The subset MMC algorithm.

```
Require: $\pi(\mathbf{x})$, $\{B_i\}_{i=1}^m$, $m^*$, $N$, $K$, $\alpha$.
Ensure: $P_F$.
procedure $P_F = \mathrm{SMMC}(\pi(\mathbf{x}), \{B_i\}_{i=1}^m, m^*, N, \alpha)$
    Initialization: $j = 0$, $\boldsymbol{\Theta}_0 = (1, \cdots, 1)$, $m_0 = 1$, $\rho_0 = 1$;
    while $m_j < m^*$ do
        $[\boldsymbol{\Theta}_{j+1}, m_{j+1}] = \mathrm{MMC}(\boldsymbol{\Theta}_j, m_j, \pi(\mathbf{x}), \{B_i\}_{i=m_j}^m, m^*, N, K, \alpha)$;
        for $i = m_j ... m$ do
            $P_i = \frac{\Theta j+1,i}{\sum_{i'=m_j}^m \Theta_{j+1,i'}} \rho_j$;
        end for
        $\rho_{j+1} = \sum_{i=m_{j+1}}^m P_i$;
        $j = j + 1$;
    end while
    $P_F = \rho_j$;
end procedure
procedure $[\boldsymbol{\Theta}^+, m^+] = \mathrm{MMC}(\boldsymbol{\Theta}^-, m^-, \pi(\mathbf{x}), \{B_i\}_{i=1}^m, m^*, N, K, \alpha)$
    $\boldsymbol{\Theta}_0 = \boldsymbol{\Theta}^-$;
    $D = \{\mathbf{x} \in R^d \mid g(\mathbf{x}) \in \cup_{i=m^-}^m B_i\}$;
    for $k = 0 ... K$ do
        Let $q_k$ be given by Eq. (4.11) with $\boldsymbol{\Theta}_k$;
        Draw $N$ samples $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ from $q_k$;
        Evaluate $S_k = \{f(\mathbf{x}_1), ..., f(\mathbf{x}_N)\}$;
        Compute $\{\Theta_{k+1,m^-}, ..., \Theta_{k+1,m}\}$ using Eqs. (4.13);
    end for
    Let $y_\alpha$ be the $(1-\alpha)$-th quantile of set $S_K$;
    Let $m^+$ be the index of the bin such that $y_\alpha \in B_{m^+}$;
    $m^+ = \min\{m^+, m^*\}$;
    $\Theta^+ = \Theta_{K+1}$;
end procedure
```

The mean square error (MSE) of Eq. (5.1) is computed as

$$\begin{aligned}\mathrm{MSE}[\hat{P}_j] &= \mathrm{VAR}[\hat{P}_j] + (\mathbb{E}[\hat{P}_j] - P_j)^2, \\ &= \frac{1}{n}\left(\left(\frac{\rho'}{\rho}\right)^2 (m-m^*)P_j^2 - \left(\frac{\rho'}{\rho}\right)^2 P_j^2\right) + \left(\left(\frac{\rho'}{\rho}\right) - 1\right)^2 P_j^2, \\ &= \frac{(m-m^*-1)}{N}\phi^2 P_j^2 + (\phi - 1)^2 P_j^2,\end{aligned}$$

where $\phi = \rho'/\rho$. It is not difficult to see that the optimal value of ϕ that minimizes the MSE is $\phi = N/(N + m - m^* - 1)$, and the resulting minimal MSE is

$$\mathrm{MSE}_{\min} = \frac{m - m^* - 1}{m - m^* - 1 + N} P_j^2.$$

It is interesting to see from the results that for the MSE to be minimal, one should choose

$$\rho' = \frac{N}{(N + m - m^* - 1)}\rho,$$

rather than $\rho' = \rho$. However, when $m - m^* \ll N$ which is the usual case, $\frac{N}{(N+m-m^*-1)} \approx 1$, and thus we choose not to include the factor $\frac{N}{(N+m-m^*-1)}$ in the estimate of ρ in Algorithm 1.

6. Numerical examples

6.1. A two-dimensional mathematical example

The first example is a two-dimensional mathematical problem. Suppose that $\mathbf{x} = (x_1, x_2)$ is a two-dimensional random variable where x_1 and x_2 both follow standard normal distribution and are independent to each other. The event of failure is defined as

$$\min\{\|\mathbf{x} - \mathbf{x}_r\|_2, \|\mathbf{x} - \mathbf{x}_l\|_2\} < 1,$$

where $\mathbf{x}_r = (8, 2)$ and $\mathbf{x}_l = (-8, 2)$. On the one hand, it is clear that the problem has two disjoint failure domains: $\{x \in R^2 | \|\mathbf{x} - \mathbf{x}_r\|_2 < 1\}$ and $\{\mathbf{x} \in R^2 | \|\mathbf{x} - \mathbf{x}_l\|_2 < 1\}$, which poses challenges for many standard IS methods. On the other hand, for this two-dimensional example, the failure probability can be accurately estimated by performing a numerical integration, yielding $P_F = 1.41 \times 10^{-13}$.

Table 1
Example 1: performance comparison of the three methods with different sample sizes.

	SS	MMC	SMMC
n	2.54×10^4	2.0×10^4	1.70×10^4
RMSE	49	25.9	0.49
n	1.14×10^5	1.0×10^5	0.98×10^5
RMSE	2.69	2.56	0.078
n	3.34×10^5	3.0×10^5	2.99×10^5
RMSE	0.16	2.13	0.017

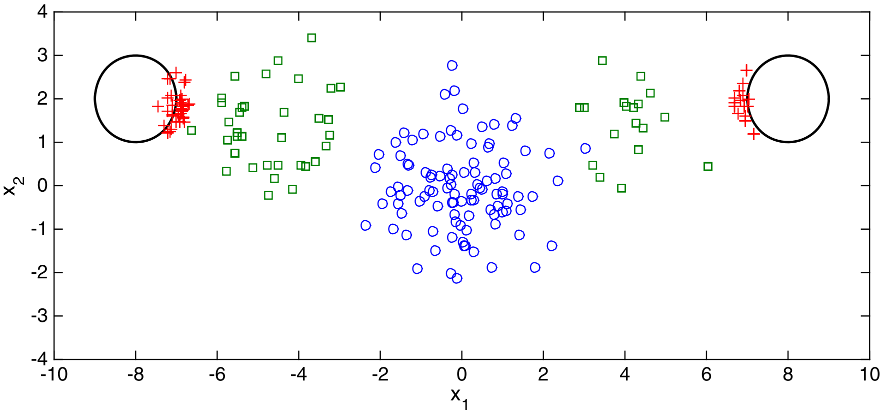

Fig. 2. Example 1: the sample distributions in one test trial for SMMC.

We estimate the probability with three methods: SS, standard MMC and SMMC. As for the SS method, we largely follow the implementation described in [18] and set $\gamma = 0.1$ as it's suggested in [18]. For both of the standard MMC and proposed SMMC methods, the entire region of interest of the output is taken to be $[0, 100]$ and is equally divided into 100 bins. For the SMMC method, we take $\alpha = 0.2$ to generate the nested intervals.

We compare the three methods using three different numbers of samples: 2×10^4, 1×10^5 and 3×10^5. It should be noted that, while for the MMC algorithm we can choose the exact sample size by fixing the total number of iterations and the number of samples used in each iteration, we can not precisely control it in the SS and the SMMC methods, and so we can only adjust the algorithms so that the total amounts of samples are close to the aforementioned numbers. Also, during all the computations, we manage to ensure that the sample size of the SMMC is smaller than those in the other two algorithms. For each of the three methods, we repeatedly perform the simulations for 100 times and computed the average number of samples as well as the relative mean square errors (RMSE):

$$\mathrm{RMSE} = \frac{\frac{1}{L}\sum_{l=1}^{L} |\hat{P}_l - P_F|^2}{P_F^2},$$

where $L = 100$ is the total number of experiments and $\hat{P}_l$ is the estimated probability at the l-th test. The test results are presented in Table 1. We have found that, in the SMMC method, most tests terminate within three iterations. We also show the sample distributions in one test trial in Fig. 2. One can see from the figure that, the SMMC method is capable of directing samples toward the failure region in a rather efficient manner. More information can be learned from the results in Table 1. In particular, we can see that in all the cases, the SMMC performs substantially better than the other two methods, even with less samples.

6.2. A high dimensional mathematical example

This one is also a mathematical problem, but of a higher dimensionality than the previous one. Specifically, we let $\mathbf{x}$ be a d-dimensional random variable following standard Gaussian distribution: $\mathbf{x} \sim N(0, I)$ where I is the $d \times d$ identity matrix. The failure event is defined as $f(\mathbf{x}) > y^*$ with

$$f(\mathbf{x}) = \|\mathbf{x}\|_2^2. \tag{6.1}$$

In our numerical tests, we choose $d = 10$ and $y^* = 75$. In this setting, the failure probability can be computed analytically as $P_F = 4.76 \times 10^{-12}$.

Table 2
Example 2: performance comparison of the three methods with different sample sizes.

	SS	MMC	SMMC
n	2.5×10^4	2.0×10^4	1.7×10^4
RMSE	6.3	7.1	0.5
n	1.09×10^5	1.0×10^5	1.07×10^5
RMSE	2.6	2.7	0.2
n	3.27×10^5	3.0×10^5	2.90×10^5
RMSE	0.15	2.4	0.02

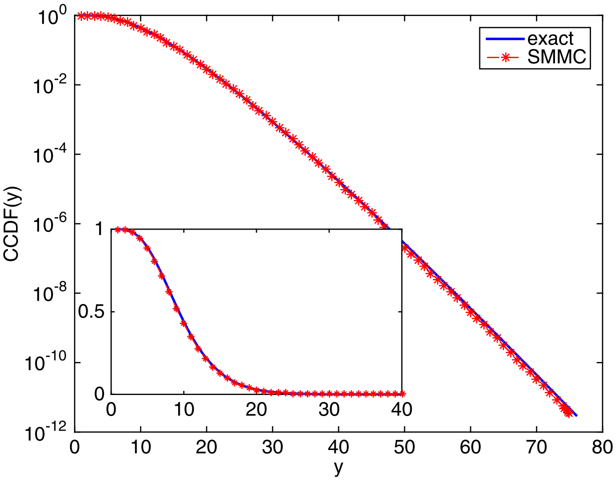

Fig. 3. Example 2: the CCDF computed by the SMMC method compared to the exact results, both are plotted on a logarithmic scale. Inset: the same plots but on a linear scale.

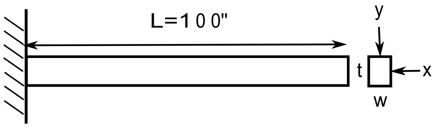

Fig. 4. Example 3: a cantilever beam subject to horizontal and vertical loads.

We test all the three methods on this example as well. The specifications of the implementations of all the three methods are kept the same as those in the first example. Like the first example, we test each method with three different sample sizes: 2×10^4, 1×10^5 and 3×10^5, and repeatedly perform the simulations 100 times for each sample size. The RMSE results are available in Table 2. The results indicate that, in this example, the SMMC method also substantially outperformed the other two methods. Besides, as it is mentioned earlier, another advantage of the SMMC method over SS method is that it can also be used to construct the complete distribution of the output y. To show this, we plot in Fig. 3 the complement cumulative distribution function (CCDF) of y obtained by the SMMC method, which is defined as

$$\mathrm{CCDF}(y) = 1 - \mathrm{CDF}(y),$$

where $\mathrm{CDF}(y)$ is the cumulative distribution function (CDF) of y. As a comparison, we also show the exact CCDF function of y, and one can see that the result of SMMC agrees very well with the exact one.

6.3. Cantilever beam

We now consider a real-world example: a cantilever beam problem studied in [24,25]. The beam is illustrated in Fig. 4, with width w, height t, length L, and subject to transverse load Y and horizontal load X. This is a popular benchmark problem in the reliability analysis literature. The quantity of interest in this problem is the maximum deflection of the beam, which can be computed as,

$$y = \frac{4L^3}{Ewt}\sqrt{\left(\frac{Y}{t^2}\right)^2 + \left(\frac{X}{w^2}\right)^2}.$$

Table 3
Example 3: the mean and variance of the random parameters.

Parameter	w	t	X	Y	E
Mean	4	4	500	1000	2.9×10^6
Variance	0.001	0.0001	100	100	1.45×10^6

Table 4
Example 3: performance comparison of the three methods with different sample sizes.

	SS	MMC	SMMC
n	2×10^4	2.0×10^4	2.05×10^4
RMSE	1.5	–	1.6
n	1.09×10^5	1.0×10^5	1.07×10^5
RMSE	0.13	0.1	0.08
n	3.0×10^5	3.0×10^5	2.88×10^5
RMSE	0.05	0.05	0.04

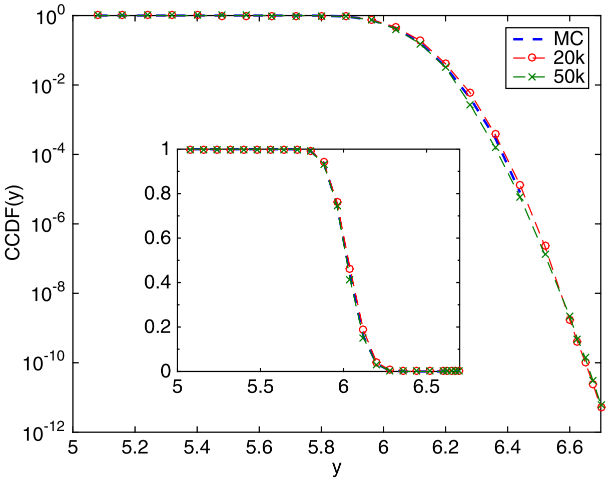

Fig. 5. Example 3: the CCDF computed by the SMMC method with two different sample sizes: 10^4 (circles) and 5×10^4 (crosses). As a comparison, we also plot in the figure the result of standard MC with 10^6 samples (dashed line). All are plotted on a logarithmic scale. Inset: the same plots but on a linear scale.

In particular, we assume that the beam length is fixed $L = 100$, and the beam width w, the height t, the applied loads X and Y, as well as the elastic modulus E of the material, are random parameters. We further assume that these random parameters are all independently distributed, with each following a normal distribution. The means and the variances of these parameters are provided in Table 3. We define the event of failure as that the deflection y exceeds a threshold value $y^* = 6.7$.

We first evaluate the resulting failure probability with the SS method of 10^6 samples, yielding an estimate 9.7×10^{-12}, and we regard this estimate as the true failure probability. We then estimate the failure probability using the SS, the MMC, and the SMMC methods with three different sample sizes: 2×10^4, 1×10^5 and 3×10^5; we repeatedly perform the simulations 100 times for each sample size. The RMSE results are available in Table 4. We note that, the MMC method can not yield a stable a estimate of the failure probability with 2×10^4 samples, and thus its RMSE result is omitted in Table 4. We can see from the results that the SS and the SMMC methods have quite similar performance for this example. Nevertheless, as is discussed earlier, the SMMC method can compute the complete distribution of y, which is illustrated by the CCDF of y computed by the SMMC method in Fig. 5.

6.4. Quarter car model

The last example is the quarter car model for vehicle suspension systems [26]. The schematic illustration of the model is shown in Fig. 6, where the sprung mass m_s and the unsprung mass m_u are connected by a nonlinear spring and a linear damper. The stiffness of the nonlinear spring is k_s and the damping coefficient of the linear damper is c. The displacement of the wheel $z(t)$ represents the interaction of the quarter car system with the terrain. Mathematically, the model is described by a two-degree-of-freedom ODE system [26]:

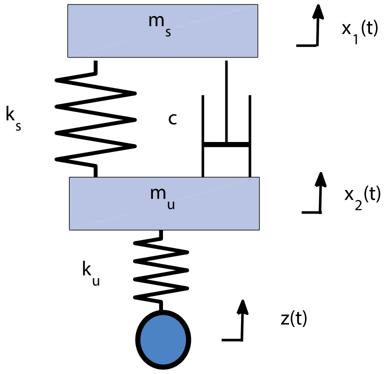

Fig. 6. Example 4: the quarter car model.

Table 5
Example 4: the parameter values of the quarter car model.

m_s	m_u	k_s	k_u	c
20	40	400	2000	600

$$m_s \frac{d^2x_1}{dt^2} = -k_s(x_1-x_2)^3 - c\left(\frac{dx_1}{dt} - \frac{dx_2}{dt}\right), \tag{6.2a}$$

$$m_u \frac{d^2x_2}{dt^2} = k_s(x_1-x_2)^3 + c\left(\frac{dx_1}{dt} - \frac{dx_2}{dt}\right) + k_u(z(t)-x_2), \tag{6.2b}$$

where x_1 and x_2 are the displacements of the sprung and the unsprung masses respectively. In our example, we assume that the uncertainty in the system arises from the random road profile, and as a result the wheel displacement $z(t)$ is modeled as a zero-mean white Gaussian random force with standard deviation $\sigma = 0.05$. The other model parameters are all taken to be fixed and the values of them are shown in Table 5. The quantity of interest is the maximum difference between displacements of the sprung and the unsprung springs in a given interval $[0, T]$,

$$y = \max_{0\le t\le T} \{|x_1(t) - x_2(t)|\},$$

and we want to reconstruct the CCDF of y. With the CCDF, we can estimate directly the probability $\mathbb{P}(y > y^*)$ for any y^* in the range of interest.

In the numerical simulations, we take $T = 1$, and the initial conditions of Eqs. (6.2) to be

$$x_1(0) = \frac{dx_1}{dt}(0) = 0, \quad x_2(0) = \frac{dx_2}{dt}(0) = 0.$$

Eqs. (6.2) are numerically solved with the classical Runge–Kutta method where the step size is taken to be $\Delta t = T/100$, which means that the random variable in this problem is effectively of 100 dimensions.

Also, the CCDF of y using a standard MC method with 10^6 samples is constructed. We perform the SMMC method with three sample sizes 10^4, 5×10^4 and 10^5 respectively, and present all the results in Fig. 7. One can see from the figure that, the results of the SMMC agree largely with those of the standard MC. Without surprise, the MC method can only obtain the CCDF at the order of 10^{-6}, while the SMMC method can compute the CCDF down to 10^{-12} and smaller with much less samples than the MC method. One can also see that the result of the SMMC of 10^4 samples departs evidently from those of 5×10^4 and 10^5, indicating that the sample size of 10^4 may not be sufficient for this problem. With around 10^5 samples, we can compute the probability as small as 10^{-12} using the SMMC method. Note that the CCDF computed with the SMMC method can also provide us with other important information such as the extreme quantiles. For instance, we can see directly from the CCDF that the $(1\text{–}10^{-8})$-th quantile is 0.0198 and the $(1\text{–}10^{-10})$-th is 0.0224. Such information can not be easily obtained with the SS method.

7. Conclusions

In summary, we propose an efficient algorithm for estimating failure probabilities of complex engineering systems, which combines the central ideas of the SS and the MMC methods. The new algorithm constructs a sequence of subdomains of the performance parameter y and performs regular MMC iterations within each subdomain only. We demonstrate that the proposed SMMC method can significantly outperform the two original methods, and moreover, like the MMC method, it can be used to reconstruct the entire distribution function of the performance parameter. We believe that the SMMC method can be a useful tool for many practical engineering problems that involve small failure probability estimations.

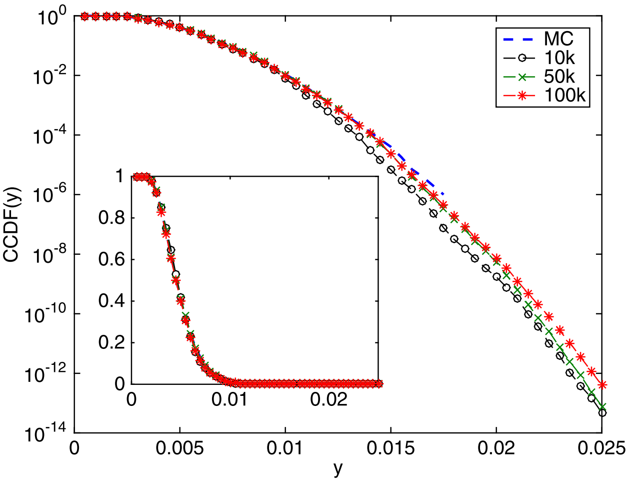

Fig. 7. Example 4: the CCDF computed by the SMMC method with three different sample sizes: 10^4 (circles), 5×10^4 (crosses) and 10^5 (asterisks). As a comparison, we also plot in the figure the result of standard MC with 10^6 samples (dashed line). All are plotted on a logarithmic scale. Inset: the same plots but on a linear scale.

Several improvements and extensions of the proposed algorithm are possible. First, for systems with highly intensive computer models, even with the SMMC method, the total computational cost is still unaffordable. In such problems, a possible solution is to construct computationally inexpensive surrogate models and use them in the simulations (see, e.g. [24,27,28]). To this end, surrogates have been used to accelerate the simulations in both the SS [29] and the MMC [15] methods. Thus we hope to develop surrogate based methods to reduce the computational cost of the SMMC algorithm. Secondly, in many practical problems, one often has computer models with different fidelities for the system. In this case, a very interesting question is how to incorporate the multi-fidelity models with the SMMC algorithm and further improve the computational efficiency. Finally we think the proposed method can also be applied to problems beyond failure probability estimations. In particular, we hope to apply the SMMC algorithm with necessary modifications to evaluate the evidence (normalization constant of the posterior distribution) in Bayesian inference problems. We plan to study these problems in future works.

Acknowledgement

The work was partially supported by the National Natural Science Foundation of China under grant number 11301337.

Appendix A. A specialized MCMC algorithm for the SMMC simulations

Here we present a specialized MCMC algorithm, largely following the modified Metropolis algorithm used in the SS method [6]. First, unlike the standard MMC algorithm which employs only one MCMC chain at each cycle, we use a multichain MCMC algorithm. In particular, in each cycle we randomly select a sample from each $D_1, D_2, \cdots, D_m$ if there are any, and then we use the obtained $m' \leq m$ samples as the seeds to perform m' chains parallely. Note that here m' is automatically determined by the algorithm and for this reason, we can not strictly specify the number of samples drawn in each MMC iteration.

Next we adopt the dimension by dimension proposal used in [6]. To do so, we need to assume that in the original distribution $\pi(\mathbf{x})$ all the components of $\mathbf{x}$ are independent; namely, $\pi(\mathbf{x})$ can be written as,

$$\pi(\mathbf{x}) = \prod_{i=1}^{d} \phi_i(x_i).$$

We use the following algorithm to generate another sample $\mathbf{x}^*$ from the MMC biasing distribution $f(\cdot)$.

(1) For $i = 1, \cdots, d$, sample $\xi_i \sim q_i(\cdot|x_i)$, where $q_i(\cdot)$ is a univariate PDF for ξ_i centered at x_i with the symmetry property $q_i(\xi_i|x_i) = q_k(x_i|\xi_i)$.

(2) Compute the acceptance probability $r_i = \min\{1, \phi_i(\xi_i)/\phi_i(x_i)\}$ for $i = 1, \cdots, d$, and then determine the i-th coordinate of the candidate sample by accepting or rejecting ξ_i according to,

$$\zeta_i = \begin{cases} \xi_i, & \text{with probability } r_i; \\ x_i, & \text{with probability } 1 - r_i. \end{cases} \tag{A.1}$$

(3) Compute the final acceptance probability $r^* = \min\{1, \Theta(\mathbf{x})/\Theta(\boldsymbol{\zeta})\}$, and accept or reject the possible sample $\boldsymbol{\zeta}$ according to

$$\mathbf{x}^* = \begin{cases} \boldsymbol{\zeta}, & \text{with probability } r^*; \\ \mathbf{x}, & \text{with probability } 1 - r^*. \end{cases} \tag{A.2}$$

The ergodicity of the modified MCMC algorithm can be proved using the same arguments of [6] and so is omitted here.

References

[1] A.J. McNeil, R. Frey, P. Embrechts, Quantitative Risk Management: Concepts, Techniques and Tools, Princeton University Press, 2015.
[2] R.E. Melchers, Structural Reliability Analysis and Prediction, John Wiley & Son Ltd., 1999.
[3] M.A. Valdebenito, G.I. Schuëller, A survey on approaches for reliability-based optimization, Struct. Multidiscip. Optim. 42 (5) (2010) 645–663.
[4] X. Du, W. Chen, A most probable point-based method for efficient uncertainty analysis, J. Design Manuf. Autom. 4 (1) (2001) 47–66.
[5] G. Schueller, H. Pradlwarter, P. Koutsourelakis, A critical appraisal of reliability estimation procedures for high dimensions, Probab. Eng. Mech. 19 (2004) 463–474.
[6] S. Au, J. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probab. Eng. Mech. 16 (2001) 263–277.
[7] S. Au, J. Beck, Subset simulation and its application to seismic risk based on dynamic analysis, J. Eng. Mech. 129 (8) (2003) 901–917.
[8] R. Rubinstein, D. Kroese, The Cross-Entropy Method, Springer Science+Business Media, Inc., New York, NY, 2004.
[9] H. Wang, X. Zhou, A cross-entropy scheme for mixtures, ACM Trans. Model. Comput. Simul. 25 (1) (2015) 6.
[10] O. Cappe, A. Guillin, J.M. Marin, C.P. Robert, Population Monte Carlo, J. Comput. Graph. Stat. 13 (4) (2004) 907–929.
[11] B.A. Berg, T. Neuhaus, Multicanonical algorithms for first order phase transitions, Phys. Lett. B 267 (2) (1991) 249–253.
[12] B.A. Berg, T. Neuhaus, Multicanonical ensemble: a new approach to simulate first-order phase transitions, Phys. Rev. Lett. 68 (1) (1992) 9.
[13] R. Holzlööhner, C.R. Menyuk, Use of multicanonical Monte Carlo simulations to obtain accurate bit error rates in optical communications systems, Opt. Lett. 28 (20) (2003) 1894–1896.
[14] D. Yevick, Multicanonical communication system modeling-application to PMD statistics, IEEE Photonics Technol. Lett. 14 (11) (2002) 1512–1514.
[15] K. Wu, J. Li, A surrogate accelerated multicanonical Monte Carlo method for uncertainty quantification, J. Comput. Phys. 321 (2016) 1098–1109.
[16] S. Au, J. Ching, J. Beck, Application of subset simulation methods to reliability benchmark problems, Struct. Saf. 29 (3) (2007) 183–193.
[17] J.L. Beck, K.M. Zuev, Rare event simulation, arXiv preprint, arXiv:1508.05047.
[18] K. Zuev, Subset simulation method for rare event estimation: an introduction, arXiv preprint, arXiv:1505.03506.
[19] A. Bononi, L. Rusch, A. Ghazisaeidi, F. Vacondio, N. Rossi, et al., A fresh look at multicanonical Monte Carlo from a telecom perspective, in: Global Telecommunications Conference, 2009, IEEE, 2009, pp. 1–8.
[20] B.A. Berg, Introduction to multicanonical Monte Carlo simulations, Fields Inst. Commun. 26 (1) (2000) 1–24.
[21] B.A. Berg, Markov Chain Monte Carlo Simulations and Their Statistical Analysis: With Web-Based Fortran Code, World Scientific, 2004.
[22] Y. Iba, N. Saito, A. Kitajima, Multicanonical MCMC for sampling rare events: an illustrative review, Ann. Inst. Stat. Math. 66 (3) (2014) 611–645.
[23] D.P. Landau, K. Binder, A Guide to Monte Carlo Simulations in Statistical Physics, Cambridge University Press, 2014.
[24] J. Li, J. Li, D. Xiu, An efficient surrogate-based method for computing rare failure probability, J. Comput. Phys. 230 (2011) 8683–8697.
[25] Y.-T. Wu, H. Millwater, T. Cruse, Advanced probabilistic structural analysis method for implicit performance functions, AIAA J. 28 (9) (1990) 1663–1669.
[26] J.Y. Wong, Theory of Ground Vehicles, John Wiley & Sons, 2001.
[27] J. Li, D. Xiu, Evaluation of failure probability via surrogate models, J. Comput. Phys. 229 (2010) 8966–8980.
[28] V. Dubourg, B. Sudret, F. Deheeger, Metamodel-based importance sampling for structural reliability analysis, Probab. Eng. Mech. 33 (2013) 47–57.
[29] V. Papadopoulos, D.G. Giovanis, N.D. Lagaros, M. Papadrakakis, Accelerated subset simulation with neural networks for reliability analysis, Comput. Methods Appl. Mech. Eng. 223 (2012) 70–80.