

An Adaptive Subset Simulation Algorithm for System Reliability Analysis with Discontinuous Limit States



Jianpeng Chan, Iason Papaioannou, and Daniel Straub

Abstract Efficient computational methods for system reliability assessment are of importance in many contexts, where crude Monte Carlo simulation is inefficient or infeasible. These methods include a variety of importance sampling techniques as well as subset simulation. Most of these methods function in an adaptive manner, whereby the sampling density gradually approaches the failure domain. The adaptation can work well when the limit state function describing system performance is continuous. However, many system reliability problems involve limit state functions that are non-continuous over the input sample space. Such situations occur in both connectivity- and flow-based problems, due to the binary or multi-state random variables entering the definition of the system performance or the discontinuous nature of the performance function. When solving this kind of problem, the standard subset simulation algorithm with fixed intermediate conditional probability and fixed number of samples per level can lead to significant errors, since the discontinuity of the output can result in an ambiguous definition of the sought percentile of the samples and, hence, of the intermediate domains. In this paper, we propose an adaptive subset simulation algorithm to determine the reliability of systems with discontinuous limit state functions. The proposed algorithm chooses the number of samples and the conditional probability adaptively. Numerical examples are provided to demonstrate the accuracy and efficiency of the proposed algorithm.

Keywords Subset simulation · System reliability analysis · Discontinuous limit state

1 Introduction

Infrastructure networks, such as power grids and water supply systems, deliver essential services to society. Failures of such networks can have severe consequences. Quantification of the probability of survival or, conversely, the probability of failure

J. Chan (✉) · I. Papaioannou · D. Straub
Engineering Risk Analysis Group, Technische Universität München, München, Germany
e-mail: jianpeng.chan@tum.de

of such systems is essential in understanding and managing their reliability; this is the main purpose of network system reliability assessment.

The performance of the system can be assessed by the limit state function (LSF), also known as performance function, $g(\mathbf{X})$. \mathbf{X} is an n -dimensional vector of random variables with joint cumulative distribution function (CDF) F_X and represents the uncertainty in the model. By convention, failure of the system occurs for all system states \mathbf{x} for which $g(\mathbf{x}) \leq 0$. That is, $g(\mathbf{x})$ represents the ‘distance’ between the system state \mathbf{x} and the failure surface, and hence can be regarded as the safety margin of state \mathbf{x} . The probability of failure of the system is defined as

$$P_f \triangleq P(g(\mathbf{X}) \leq 0) = \int_{g(\mathbf{x}) \leq 0} dF_X(\mathbf{x}). \quad (1)$$

Unlike in structural system reliability analysis, the vector of basic random variables \mathbf{X} entering the definition of the LSF of network systems usually contains discrete random variables, which results in a discontinuous LSF. This is due to the fact that the performance of the network is often calculated through a function of a large number of binary or multi-state components. Moreover, real-world infrastructure networks are often designed to be highly reliable. This leads to high-dimensional reliability assessment problems with small failure probabilities [1].

Many methods have been proposed for evaluating system reliability, among which sampling based methods such as Monte Carlo simulation (MCS) and its variants feature prominently. For rare events simulation, crude MCS is inefficient and often infeasible when the LSF is expensive to compute. This is because the coefficient of variation of the crude Monte Carlo estimate is $\sqrt{(1 - P_f)/NP_f}$, with N denoting the sample size, and as a result, for small P_f the required sample size for an accurate estimate is very large. Therefore, advanced sampling techniques have been developed that decrease the required number of LSF evaluations for obtaining an accurate probability estimate. These techniques include a variety of importance sampling methods [2, 3] as well as subset simulation [4]. They mostly function in an adaptive manner, whereby the sampling gradually approaches the failure domain. The basic idea of importance sampling is to sample from a proposal distribution under which the rare event is more likely to happen and to correct the resulting bias in the estimate by multiplying each sample contribution in the estimator with the appropriate likelihood ratio [5]. In contrast, subset simulation expresses the probability of failure as a product of larger conditional probabilities of a set of intermediate nested events. This requires sampling conditional on the intermediate events, which is performed with Markov Chain Monte Carlo (MCMC) methods [6]. Subset simulation can be viewed as a generalization of the splitting method for static rare event simulation [7].

In the standard subset simulation algorithm [4], the intermediate failure events are chosen adaptively, so that the estimates of the conditional probabilities equal a predefined value p_0 . This is achieved through generating a fixed number of samples in each conditional level, sorting the samples according to their LSF values and

determining the p_0 -percentile of the samples, which is set as the threshold defining the next failure event.

When solving network reliability problems, the discontinuous nature of the LSF can result in a large number of samples in a certain conditional level having the same LSF values. In such cases, the standard fixed effort subset simulation method will result in an ambiguous definition of the intermediate domains. In extreme conditions, all samples generated in a certain level might have the same LSF value, in which case the sample process can get stuck and might not reach the failure domain.

To address this issue, we introduce a novel variant of subset simulation, which chooses the number of samples per level and the respective conditional probability adaptively to ensure that an adequate number of samples fall in the subsequent intermediate domain. The performance of the method is illustrated by two numerical examples, a one-dimensional multi-state problem and a benchmark transmission power network system.

2 Brief Introduction of Standard Subset Simulation

2.1 Brief Introduction of Subset Simulation

The basic idea of subset simulation (or generalized splitting) is to express the rare failure event F as the intersection of a series of nested intermediate events $F_1 \supset F_2 \supset \dots \supset F_m = F$. The failure probability is then written as

$$P(F) = \prod_{i=1}^m P(F_i|F_{i-1}) \tag{2}$$

where F_0 is the certain event. Ideally, the intermediate events are selected such that each conditional probability is appropriately large. In this way, the original problem of estimating a small probability is transformed to a sequence of m intermediate problems of evaluating larger conditional probabilities.

The estimation of each conditional probability $P(F_i|F_{i-1})$ requires sampling from the distribution of the random variables conditional on F_{i-1} , denoted $Q(\mathbf{x}|F_{i-1})$, where $Q(\mathbf{x}|F_0) = F_X(\mathbf{x})$. $Q(\mathbf{x}|F_0)$ can be sampled by standard Monte Carlo sampling, but the distributions $Q(\cdot|F_i), i > 0$, are only known point-wise up to a normalizing constant and, hence, cannot be sampled directly. Therefore MCMC sampling is employed as an alternative. The sampling process in the j th sampling level is performed as follows: (1) Select the samples $\mathcal{P}^{(j-1)}$ from the $(j-1)$ -th level that fall in F_j as the seeds $S^{(j)}$. ($\mathcal{P}^{(0)}$ are generated through Monte Carlo sampling) (2) From each seed, start a Markov chain that has the target distribution $Q(\cdot|F_j)$ as the stationary distribution, and record all the states as new samples $\mathcal{P}^{(j)}$. (3) Take the samples $\mathcal{P}^{(j)}$ located in F_{j+1} as new seeds $S^{(j+1)}$ and estimate $P(F_{j+1}|F_j)$. The above three steps are repeated successively until F is approached. We note that the

number of the samples per level or the sampling effort $\text{card}(\mathcal{P}^{(j)})$ is usually fixed prior to the analysis.

Defining the intermediate events a priori is typically challenging. Hence, in standard subset simulation the intermediate failure events are chosen adaptively during the simulation such that each conditional probability equals a predefined constant p_0 . This standard subset simulation approach is also termed (fixed effort) adaptive multilevel splitting [7]. In this variant, step (3) in the above sampling process is modified as follows: Order the samples $\mathcal{P}^{(j)}$ by their safety margins. The first p_0 -percent of these sorted samples are then taken as seeds for the next sampling level and the safety margin of the p_0 -percentile $g(\mathbf{x}_0^{(j)})$ is used to define the boundary of the intermediate domain, such that $F_{j+1} = \{\mathbf{x} : g(\mathbf{x}) \leq g(\mathbf{x}_0^{(j)})\}$.

Various MCMC algorithms are proposed for constructing the Markov chains for subset simulation. These include the modified (or component-wise) Metropolis-Hasting [4] and the conditional sampling (CS) [6] algorithms, both developed to tackle high-dimensional problems. In this paper, we adopt the adaptive conditional sampling (aCS) as the MCMC algorithm [6]. This method is remarkably simple since it no longer involves the explicit choice of a proposal distribution [8]. Instead it adaptively tunes the correlation between candidate and current samples to achieve a near-optimal acceptance probability [6]. aCS is proposed for sampling in standard normal space, hence, it is necessary to transform the original sample space of \mathbf{X} and define the reliability problem in standard normal space. This can be achieved by the Rosenblatt transformation [9]. We discuss this transformation in the next section, focusing on its implementation for discrete original sample spaces, which is particularly relevant for network reliability assessment. However, the proposed adaptive effort subset simulation algorithm can be implemented with any MCMC algorithm, including those that work in the original sample space.

2.2 Implementation in Standard Normal Space

Let \mathbf{U} denote an n -dimensional random vector that has the independent standard normal distribution. One can define the reliability problem in the \mathbf{U} -space through an isoprobabilistic mapping $\mathbf{T} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that

$$P_f = \text{P}(g(\mathbf{X}) \leq 0) = \text{P}(G(\mathbf{U}) \leq 0) = \int_{G(\mathbf{u}) \leq 0} \varphi_n(\mathbf{u}) d\mathbf{u} \quad (3)$$

where $G(\mathbf{U}) = g(\mathbf{T}(\mathbf{U}))$ and $\varphi_n(\mathbf{u})$ is the independent standard normal joint probability density function (PDF). The mapping $\mathbf{T}(\cdot)$ can be obtained by the Rosenblatt transformation, which is implemented as follows

$$\begin{aligned}
 x_1 &= F_{X_1}^{-1}(\Phi(u_1)) \\
 x_2 &= F_{X_2|x_1}^{-1}(\Phi(u_2)) \\
 &\vdots \\
 x_m &= F_{X_m|x_1, \dots, x_{m-1}}^{-1}(\Phi(u_m))
 \end{aligned}
 \tag{4}$$

where Φ represents the CDF of standard normal distribution and $F_{X_i|x_1, \dots, x_{i-1}}(\cdot)$ denotes the conditional CDF of X_i given $X_1 = x_1, \dots, X_{i-1} = x_{i-1}$. If any subset of X consists of discrete random variables, then it is possible that the functions $F_{X_i|x_1, \dots, x_{i-1}}(\cdot)$ are not strictly invertible. Therefore, we use the following extended definition of the inverse of a CDF

$$F^{-1}(a) = \inf(x : F(x) \geq a)
 \tag{5}$$

We note that in such cases the Rosenblatt transformation is not one-to-one and, hence, the inverse mapping from X to U is not uniquely defined.

2.3 Statistics of the Subset Simulation Estimator

Assume that the intermediate events are defined prior to the simulation. In the Monte Carlo level, samples $\mathcal{P}^{(0)}$ are generated from $Q(\cdot|F_0)$ independently, and therefore the corresponding seeds $\mathcal{S}^{(1)}$ follow distribution $Q(\cdot|F_1)$. This will lead to so called perfect sampling when simulating the Markov chains in the next level. Since the chains have already reached the stationary state at the beginning, no burn-in period is needed, and all the samples $\mathcal{P}^{(1)}$ will follow $Q(\cdot|F_1)$. In this way, all samples $\mathcal{P}^{(j)}$ generated in any j -th level will follow the target distribution $Q(\cdot|F_j)$ and the corresponding estimator of the conditional probability $\hat{P}(F_{j+1}|F_j)$ will be unbiased. Moreover, [7] proves that the resulting failure probability estimator $\hat{P}_f(F)$ is also unbiased if both intermediate events and length of the Markov chain are predefined, i.e. if they are independent of the simulation process.

Since the intermediate events are usually selected adaptively and as a result, samples $\mathcal{P}^{(j)}$ will not completely follow the target distribution. Both conditional probability estimator and failure probability estimator will be slightly biased. Nevertheless, compared to the variance of the estimator, the squared bias is one order of magnitude smaller [4] and, hence, its contribution to the mean-square error (MSE) of the estimator is negligible, since the latter can be decomposed as

$$\text{MSE}(\hat{P}_f) = \text{Var}(\hat{P}_f) + \left(P_f - \mathbb{E}(\hat{P}_f) \right)^2
 \tag{6}$$

In other words, the error of the subset simulation is mainly due to the variance of the failure probability estimator rather than the bias. The most common and reliable way to calculate the variance $\text{Var}(\hat{P}_f)$ is to run subset simulation several times and

to use the sample variance as the unbiased estimation of the $\text{Var}(\hat{P}_f)$. One can also evaluate the variance approximately through a single run of the subset simulation. More details can be found in [4, 6].

3 Adaptive Effort Subset Simulation Method

In each conditional level of the subset simulation method with fixed number of samples N and adaptive estimation of the intermediate events, the p_0 -percentile of the safety margins of the samples $\mathcal{P}^{(j)}$, $g(\mathbf{x}_0^{(j)})$ is used to define the boundary of the intermediate domain. This works well when only a few samples are located on the boundary $g(\mathbf{x}) = g(\mathbf{x}_0^{(j)})$, i.e. a few samples have the same LSF value as the p_0 -percentile. However, it may happen that many samples fall on this boundary, particularly in either of the following cases:

1. $\mathbf{1X}$ includes discrete random variables.
2. The LSF is defined such that the probability measure of the set $\{\mathbf{x} : g(\mathbf{x}) = g(\mathbf{x}_0^{(j)})\}$ is positive.

The parameters of the MCMC algorithm are inappropriately set, resulting in the candidates being rejected successively many times.

While case (3) can be avoided by an appropriate implementation of the algorithm, cases (1) and (2) are common in the context of network reliability assessment. This will result in an ambiguous definition of the intermediate domain F_{j+1} and can lead to an inaccurate estimate of the failure probability. In extreme situations, all samples generated in a certain level will have the same LSF value and the sample process can get stuck and never reach the failure domain.

To circumvent this problem and provide a clear (unambiguous) definition of the intermediate domains, we propose to discard all the samples on the boundary $g(\cdot) = g(\mathbf{x}_0^{(j)})$, and redefine the intermediate event F_{j+1} as $F_{j+1} = \{\mathbf{x} : g(\mathbf{x}) < g(\mathbf{x}_0^{(j)})\}$. Then, we calculate the number of samples that fall in the domain F_{j+1} (number of the seeds). If this number is smaller than a predefined constant, we increase the sampling effort and append $\mathcal{P}^{(j)}$ with new samples. With a fixed F_{j+1} and increasing number of samples, the number of the seeds will keep increasing until the desired threshold is achieved. By doing this, for any state \mathbf{x} in F_{j+1} , there exists $g(\mathbf{x}) < g(\mathbf{x}_0^{(j)}) < g(\mathbf{x}_0^{(j-1)})$, and thus $F_{j+1} \subset F_j$ is always true, which avoids a degeneracy of the sampling process. Even in the extreme case where all the samples in $\mathcal{P}^{(j)}$ have the same safety margin, the sampling process will keep moving forward towards the failure domain and will no longer get stuck in this level as in the standard subset simulation algorithm. Unlike standard subset simulation, the number of samples per level (sampling effort) is adapted throughout the simulation. We, hence, term the proposed approach *adaptive effort subset simulation method*.

In the following, we discuss the proposed adaptive effort subset simulation algorithm for implementation in standard normal space. The samples in each intermediate level are generated with the aCS algorithm. In addition to the initial number of samples per level N_0 and conditional probability p_0 , the algorithm requires the choice of the parameter $tol \in (0, 1)$ that defines the minimum number of seeds through $tol \cdot N_0 \cdot p_0$. We have found that $tol \in (0.5, 0.8)$ is a good choice.

Adaptive effort subset simulation algorithm

$tol \in (0,1)$: Parameter which limits the minimum number of seeds inside each intermediate domain.

$p_0 \in (0,1)$: Parameter which represents the initial conditional probability when starting the iteration.

N_0 : Parameter which represents initial number of samples when starting the iteration.

Set $l = 0, b^{(0)} = \infty$.

while $b^{(l)} > 0$ **do**

 Set $iter = 1, N^{(l,iter)} = N_0$ and $N_s^{(l,iter)} = 0$.

while $N_s^{(l,iter)} < tol \cdot N_0 \cdot p_0$ **do**

if $l = 0$

 Sample $N^{(l,iter)}$ samples $\{\mathbf{u}_i^{(l,iter)}\}_{i=1}^{N^{(l,iter)}}$ from the standard normal distribution $\varphi_n(\mathbf{u})$.

else

 Sample $N^{(l,iter)}$ samples $\{\mathbf{u}_i^{(l,iter)}\}_{i=1}^{N^{(l,iter)}}$ starting from the seeds $S^{(l)}$ with the aCS algorithm.

end if

 Calculate the LSF values of the samples $\{G(\mathbf{u}_i^{(l,iter)})\}_{i=1}^{N^{(l,iter)}}$ and sort the samples, such that $G(\mathbf{u}_{i_j}^{(l,iter)}) \leq G(\mathbf{u}_{i_{j+1}}^{(l,iter)}), \forall j \in \{1, \dots, N^{(l,iter)} - 1\}$. Denote the sorted samples as $\{\bar{\mathbf{u}}_i^{(l,iter)}\}_{i=1}^{N^{(l,iter)}}$ and let $b^{(l+1)} = G(\bar{\mathbf{u}}_{p_0 N_0}^{(l,1)})$.

If $b^{(l+1)} \leq 0$

 Set $b^{(l+1)} = 0$ and $N_s^{(l,iter)} \triangleq \sum_j \mathbb{I}\{G(\bar{\mathbf{u}}_j^{(l,iter)}) \leq 0\}$.

else

$N_s^{(l,iter)} \triangleq \sum_j \mathbb{I}\{G(\bar{\mathbf{u}}_j^{(l,iter)}) < b^{(l+1)}\}$.

end if

If $N_s^{(l,iter)} < tol \cdot N_0 \cdot p_0$

$N^{(l,iter+1)} = 2 \cdot N^{(l,iter)}$.

$iter = iter + 1$.

end if

end while

 Define the intermediate failure event: $F^{(l+1)} \triangleq \{\mathbf{u}: G(\mathbf{u}) < b^{(l+1)}\}$.

 Take the $S^{(l+1)} \triangleq \{\bar{\mathbf{u}}_j^{(l,iter)}\}_{j=1}^{N_s^{(l,iter)}}$ as the seeds for the next level.

 Calculate the conditional probability estimator: $\hat{\mathbb{P}}(F^{(l+1)}|F^{(l)}) = \frac{N_s^{(l,iter)}}{N^{(l,iter)}}$.

$l = l + 1$.

end while

Estimate the failure probability:

$$\hat{\mathbb{P}}(F) = \hat{\mathbb{P}}(F^{(1)}) \prod_{j=2}^l \hat{\mathbb{P}}(F^{(j)}|F^{(j-1)}).$$

4 Examples

4.1 Multistate Random Variable

Consider a discrete random variable X with 7 states $\{x_1, \dots, x_7\}$. We consider two cases. In case 1, the CDF of X $F(\cdot)$ is set, such that $F(x_i)/F(x_{i+1}) \geq 0.1$, while in case 2, there is a big ‘jump’ between the third and the fourth state, i.e. $F(x_3)/F(x_4) \approx 1.67 \cdot 10^{-3}$. The CDF of X for the two considered cases is given in Table 1. The LSF is defined as $g(X) = X + 4$ such that the failure probability $P(X < -4)$ equals 10^{-5} for the first case and $2 \cdot 10^{-5}$ for the second.

We implement standard subset simulation (SuS) and the proposed adaptive effort subset simulation (aE-SuS) respectively to evaluate the failure probability. For SuS, the sampling effort is fixed to 1000, and the conditional probability is 0.1. For aE-SuS, the parameters are set to be $tol = 0.5$, $N_0 = 1000$, $p_0 = 0.1$. Each method is run 1000 times to get the relative bias, coefficient of variation and average computation cost of the failure probability estimator. The results for case 1 and case 2 are shown in Tables 2 and 3 separately. In both cases, aE-SuS shows good accuracy, a negligible bias and a much smaller variance than the crude Monte Carlo results (shown in the red brackets). We note that the coefficient of variation of crude Monte Carlo is given for the same computational effort as the proposed aE-SuS method. In contrast, SuS gives the wrong estimate of the failure probability in the first case and falls into a dead loop in the second case.

Table 1 CDF of X

State	-6	-5	-3	-2	-1	0	1
CDF (case1)	1e-5	1e-4	1e-3	1e-2	1e-1	5e-1	1
CDF (case2)	1e-5	2e-5	5e-5	3e-2	1e-1	5e-1	1

Table 2 Statistical characteristics of the estimator (Case 1)

	Relative bias%	Coefficient of variation	Average computation effort
SuS	-97.8	3.747	7222
aE-SuS	-3.5	0.377(3.580)	7804

Table 3 Statistical characteristics of the estimator (Case 2)

	Relative bias%	Coefficient of variation	Average computation effort
SuS	/	/	/
aE-SuS	1.4	0.206(0.760)	86561

4.2 Power Network System

A transmission power network with the same topology as the IEEE39 bus benchmark system is considered here. It consists of 39 nodes and 43 weighted edges, whose weights represent the line reactance values. This example was previously investigated by Scherb et al. [10] to quantify the network reliability considering cascading effects and spatially distributed hazards, and by Rosero-Velasquez and Straub [11] to select representative failure scenarios. More details about the example can be found in these references.

The state of each node is considered as an independent Bernoulli random variable, with component failure probability set to 10^{-3} . The LSF is then defined as the function of the system state \mathbf{x} , which is a binary vector, as follows:

$$g(\mathbf{x}) = \frac{E(\mathbf{x})}{E(\mathbf{1})} - threshold \quad (7)$$

$$E(\mathbf{x}) = \frac{1}{|SN||TN|} \sum_{\substack{s \in SN, t \in TN \\ t \neq s}} eff_{st} \quad (8)$$

eff_{st} is the efficiency of the most efficient path from source node s to terminal node t and can be evaluated by adding up the reciprocals of the reactance values along that path. $E(\mathbf{x})$ is the efficiency of the whole system associated to the system state \mathbf{x} (where the vector $\mathbf{1}$ is the intact system state) and is equal to the mean value of all the eff_{st} from each source node in set SN to each terminal nodes in set TN .

In order to model the cascading effects, Eq. (7) is modified to

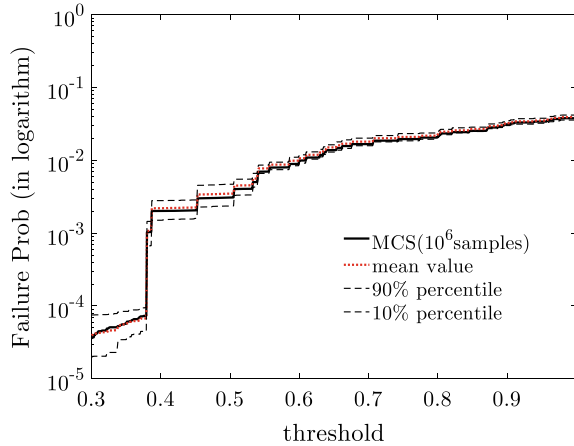
$$g(\mathbf{x}) = \frac{E(\tilde{\mathbf{x}})}{E(\mathbf{1})} - threshold \quad (9)$$

where $\tilde{\mathbf{x}}$ is the final system state after cascading effects. These are triggered by overloading in individual lines following initial failures, and are modeled following [10, 12].

The threshold is fixed to 0.3, which means the system fails when its efficiency is less than 30% of that of the intact system. We then apply aE-SuS algorithm to this problem and set the parameters $N = 2000$, $p_0 = 0.1$, $tol = 0.8$. Figure 1 shows the empirical CDF of $g(\mathbf{X})$ obtained by Monte Carlo Simulation and the aE-SuS algorithm respectively. The aE-SuS algorithm is run 25 times to obtain the mean value, 10 percentile and 90 percentile of the empirical CDF, while a single Monte Carlo Simulation run with 10^6 samples is carried out for validation.

The average computation cost of aE-SuS is 24437 calculations of the LSF $g(\cdot)$ and the relative bias of the failure probability is 9.17%, while the coefficient of variation is 0.57. Under the same computation cost, the coefficient of variation of

Fig. 1 Results obtained by aE-SuS for IEEE39 network.



Monte Carlo Simulation is about 1.06 which is significantly larger than that of aE-SuS. The standard SuS algorithm is not applicable for this example due to the large jump in the CDF of the LSF below the threshold of 0.4.

5 Conclusions

We introduce an adaptive effort subset simulation algorithm that enables solving reliability problems with discontinuous limit states. Such problems often occur in network reliability assessment because of discrete random variables appearing in the input random vector or due to discontinuities in the function that defines the system performance. The proposed algorithm extends the applicability of standard subset simulation to problems where significant jumps in the distribution of the limit-state function occur. Numerical examples demonstrate the accuracy and efficiency of the proposed method and show that the method has increased efficiency compared to crude Monte Carlo in some problems where standard subset simulation fails to converge. For connectivity-based problems where the LSF can only take two possible values, the proposed algorithm will turn to crude Monte Carlo simulation and therefore becomes inefficient in rare event context.

References

1. Zuev, K. M., Wu, S., & Beck, J. L. (2015). General network reliability problem and its efficient solution by subset simulation. *Probabilistic Engineering Mechanics*, 40, 25–35.
2. Papaioannou, I., Geyer, S., & Straub, D. (2019). Improved cross entropy-based importance sampling with a flexible mixture model. *Reliability Engineering & System Safety*, 191, 106564.

3. Rubinstein, R. Y., & Kroese, D. P. (2016). *Simulation and the Monte Carlo Method*. Hoboken: Wiley.
4. Au, S. K., & Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4), 263–277.
5. Rubino, G., & Tuffin, B. (2009). *Rare Event Simulation Using Monte Carlo Method*. Chichester: Wiley.
6. Papaioannou, I., Betz, W., Zwirgmaier, K., & Straub, D. (2015). MCMC algorithms for subset simulation. *Probabilistic Engineering Mechanics*, 41, 89–103.
7. Botev, Z. I., & Kroese, D. P. (2012). Efficient Monte Carlo simulation via the generalized splitting method. *Statistics and Computing*, 22(1), 1–16.
8. Au, S. K. (2016). On MCMC algorithm for subset simulation. *Probabilistic Engineering Mechanics*, 43, 117–120.
9. Rosenblatt, M. (1952). Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3), 470–472.
10. Scherb, A., Garrè, L., & Straub, D. (2017). Reliability and component importance in networks subject to spatially distributed hazards followed by cascading failures. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 3(2), 021007.
11. Rosero-Velasquez, H., & Straub, D. (2019). Representative natural hazard scenarios for risk assessment of spatially distributed. In *Proceedings of the 29th European Safety and Reliability Conference* (pp. 1–7). Submitted.
12. Crucitti, P., Latora, V., & Marchiori, M. (2004). Model for cascading failures in complex networks. *Physical Review E*, 69(4), 045104.