Computation of Failure Probability via Hierarchical Clustering

Chao Yin, Ahsan Kareem

Nathaz Modeling Laboratory, University of Notre Dame

Abstract

Most of the recent reliability analysis methods are based on supervised learning approaches, e.g., neural network, support vector machine, importance sampling (IS), Markov Chain Monte Carlo (MCMC), subset simulation (SS), etc. Among these, SS approach has been shown to outperform in effectiveness and robustness, as it classifies failure samples through multiple levels instead of conventional use of a single level. Inherent to supervised learning approaches is the limitation that these only utilize the evaluated information, i.e. the information from the samples whose performance functions have been computed. To overcome this, a new scheme based on modified hierarchical clustering is proposed. This scheme, referred to here as ‘hierarchical failure clustering’ (HFC), adds a pre-processing step via an unsupervised learning approach, e.g. various clustering algorithms. The proposed method exploits the statistical structure embedded in the input samples prior to computing the performance functions of these samples. This statistical structure has a tree based on these input samples with nodes as multi-level clusters. Father clusters at a specific level are used to explore an intermediate failure region and the failure portions are sifted out. Child clusters that are attached to the father failure clusters continue to explore a smaller intermediate failure region. Accordingly, the HFC method proceeds recursively until the target failure region is detected. In contrast with the SS and IS, HFC benefits from additional information from unlabeled samples with the help of a pre-built tree. Through a relatively small added premium in terms of computations for pre-processing, the additional information helps to improve the quality of the intermediate failure prob-

ability estimates and thereby significantly reduces the overall computational effort. The efficacy of HFC is theoretically demonstrated here and supported by several examples. HFC holds the promise for complex and large-scale systems for which the performance evaluation is computationally intensive.

Keywords: Reliability, Failure probability, Monte Carlo simulation, Subset simulation, Unsupervised learning, Hierarchical clustering

1. Introduction

Reliability analysis aims to obtain the probability of failure of an event that is defined as

$$P(F) = \int_F p(\boldsymbol{\theta}) d\boldsymbol{\theta} = \int_{\Omega_{\boldsymbol{\theta}}} I_F(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (1)$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_d] \in \Omega_{\boldsymbol{\theta}} \subset \mathbb{R}^d$ represents an input random vector with probability density function (PDF) $p(\boldsymbol{\theta})$; F is the failure region defined on $\Omega_{\boldsymbol{\theta}}$, constrained by a limit state function $g(\boldsymbol{\theta})$, i.e. $F = \{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}} : g(\boldsymbol{\theta}) < 0\}$; I_F is an indicator function, that is $I_F(\boldsymbol{\theta}) = 1$ if $\boldsymbol{\theta} \in F$ and $I_F(\boldsymbol{\theta}) = 0$ otherwise. The estimation of small failure probability poses many challenges in the case of a large-scale engineering application, when each sample requires considerable computational effort. Indeed, if direct sampling-based methods are used, whose popularity is due to their robustness and straightforward implementation, a large number of simulations are needed to ensure the fidelity of the failure probability estimate, and this can be computationally demanding. On the other hand, when a computationally less intensive response surface scheme is adopted to approximate the limit state function using the first-order or second-order approximation FORM/SORM (Thoft-Christensen and Baker, 1982; Faravelli, 1989; Bucher and Bourgund, 1990; Rajashekhar and Ellingwood, 1993), the accuracy in the estimate of the failure probability can be insufficient, especially in the case of limit state functions with large curvatures or multiple design points (Der Kiureghian and Dakessian, 1998). Also, non-sampling-based integra-

tion methods, such as the sparse-grid quadrature (Smolyak, 1963; Gerstner and Griebel, 1998), can fall short when the failure domain of input random variables is complicated.

In the case of sampling-based methods, one way of reducing their computational effort, especially in the case of small failure probabilities, is given by the use of variance reduction strategies with advanced sampling schemes, which have the aim of limiting the number of simulations without losing accuracy. These methods can be categorized as importance sampling-based methods (Rubinstein and Kroese, 2011; Robert and Casella, 2004), splitting and subset simulation (SS) methods (Ching et al., 2005; Au and Beck, 2001), and methods that focus on improving the quality of the samples, such as quasi-Monte Carlo (Fang and Wang, 1994; Nie and Ellingwood, 2004; Li and Chen, 2008), common random numbers (Hammersley and Handscomb, 1964), antithetic variates (Hammersley and Morton, 1956; Kareem, 1990).

Importance sampling (IS) aims to reduce the redundant samples by shifting the importance density towards the failure region (Rubinstein and Kroese, 2011; Robert and Casella, 2004; Shinozuka, 1983). The determination of importance density is crucial to IS. To achieve this goal, various IS techniques have been developed, which include (1) establishing the importance density via statistical moments estimation (Bucher, 1988) or kernel density estimation (KDE) based on pre-samples (Ang et al., 1992); (2) seeking the near-optimal IS density by minimizing the Kullback-Leibler cross entropy (CE) based on pre-samples (Rubinstein and Kroese, 2004; Kurtz and Song, 2013). Undoubtedly, a strategy is more effective when the number of pre-samples needed to identify the optimal importance density is low. However, this is not a trivial task. Uni-mode approximation tends to fail or lose effectiveness where multiple failure regions exist. Multi-mode approximations, such as the Gaussian mixture model (GMM) or KDE, face common concerns of how to determine the choice of number of kernels and how to determine the shape and weight of each kernel, which may cause overfitting or underfitting problems. This issue remains a challenge and is usually addressed manually with experience.

SS reduces the number of samples by narrowing down the failure region successively (Au and Beck, 2001). In SS, the failure probability is a product of conditional probabilities of multi-level intermediate failure events. Each intermediate failure event occurs more frequently than the target rare-event and thus requires fewer samples for evaluation. SS adopts Markov-Chain Monte Carlo (MCMC) (Robert and Casella, 2004) in the generation of new samples. However, MCMC still faces the problem of how to choose an appropriate proposal density function. In addition, the samples generated by MCMC are mutually dependent, and hence increase the coefficient of variation (c.o.v.).

A new simulation approach, termed ‘hierarchical failure clustering’ (HFC), is proposed here to compute the failure probabilities. This approach adopts the basic concept of SS by invoking hierarchical clustering methods (Ward Jr, 1963; Murphy, 2012). This method does not need the importance density function or the proposal density function, therefore it is a good candidate for automation. HFC is straightforward to implement in the following steps: (1) a tree structure is first constructed via hierarchical clustering of the raw samples drawn from the input PDF; (2) higher-level father clusters are used to evaluate more frequent failure events, and the failure portions are sifted out as well as their child clusters, which are used to evaluate more rare events; (3) like SS, the target failure probability is expressed by a product of conditional probabilities of multi-level intermediate failure events. In fact, via the pre-built tree, HFC is able to exploit some useful information from the unlabeled samples, while SS and IS only utilize the information from the labeled samples. In this manner, the quality of the simulations can be enhanced since additional information is involved. Details of this approach are illustrated in the following sections.

2. Subset Simulation

Consider the failure probability defined by Eq.1. The integrand

$$p(\boldsymbol{\theta}|F) \propto I_F(\boldsymbol{\theta})p(\boldsymbol{\theta}) \quad (2)$$

represents the distribution of samples falling in the target failure region F . Obviously, if F is a small subset of $\Omega_{\boldsymbol{\theta}}$, P_F should also be small, and hence a large number of simulations are required to ensure that a sufficient number of samples fall in this region. The subset simulation provides an effective way of reducing the number of simulations. Let $\{F_1 \supset F_2 \cdots \supset F_n = F\}$ be a decreasing sequence of failure events. By definition of conditional probability, the failure probability $P(F)$ is expressed as a product of a sequence of conditional probabilities $\{P(F_i|F_{i-1}) : i = 2, 3, \dots, n\}$:

$$\begin{aligned} P(F) &= P\left(\bigcap_{i=1}^n F_i\right) \\ &= P(F_n|F_{n-1}) \cdots P(F_2|F_1)P(F_1) \\ &= P(F_1) \prod_{i=2}^n P(F_i|F_{i-1}) \end{aligned} \quad (3)$$

Thus, evaluating the target failure event amounts to evaluating this sequence of conditional failure events, which are given by

$$P(F_i|F_{i-1}) = \int_{\Omega_{\boldsymbol{\theta}}} I_{F_i}(\boldsymbol{\theta})p(\boldsymbol{\theta}|F_{i-1})d\boldsymbol{\theta} \quad (4)$$

with

$$p(\boldsymbol{\theta}|F_{i-1}) \propto I_{F_{i-1}}(\boldsymbol{\theta})p(\boldsymbol{\theta}) \quad (5)$$

111 Assuming there are N_{i-1} samples $\{\boldsymbol{\theta}_k, k = 1, 2, \dots, N_{i-1}\}$ distributed according to $p(\boldsymbol{\theta}|F_{i-1})$,
 112 i.e. $\boldsymbol{\theta}_k \sim p(\boldsymbol{\theta}|F_{i-1})$, the conditional failure probability $P(F_i|F_{i-1})$ can be estimated by

$$P(F_i|F_{i-1}) \approx \hat{P}(F_i|F_{i-1}) = \frac{1}{N_{i-1}} \sum_{k=1}^{N_{i-1}} I_{F_i}(\boldsymbol{\theta}_k) \quad (6)$$

113 Failure samples are generated successively from distributions $\{p(\boldsymbol{\theta}|F_{i-1}), i = 2, \dots, n\}$
 114 in which the number required at each intermediate level is smaller because each interme-
 115 diate event $F_i|F_{i-1}$ occurs more frequently than the target failure event F . This concept
 116 is straightforward but not trivial to implement. For example, the number of samples
 117 decreases after sifting at each intermediate level and thus needs to be replenished to con-
 118 tinue the subsequent computations. MCMC methods have been used to help SS achieve
 119 this goal (Au and Beck, 2001). As mentioned earlier, MCMC requires the choice of a
 120 proposal PDF which significantly affects the computational efficiency. For example, a
 121 global proposal PDF may not be efficient, while, for a local proposal PDF, samples may
 122 get trapped in a local region and have no opportunity to shift to other regions. In general,
 123 the performance of MCMC relies on the density estimation of the failure probabilities at
 124 every intermediate level. Another similar approach, referred to as the splitting method
 125 (Ching et al., 2005), uses IS instead of MCMC. A common limitation of SS and IS is
 126 that the input PDF reduces to a small number of independent samples and a signifi-
 127 cant amount of information is not exploited. The information gained after each level of
 128 evaluation is only from labeled samples. A possible improvement consists therefore in
 129 exploiting the additional information provided by the unlabeled samples. To this aim, it
 130 is here proposed to construct a tree-based data structure for raw samples drawn from the
 131 input PDF. The number of the raw samples is large but only a small portion needs to
 132 be evaluated. The unlabeled samples also contribute with some information to the anal-
 133 ysis, albeit in an implicit manner. Obviously, methods that can utilize more information
 134 without additional computational cost have the potential to perform better. The central

concept of the proposed approach is presented in the following section.

3. The Method of Hierarchical Failure Clustering

3.1. Basic Concept

The proposed ‘hierarchical failure clustering’ (HFC) for computing failure probabilities embodies the basic idea detailed in the following. Let us suppose that the direct MC requires N samples to compute a target failure probability. With HFC, the same number of samples are used but reorganized as a tree structure in advance. This tree structure is built in such a manner that the most similar pair of samples are merged successively until a single sample remains. At each step, the pair of samples merged belong to the same father cluster, and the merged sample serves as the representative of this cluster. For example, Fig.1 shows a tree built on 4 samples, $\{\theta_1\}$, $\{\theta_2\}$, $\{\theta_3\}$ and $\{\theta_4\}$. At the initial or the lowest level, there are four clusters, and each one contains a single sample, such as $C_{4,1} = \{\theta_1\}$, $C_{4,2} = \{\theta_2\}$, $C_{4,3} = \{\theta_3\}$, $C_{4,4} = \{\theta_4\}$, where C denotes a cluster; the first subscript of C denotes the number of total clusters at a specific level; the last subscript denotes the order of a cluster at this level. We merge the most similar pair of clusters into a new one, based on a similarity function which will be explained later in Section 3.2. For example, if $\{\theta_2, \theta_3\}$ is the most similar pair among all the possible pairs like $\{\theta_1, \theta_2\}$, $\{\theta_1, \theta_3\}$, $\{\theta_1, \theta_4\}$, $\{\theta_2, \theta_3\}$, $\{\theta_2, \theta_4\}$ and $\{\theta_3, \theta_4\}$, then $\{\theta_2\}$ and $\{\theta_3\}$ belongs to the same cluster $\{\theta_2, \theta_3\}$. The new level clusters are therefore given by $C_{3,1} = \{\theta_1\}$, $C_{3,2} = \{\theta_2, \theta_3\}$, $C_{3,3} = \{\theta_4\}$. Now, the mean or medoid of θ_2 and θ_3 is used as the representative sample of cluster $C_{3,2}$, and denoted as $\bar{\theta}_{3,2}$. At this point, the pairs $\{\theta_1, \bar{\theta}_{3,2}\}$, $\{\theta_1, \theta_4\}$ and $\{\bar{\theta}_{3,2}, \theta_4\}$ are compared and the most similar pair is chosen. If $\{\bar{\theta}_{3,2}, \theta_4\}$ is the most similar one, $\{\theta_2\}$, $\{\theta_3\}$ and $\{\theta_4\}$ should belong to the same cluster. By merging $\{\theta_2, \theta_3\}$ and $\{\theta_4\}$, the new level clusters $C_{2,1} = \{\theta_1\}$, $C_{2,2} = \{\{\theta_2, \theta_3\}, \theta_4\}$ are obtained, where the representative sample of $C_{2,2}$ is $\bar{\theta}_{2,2}$, which is the mean or medoid of $\bar{\theta}_{3,2}$ and θ_4 . Obviously, at the highest level, only one cluster exists, that is $C_1 = \{\theta_1, \{\{\theta_2, \theta_3\}, \theta_4\}\}$.

$C_{4,2}$ and $C_{4,3}$ are the child clusters of their father cluster $C_{3,2}$; $C_{3,2}$ and $C_{3,3}$ are the child clusters of their father cluster $C_{2,2}$; $C_{2,2}$ and $C_{2,1}$ are the child clusters of C_1 . The basic assumption of HFC is that the representative sample of a father cluster falls in the same region with the representative samples of its child clusters. In other words, the entire ‘territory’ of a father cluster is the union of ‘territories’ of all its child clusters. For example, if $\bar{\theta}_{2,2}$ falls in the failure region F_1 , the representative samples of the child clusters, $\bar{\theta}_{3,2}$ and $\bar{\theta}_{3,3}$, should fall in the same region F_1 . Thus the failure probability in failure region F_1 is $\frac{1}{2}$. Suppose we plan to compute the failure probability of a smaller failure region $F_2 \subset F_1$. Since $\bar{\theta}_{2,2} \in F_1$, only $\bar{\theta}_{3,2}$ and $\bar{\theta}_{3,3}$ have to be checked whether falling in F_2 or not. If $\bar{\theta}_{3,3} = \theta_4 \in F_2$, the failure probability of F_2 is $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$. This idea may speed up the reliability analysis when many samples are involved. We can use father clusters of a specific level to detect a larger failure region, e.g. F_{i-1} , and then pick the child failure clusters to detect a smaller failure region, e.g. F_i . At each level, only the child failure clusters get involved in the subsequent computation. In fact, not every level is involved in the evaluation of the failure probability. Which levels should be employed depends on how many levels of failure regions are involved and how large each region is. Fig.2(a) shows a tree with three levels involved in the computation of the failure probability. These three levels are respectively the m_1 th, m_2 th and m_3 th levels of the tree, and have respectively m_1 , m_2 and m_3 clusters. Fig.2(b) illustrates the configuration of the m_1 th and m_2 th clusters, with failure region F_1 denoted by a rectangle. Fig.2(c) illustrates the configuration of the m_2 th and m_3 th clusters, with failure region F_2 denoted by a rectangle.

3.2. Clustering Criteria

Clustering is a process of grouping similar objects together. There are two kinds of clustering methods: one features a ‘flat’ clustering, such as K -means clustering (MacQueen, 1967); the other is the hierarchical clustering, based on an agglomerative or di-

Figure 1: Tree or hierarchical cluster

visive, clustering procedure (Ward Jr, 1963; Macnaughton-Smith et al., 1964; Murphy, 2012).

To cluster, the total number of samples must be determined at the beginning. Given a target failure probability $P(F)$, the coefficient of variation (c.o.v.) of the estimate $\hat{P}(F)$ in a direct MC is

$$\delta(F) = \sqrt{\frac{1 - P(F)}{P(F)N}} \quad (7)$$

Thus, to achieve the accuracy with $\text{c.o.v.} = \delta(F)$, the number of samples N should satisfy

$$N = \frac{1 - P(F)}{\delta^2(F)P(F)} \quad (8)$$

The number of raw samples used in HFC must be equal to or larger than N to achieve the same accuracy.

Given a probability distribution $p(\boldsymbol{\theta})$, N samples are drawn independently and marked as $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_N$. Each sample constitutes a singleton cluster at the lowest level, i.e. $\{\boldsymbol{\theta}_1\}, \{\boldsymbol{\theta}_2\}, \dots, \{\boldsymbol{\theta}_N\}$. Two closest clusters are merged each time until a single cluster C_1 at the highest level is reached. Hence, there should be N levels of clusters of this tree. This hierarchical clustering procedure features a bottom-up or agglomerative clustering. If instead the divisive clustering is adopted, the procedure starts with a single cluster, and then recursively divides each cluster into two clusters, in a top-down fashion. The merging procedure is based on the distance between a pair of clusters, which is called the

(a) A tree with three level clusters involved in the evaluation.

(b) The m_1 th level and m_2 th level clusters.

(c) The m_2 th level and m_3 th level clusters.

Figure 2: An illustration of the computation of failure probabilities via hierarchical clustering. The bolder rectangle indicates the target failure region and the thinner one indicates the intermediate level failure region.

similarity function. The similarity function can be defined in various ways, often in terms of a linkage criterion. For example, let $\{\boldsymbol{\theta}_{k'} \in C_{m,k}\}$ be the member samples of cluster $C_{m,k}$. The linkage criterion between $C_{m,i}$ and $C_{m,j}$, giving the pairwise distance Δ_{ij} , can be defined as (Ward Jr, 1963; Murphy, 2012):

(1) Single-link:

$$\Delta_{ij} = \min_{\forall i', j'} \|\boldsymbol{\theta}_{i'} - \boldsymbol{\theta}_{j'}\| \quad (9)$$

(2) Complete-link:

$$\Delta_{ij} = \max_{\forall i', j'} \|\boldsymbol{\theta}_{i'} - \boldsymbol{\theta}_{j'}\| \quad (10)$$

(3) Ward's link:

$$\begin{aligned} \Delta_{ij} &= \sum_{\boldsymbol{\theta}_{k'} \in C_{m,i} \cup C_{m,j}} \|\boldsymbol{\theta}_{k'} - \bar{\boldsymbol{\theta}}_{C_{m,i} \cup C_{m,j}}\|^2 - \sum_{\boldsymbol{\theta}_{i'} \in C_{m,i}} \|\boldsymbol{\theta}_{i'} - \bar{\boldsymbol{\theta}}_{C_{m,i}}\|^2 - \sum_{\boldsymbol{\theta}_{j'} \in C_{m,j}} \|\boldsymbol{\theta}_{j'} - \bar{\boldsymbol{\theta}}_{C_{m,j}}\|^2 \\ &= \frac{n_{C_{m,i}} n_{C_{m,j}}}{n_{C_{m,i}} + n_{C_{m,j}}} \|\bar{\boldsymbol{\theta}}_{C_{m,i}} - \bar{\boldsymbol{\theta}}_{C_{m,j}}\|^2 \end{aligned} \quad (11)$$

where $\|\cdot\|$ represents a general norm in the two former linkages, and the 2-norm in the Ward's linkage; $\bar{\boldsymbol{\theta}}_{C_{m,i} \cup C_{m,j}}$ represents the mean of the member samples in the union $C_{m,i} \cup C_{m,j}$; $n_{(\cdot)}$ represents the number of member samples of cluster $(\cdot)$. The Ward's link is recommended here for the purpose of this study because the definition of the Ward's link corresponds to the Centroidal Voronoi Tessellation.

3.3. Estimation of Failure Probability

The failure probability can be estimated in the two ways, illustrated in the following: The first method consists in counting the portion of samples falling in the target region F_n ; the second consists in estimating the sequence of conditional probabilities $\{P(F_i|F_{i-1}), i = 2, 3, \dots, n\}$ and then use them to calculate the target failure probability.

In the following, $\{\boldsymbol{\theta}_k\}_{k=1}^N$ is also used as the equivalent notation of $\{\boldsymbol{\theta}_k, k = 1, \dots, N\}$ for brevity.

3.3.1. Method I: pruning the safe branches

The failure probability can be estimated through the following 5 steps:

- (1) Draw N samples $\{\boldsymbol{\theta}_k\}_{k=1}^N$ independently according to $p(\boldsymbol{\theta})$ and then build a tree $\{C_{i,\cdot}\}_{i=1}^N$ for these samples;
- (2) Choose n levels of nested failure regions $\{F_1 \supset F_2 \cdots \supset F_n = F\}$, and select n levels of clusters: $C_{m_1,\cdot}, C_{m_2,\cdot}, \dots, C_{m_n,\cdot}$, which contain the number $m_1, m_2, \dots, m_n$ of clusters respectively, where $m_1 < m_2 < \dots < m_n = N$. Each cluster $C_{m_i,k}$ has a representative sample $\bar{\boldsymbol{\theta}}_{m_i,k}$;
- (3) Use $\{\bar{\boldsymbol{\theta}}_{m_1,k}\}_{k=1}^{m_1}$ to explore the failure region F_1 ;
- (4) Suppose $\tilde{m}_1$ representatives of $\{\bar{\boldsymbol{\theta}}_{m_1,k}\}_{k=1}^{m_1}$ fall in F_1 . Mark them as $\{\tilde{\boldsymbol{\theta}}_{m_1,k}\}_{k=1}^{\tilde{m}_1}$. Turn to the m_2 th level, find the children of $\{\tilde{\boldsymbol{\theta}}_{m_1,k}\}_{k=1}^{\tilde{m}_1}$ and mark them as $\{\tilde{\boldsymbol{\theta}}_{m_2,k}\}_{k=1}^{\tilde{m}_2}$. Use $\{\tilde{\boldsymbol{\theta}}_{m_2,k}\}_{k=1}^{\tilde{m}_2}$ to explore the failure region F_2 , and then mark the failure representatives as $\{\tilde{\boldsymbol{\theta}}_{m_2,k}\}_{k=1}^{\tilde{m}_2}$;
- (5) Repeat step 4 until the failure samples $\{\tilde{\boldsymbol{\theta}}_{m_n,k}\}_{k=1}^{\tilde{m}_n}$ are sifted out. Note that $\{\tilde{\boldsymbol{\theta}}_{m_n,k}\}_{k=1}^{\tilde{m}_n} = I_{F_n}(\{\boldsymbol{\theta}_k\}_{k=1}^N)$. Then the failure probability $P(F_n)$ is estimated as:

$$\hat{P}(F_n) = \frac{1}{N} \sum_{k=1}^N I_{F_n}(\boldsymbol{\theta}_k) = \frac{\tilde{m}_n}{N} \quad (12)$$

3.3.2. Method II: estimate the sequence of conditional probabilities

The primary procedure is analogous to Method I, but in this case the failure probability is estimated by the product of conditional probabilities rather than the portion of the

240 failure samples, according to the following equation

$$\begin{aligned}
\hat{P}(F_n) &= \hat{P}(F_1) \prod_{i=2}^n \hat{P}(F_i|F_{i-1}) \\
&= \frac{1}{m_1} \sum_{j=1}^{m_1} I_{F_1}(\boldsymbol{\theta}_j) \times \prod_{i=2}^n \frac{1}{\tilde{m}_i} \sum_{k=1}^{\tilde{m}_i} I_{F_i}(\boldsymbol{\theta}_k) \\
&= \frac{\tilde{m}_1}{m_1} \times \prod_{i=2}^n \frac{\tilde{m}_i}{\tilde{m}_i}
\end{aligned} \tag{13}$$

241 where $I_{F_1}(\{\boldsymbol{\theta}_k\}_{k=1}^{m_1}) = \{\tilde{\boldsymbol{\theta}}_{m_1, \ell}\}_{\ell=1}^{\tilde{m}_1}$; $I_{F_n}(\{\boldsymbol{\theta}_k\}_{k=1}^{\tilde{m}_n}) = \{\tilde{\boldsymbol{\theta}}_{m_i, \ell}\}_{\ell=1}^{\tilde{m}_i}$.

242 3.3.3. Remarks

243 (1) It is interesting to observe that, if $P(F_1) = \frac{P(F_2)}{P(F_1)} = \dots = \frac{P(F_n)}{P(F_{n-1})}$, the number of
244 clusters to be used to explore the failure region should be similar between different
245 levels, i.e. $m_1 \approx \tilde{m}_2 \approx \dots \approx \tilde{m}_n$. Since $\frac{P(F_i)}{P(F_{i-1})} \gg P(F_n)$, then $\tilde{m}_i \ll N$ and even
246 $n \times \tilde{m}_i \ll N$, which is considerably advantageous from a computational viewpoint.
247 Some details of the implementation are discussed in the following section;

248 (2) In practice, the nested failure regions $\{F_1 \supset F_2 \cdots \supset F_{n-1}\}$ are determined as follows.
249 For example, let us consider the limit state function $g(\boldsymbol{\Theta}) = f(\boldsymbol{\Theta}) - b < 0$. First,
250 at a particular level, after calculating $f(\boldsymbol{\Theta})$ of each cluster, a prescribed percentile,
251 like the 10th or 20th, is taken as b , which is the threshold of this level. Second, the
252 clusters whose performance is higher than the threshold are dropped along with their
253 child clusters. In this case, $\hat{P}(F_i|F_{i-1}) = 0.1$ or 0.2 . It should be emphasized that the
254 child clusters of a failure father cluster may not totally fail, as shown in Fig.4. Here,
255 $C_{m_1,1}$ is a failure father cluster at level m_1 and it has two child clusters. But actually,
256 child cluster $C_{m_2,1}$ falls outside the failure region. When determining the threshold
257 of level m_2 , $C_{m_2,1}$ should not be counted.

4. Implementation Issues

4.1. Techniques for Improving Clustering Speed

4.1.1. Agglomerative clustering with K -means clustering

From the agglomerative clustering procedure, it is possible to observe that optimally merging N clusters to $N-1$ clusters requires the evaluation of $\frac{N(N-1)}{2}$ possible pairs. Similarly, the selection of an optimal pair to merge $N-1$ clusters to $N-2$ requires $\frac{(N-1)(N-2)}{2}$ evaluations. Each merging takes $O(N^2)$ time. Considering that there are N levels in the algorithm, the total running time is $O(N^3)$. However, it is not necessary to build an N -level tree with pairs merged sequentially. Actually, pairs can be merged in parallel at each level such that an n -level tree is sufficient for the evaluation of n -level nested failure regions. Levels of this n -level tree are indexed by $m_1, m_2, \dots, m_n$ for consistency. The m_i th level has m_i clusters. This tree can be build with the aid of K -means clustering. K -means clustering (MacQueen, 1967) is a method commonly used to automatically partition a data set into k clusters in which each data point belongs to the cluster with the nearest mean. Table 1 provides a K -means clustering algorithm. This ‘hierarchical K -means clustering’ proceeds as follows:

- (1) Draw N samples $\{\boldsymbol{\theta}_k\}_{k=1}^N$ independently according to $p(\boldsymbol{\theta})$ and initialize them as the m_n th level clusters;
- (2) Implement K -means clustering for $\{\boldsymbol{\theta}_k\}_{k=1}^N$ and establish the m_{n-1} th level clusters $\{\bar{\boldsymbol{\theta}}_{m_{n-1},k}\}_{k=1}^{m_{n-1}}$;
- (3) Repeat step 2 for the $m_{\ell-1}$ th level clustering on the m_ℓ clusters, for $\ell = n \rightarrow 2$;

K -means clustering takes $O(m_{\ell-1}N \times \text{Iter})$ time for each level, where ‘Iter’ denotes the number of iterations. Thus the overall complexity for n levels is $O(\sum_{\ell=2}^n m_{\ell-1}N \times \text{Iter})$, which is usually lower than $O(N^3)$.

Table 1: Algorithm for K -means clustering (MacQueen, 1967)

1	Initialize the m_ℓ -level clusters $C_{m_\ell,1}, C_{m_\ell,2}, \dots, C_{m_\ell,m_\ell}$ with $\bar{\theta}_{m_\ell,1}, \bar{\theta}_{m_\ell,2}, \dots, \bar{\theta}_{m_\ell,m_\ell}$;
2	Repeat
3	Assign each sample to its closet cluster center: $\theta_k \xrightarrow{\arg \min_{i \in \{1, \dots, m_\ell\}} \ \theta_k - \bar{\theta}_{m_\ell,i}\ } C_{m_\ell,i}$;
4	Update each cluster center by computing the mean of all samples assigned to it: $\bar{\theta}_{m_\ell,i} = \frac{1}{N} \sum_{k=1}^N I_i(\theta_k) \theta_k;$
5	until converged.

4.1.2. Bisecting K -means clustering and bisecting K -medoids clustering

Bisecting K -means clustering is a divisive clustering algorithm, but it can also be used as an alternative to the traditional K -means clustering. Bisecting K -means clustering proceeds as follows (Steinbach et al., 2000):

- (1) Simulate N samples $\{\theta_k\}_{k=1}^N$ and initialize the highest level cluster as the set of these samples;
- (2) Split the largest cluster into 2 clusters using K -means algorithm;
- (3) Repeat step 2 till the m_{n-1} th level;

Note that it is not necessary to split the clusters till the lowest level, because the m_{n-1} th level clusters have already indicated which clusters the samples belong to. For step 2, the largest cluster means the one with the least overall similarity. Bisecting K -means is only required to run once for the m_{n-1} th level, during which the clusters on the other levels are obtained as byproducts. Complexity of the bisecting K -means algorithm is $O(m_{n-1}N \times \text{Iter})$. Thus it performs faster than the regular agglomerative clustering. In high-dimensional cases, a representative point resulting from K -means clustering deviates from the actual representative of a cluster because the representative point is an artificial ‘mean’ of the cluster. K -medoids clustering, which chooses the medoid of the cluster, is more meaningful for the purpose of this study (Kaufman and Rousseeuw, 2009). The

algorithm of K -medoids clustering is similar to K -means except the step of choosing
 representative points. K -medoids picks in a cluster the point closest to its mean instead
 of the mean itself.

4.2. Statistics of The Failure Probability Estimator via HFC

4.2.1. Relation to the Voronoi tessellation

The distribution of clusters at each level is crucial to the estimation of the failure
 probability. Every specific distribution implies a partition of the random space. The
 optimal partitioning problem is widely studied in the area of quasi-Monte Carlo (QMC)
 methods and clustering algorithms.

Consider the quadrature rule of the type

$$\int_{\Omega_{\theta}} p(\theta) d\theta \approx \sum_{k=1}^{N_0} p(\theta_k) A_k \quad (14)$$

where $\{\theta_k\}_{k=1}^{N_0}$ are N_0 points distributed in Ω_{θ} and $\{A_k\}_{k=1}^{N_0}$ are the volumes of regions
 $\{V^{(k)}\}_{k=1}^{N_0}$ that tessellate Ω_{θ} . For the open set $\Omega_{\theta} \subseteq \mathbb{R}^d$, the set of regions $\{V^{(k)}\}_{k=1}^{N_0}$ is
 called a tessellation of Ω_{θ} if $V^{(j)} \cap V^{(k)} = \emptyset$ for $j \neq k$ and $\bigcup_{k=1}^{N_0} V^{(k)} = \Omega_{\theta}$. Assume
 $\mathcal{P} = \{\theta_k\}_{k=1}^{N_0}$ to be a candidate set of representative points distributed according to
 $p(\theta)$, and a subset $V^{(k)}$ to be associated to each point θ_k . It is expected that such
 a representative point set (or rep-point set for simplicity) $\mathcal{P}$ over Ω_{θ} serves as a good
 representation for $p(\theta)$. Quadrature error is used here to measure the difference between
 the real probability and the approximate one over a certain domain, as follows

$$\begin{aligned} \epsilon &= \left\| \int_{\Omega_{\theta}} p(\theta) d\theta - \sum_{k=1}^{N_0} p(\theta_k) A_k \right\| \\ &= \left\| \sum_{k=1}^{N_0} \int_{V^{(k)}} [p(\theta) - p(\theta_k)] d\theta \right\| \\ &\leq L \sum_{k=1}^{N_0} \int_{V^{(k)}} \|\theta - \theta_k\| d\theta \end{aligned} \quad (15)$$

where the last inequality exists on the premise that $p(\boldsymbol{\theta})$ is a Lipschitz continuous function with the Lipschitz constant L . It has been proved that the right-hand side of the inequality is minimized when $\{V^{(k)}\}_{k=1}^{N_0}$ are the Voronoi sets for $\{\boldsymbol{\theta}_k\}_{k=1}^{N_0}$, and $\{\boldsymbol{\theta}_k\}_{k=1}^{N_0}$ are the mass centroids of $\{V^{(k)}\}_{k=1}^{N_0}$ (Du et al., 1999). The same Voronoi sets can also be obtained through K -means clustering, which instead minimizes $\sum_{k=1}^{N_0} \int_{V^{(k)}} p(\boldsymbol{\theta}_k) \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\| d\boldsymbol{\theta}$. Actually both ways the solution is generated, which is a centroidal Voronoi tessellation (Du et al., 1999). The definition of the mass centroid is given by

$$\boldsymbol{\theta}_k = \frac{\int_{V^{(k)}} \boldsymbol{\theta} p(\boldsymbol{\theta}) d\boldsymbol{\theta}}{\int_{V^{(k)}} p(\boldsymbol{\theta}) d\boldsymbol{\theta}} \quad (16)$$

While the Voronoi sets are defined by

$$\hat{V}^{(k)} = \left\{ \boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}} \mid \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\| < \|\boldsymbol{\theta} - \boldsymbol{\theta}_j\| \mid \text{for } j = 1, \dots, N_0, j \neq k \right\} \quad (17)$$

Using Eq.14, the failure probability can be expressed as

$$P(F) = \int_{\Omega_{\boldsymbol{\theta}}} I_F(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = \int_F p(\boldsymbol{\theta}) d\boldsymbol{\theta} \approx \sum_{k=1}^{N_0} p(\tilde{\boldsymbol{\theta}}_k) \tilde{A}_k \quad (18)$$

where $\{\tilde{\boldsymbol{\theta}}_k\}_{k=1}^{N_0}$ and $\{\tilde{A}_k\}_{k=1}^{N_0}$ represent the points and tessellation of F respectively. The quadrature error is given by

$$\begin{aligned} \tilde{\epsilon} &= \left\| \int_F p(\boldsymbol{\theta}) d\boldsymbol{\theta} - \sum_{k=1}^{N_0} p(\tilde{\boldsymbol{\theta}}_k) \tilde{A}_k \right\| \\ &\leq \tilde{L} \sum_{k=1}^{N_0} \int_{\tilde{V}^{(k)}} \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}_k\| d\boldsymbol{\theta} \end{aligned} \quad (19)$$

Similarly, the constrained mass centroid is given by

$$\tilde{\boldsymbol{\theta}}_k = \frac{\int_{\tilde{V}^{(k)}} \boldsymbol{\theta} p(\boldsymbol{\theta}) d\boldsymbol{\theta}}{\int_{\tilde{V}^{(k)}} p(\boldsymbol{\theta}) d\boldsymbol{\theta}} \quad (20)$$

(a) The optimal Voronoi tessellation for a bivariate Gaussian distribution. '□' represents the bound of a failure region.

(b) The optimal constrained Voronoi tessellation of the failure region

Figure 3: A Voronoi tessellation with a failure region. A polygon denotes a Voronoi cell, while '•' denotes the centroid of the Voronoi cell;

330 F is an unknown region determined by the limit state function, such as $F = \{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}} :$
331 $g(\boldsymbol{\theta}) < 0\}$. Thus, to find the shape of F coincides with finding the limit state function.
332 The computational effort necessary to find the exact limit state function is equivalent
333 to the effort required to identify the failure PDF. However the failure probability can
334 be more easily assessed, since it can be computed with an approximate limit function.
335 Thus, an effective way to do this is to determine an approximate constrained Voronoi
336 tessellation rather than a perfect constrained Voronoi tessellation, which is a constrained
337 optimization problem (Du et al., 2003). The approximate constrained Voronoi tessellation
338 can be determined after removing the part outside F from the original Voronoi tessellation.
339 Errors will in general occur in the procedure as the boundary of the failure region will
340 not be exactly detected. An example of this is shown by Figs.3(a) and 3(b), where it
341 can be seen that the boundary of the approximate constrained Voronoi tessellation differs
342 from the exact one. This error can be reduced by increasing the number of partitions.
343 Fortunately, the failure probability is less sensitive to this error than the failure PDF.

Figure 4: Hierarchical failure branches

4.2.2. Propagation of the estimation error in HFC

Section 4.2.1 mentions that an error arises in the estimation of the failure clusters at a specific level. In this section, the propagation of this error among different levels is examined. In fact, even if the boundary of a failure region is not exactly detected, the failure probability can still be estimated with good accuracy. This proposition is referred to the law of large numbers. It should be noticed, however, that in the application to HFC, the Voronoi tessellation is applied to multiple levels, and this requires to pay particular attention to this problem. The boundary of a failure region defines the failure clusters as well as their territories. The indicator (safe or failure) of the child clusters depends on their fathers' territories. The consequences of this are examined for the two estimation methods mentioned above:

1. For Method I, there is no approximation error if and only if all the child failure clusters fall in the father failure clusters' territories, and these territories contain no child safe clusters. A situation may occur as illustrated in Fig.4: the child cluster $C_{m_2,2}$ should fail but escapes the detection because its father $C_{m_1,2}$ is labeled as safe; meanwhile, the child cluster $C_{m_2,1}$ is erroneously judged as failing because its

360 father $C_{m_{1,1}}$ failed. The escaped child cluster will therefore be removed from the
 361 subsequent computation, and thereby will lead to an underestimation of the final
 362 failure probability. As a consequence, the estimation of the failure probability by
 363 Method I, i.e. labeling the hierarchical failure branches, tends to underestimate the
 364 target failure probability. This bias may be reduced by improving the quality of the
 365 partition, but not totally eliminated. A positive feature is that the detected failure
 366 samples are certainly located in the target failure region.

- 367 2. Method II uses the father clusters to explore the failure region F_{i-1} and then sift out
 368 the corresponding child clusters falling in this region. Then, the sifted child clusters
 369 are used to explore the failure region F_i and to obtain the proportion of child clusters
 370 falling in F_i as the conditional failure probability $P(F_i|F_{i-1})$. The accuracy of the
 371 computed $P(F_i|F_{i-1})$ depends on two factors: the first is the detection of the child
 372 clusters in F_{i-1} ; the second is the detection of the child clusters in F_i . If a child
 373 cluster outside F_{i-1} is misjudged as failure, this would result in an underestimation
 374 of $P(F_i|F_{i-1})$ because the area of F_{i-1} is overestimated; conversely, if a child cluster
 375 inside F_{i-1} escapes the detection, $P(F_i|F_{i-1})$ is overestimated because the area of
 376 F_{i-1} is underestimated. For a good partition, the chance of escape and misjudgment
 377 should be close. Thus, on average, the bias of the estimation of $P(F_i|F_{i-1})$ can be
 378 considered null. Note that although Method II follows the same procedure of Method
 379 I, it solves a different problem. Method I indeed focuses on how to sift out the
 380 failure samples from a huge number of raw samples. On the other hand, Method II
 381 only aims to examine the proportion of failure samples, albeit the failure samples
 382 are not totally sifted out. The propagation error is therefore reduced because the
 383 escaped clusters affect the sifting procedure but may not influence the computation
 384 of the proportion. For a good partition, a fraction of raw samples has the same
 385 ratio failure/safe samples as the entire set of raw samples. Thus, to make sure that

$P(F_i|F_{i-1})$ is unbiased, more work should be done to find a nested sequence of good partitions, which involves the problems to choose a proper number of levels and to choose a proper number of clusters at each level. Section 4.2.3 provides a guidance to these options.

4.2.3. Selection of the number of levels and the number of clusters at each level

Assume there are n levels in the nested subsets, and each level has $m_i, i = 1, \dots, n$ clusters. Without loss of generality, let $P(F_1) = P(F_2|F_1) = \dots = P(F_n|F_{n-1}) = P_0$, from which

$$P(F) = P(F_1) \prod_{i=2}^n P(F_i|F_{i-1}) = P_0^n \quad (21)$$

or

$$P_0 = \sqrt[n]{P(F)} \quad (22)$$

The number of clusters to compute each conditional probability should satisfy the following relation

$$N_0 \geq \frac{1 - P_0}{\delta^2 P_0} \quad (23)$$

Thus the total simulations by HFC is

$$N_{\text{sim}} = nN_0 \geq n \frac{1 - \sqrt[n]{P(F)}}{\delta^2 \sqrt[n]{P(F)}} \quad (24)$$

The minimum value of $N_{\text{sim}} = n \frac{1 - \sqrt[n]{P(F)}}{\delta^2 \sqrt[n]{P(F)}}$, with varying n and $P(F)$, and for $\delta = 0.05$ is plotted in Fig.5(a). The relation implies a monotonic decrease of N_{sim} with increasing n and increasing $P(F)$. For fixed $P(F)$ and δ , the relation between N_{sim} and n is plotted in Fig.5(b). For example, in the case $P(F) = 10^{-4}, \delta = 0.05$, the total number of simulations required in HFC is only approximately $\frac{1}{300}$ th of the number required by the direct MC when $n = 4$. The number of simulations can be further reduced by adding more levels. However, adding more levels may increase the propagation error. The propagation error

405 results in introducing a bias in the probability estimate. Note that the efficiency is
 406 primarily given by the fact that levels are used, but the addition of more levels gives a
 407 minor improvement. Another point worth mentioning is that N_0 represents the number
 408 of clusters to be evaluated at a specific level i , but, indeed, there are m_i clusters at this
 409 level. m_i should therefore satisfy the following requirement

$$m_i \geq \frac{1 - P(F_i)}{\delta^2 P(F_i)} = \frac{1 - P_0^i}{\delta^2 P_0^i} = \frac{1 - P(F)^{\frac{i}{n}}}{\delta^2 P(F)^{\frac{i}{n}}}; \quad i = 1, 2, \dots, n \quad (25)$$

410 The above requirement provides the first criterion to determine which levels of the tree
 411 structure, i.e. m_i , should be employed in the failure analysis. Considering N_0 clusters
 412 to be evaluated, there should be $N_0 P_0$ clusters falling in the smaller failure region. To
 413 continue the subsequent analysis, N_0 should be refilled by the child clusters of these $N_0 P_0$
 414 clusters. However, there might not be exactly N_0 child clusters attributed to these $N_0 P_0$
 415 clusters. This requirement indicates a second criterion to determine m_i , that consists in
 416 ensuring that there are exactly or approximately N_0 clusters at m_i th level attributed to
 417 the $N_0 P_0$ failure clusters at m_{i-1} th level. This is not a strong requirement but we expect
 418 that the number of clusters to be evaluated at each level should be close to N_0 . Another
 419 observation concerns the fact that the above criteria rely on a known $P(F)$. In fact,
 420 $P(F)$ is unknown before the analysis is started. It is suggested here to choose a prior
 421 value of $P(F)$. The selection of a priori value requires that it should be smaller than
 422 the real $P(F)$. In dealing with a specific project with risks, there should be a suggested
 423 or acceptable failure probability. When the real $P(F)$ is larger than the prior value, the
 424 result is still feasible but more simulations are needed. When the real $P(F)$ is smaller, the
 425 result tends to be smaller and may not be accurate. There are two ways to obtain accurate
 426 results. Firstly, take a conservative value in the beginning, such as 1/10th or 1/100th of
 427 the acceptable failure probability. According to Eq.24, the total number of simulations
 428 increases linearly with $P(F)^{-\frac{1}{n}}$ rather than $P(F)^{-1}$, so that the actual simulations are only

Figure 5: Required number of simulations

1.78 or 3.16 times of the ideal situation when $n = 4$. Secondly, update the first estimation with a new run. The first estimation is able to determine where the magnitude of the real $P(F)$ lies in, and then a posterior value can be set appropriately.

4.2.4. Relation to machine learning

Most supervised learning methods in the machine learning area are devoted to the approximation of the limit state function, e.g. the neural network, support vector machine, naive Bayes, surrogate model and so on (Hurtado, 2004; Murphy, 2012). According to the no free lunch theorem (Wolpert and Macready, 1997), given the same number of samples, most well-designed supervised learning methods on the average result in a similar accuracy. It is known that supervised learning methods only utilize the information inferred by the labeled samples. HFC is able to circumvent the no free lunch restriction because it can exploit the information from unlabeled samples. HFC is indeed a combination of supervised learning methods and unsupervised learning methods. The pre-processing step, i.e. the construction of a tree, is the unsupervised learning link. The computational effort spent on the construction of a tree pays dividends in saved effort in the evaluation of performance functions, which dominates the computational effort involved in the analysis of large-scale systems.

5. Examples

5.1. Example 1: Rosenbrock Function

Consider a 2-dimensional input random vector Θ , which has a uniform distribution on $\Omega_{\theta} = \{-2 \leq \theta_1 \leq 2, -2 \leq \theta_2 \leq 2\}$. The input PDF is defined as $p(\theta) = \begin{cases} 1/16, \theta \in \Omega_{\theta} \\ 0, \text{otherwise} \end{cases}$. The limit state function is defined as:

$$g(\Theta) = f(\Theta) - b \quad (26)$$

where

$$f(\Theta) = 100(\Theta_2 - \Theta_1^2)^2 + (1 - \Theta_1)^2 \quad (27)$$

is the Rosenbrock function; b is the threshold. Note that the Rosenbrock function increases significantly from the center to the boundaries. For example, if $b = 1$, the failure region is $F = \{\theta \in \Omega_{\theta} \mid f(\theta) - b < 0 = f(\theta) - 1 < 0\}$. The exact solution to $P(F)$ is 0.0149, which means that the number of simulations for MC to achieve the accuracy with c.o.v. = 0.05 is

$$N = \frac{1 - 0.0149}{0.05^2 \times 0.0149} = 26446 \quad (28)$$

To compare the efficiency of the methods, we develop a tree for these 26446 samples before assessment. This tree has 4 levels, with a number of clusters at each level determined by Eq.25 as follows: $m_1 = 744$, $m_2 = 2876$, $m_3 = 8979$, $m_4 = 26446$. At each level, the number of clusters to be evaluated was approximately $N_0 = m_1 = 744$. Different cases were studied, with their results listed in Tables 2-4. All c.o.v.s. were calculated based on 100 runs. While HFC results in matching the analytical prediction, its performance can be enhanced if the tree is established by more than 26446 samples.

Table 2: Case 1 with threshold $b = 1$

Simulation method	Number of simulations	Mean of failure probability	c.o.v.
Direct MC	26446	0.0149	0.0500
HFC (Method I)	$\approx 4 \times 744 = 2976$	0.0149	0.0442
HFC (Method II)	$\approx 4 \times 744 = 2976$	0.0149	0.0483

Table 3: Case 2 with threshold $b = 0.05$

Simulation method	Number of simulations	Mean of failure probability	c.o.v.
Direct MC	1×10^5	9.690×10^{-4}	0.0986
HFC (Method I)	$\approx 4 \times 462 = 1848$	9.552×10^{-4}	0.1025
HFC (Method II)	$\approx 4 \times 462 = 1848$	9.638×10^{-4}	0.1013

464 5.2. Example 2: Eggholder Function

465 Consider a 2-dimensional input random vector Θ , which has a uniform distribution
 466 on $\Omega_{\Theta} = \{-512 \leq \theta_1 \leq 512, -512 \leq \theta_2 \leq 512\}$. The input PDF is defined as $p(\theta) =$
 467 $\begin{cases} 1/1024^2, \theta \in \Omega_{\Theta} \\ 0, \text{otherwise} \end{cases}$. The limit state function is defined as:

$$g(\Theta) = f(\Theta) - b \quad (29)$$

468 where

$$f(\Theta) = -(\Theta_2 + 47) \sin \left(\sqrt{\left| \Theta_2 + \frac{\Theta_1}{2} + 47 \right|} \right) - \Theta_1 \sin \left(\sqrt{|\Theta_1 - (\Theta_2 + 47)|} \right) \quad (30)$$

Table 4: Case 3 with threshold $b = 0.005$

Simulation method	Number of simulations	$\hat{P}(F)$	c.o.v.
Direct MC	1×10^6	9.813×10^{-5}	0.1016
HFC (Method I)	$\approx 4 \times 899 = 3596$	9.675×10^{-5}	0.0982
HFC (Method II)	$\approx 4 \times 899 = 3596$	9.887×10^{-5}	0.0989

469 is the Eggholder function. The failure region is defined by $F = \{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}} \mid g(\boldsymbol{\theta}) =$
 470 $f(\boldsymbol{\theta}) - b < 0\}$. Note that the Eggholder function has multiple local minima. Using SS,
 471 there should be sufficient samples at each level to identify the failure regions as well as
 472 their shapes, because the efficacy of SS relies on the quality of the estimated intermediate
 473 PDFs. Insufficient samples result in sparsely distributed failure samples, and thereby end
 474 up with poorly estimated intermediate PDFs. Even in the ideal situation that the sifted
 475 failure samples successfully pinpoint all the failure regions, more failure samples are still
 476 required to estimate the shapes of these regions. Some relevant discussion remarks have
 477 been made in Section 4.2.1. Instead, HFC reduces the complexity of this problem as it
 478 relies on how to pinpoint the failure regions with disregard of their shapes. Once the
 479 failure regions are detected by the father failure clusters, the child failure clusters would
 480 portray the shape of the failure regions. Furthermore, since the clusters at higher levels are
 481 distributed more uniformly than the random samples, the failure regions can be captured
 482 more accurately. Figs.6(a) and 6(b) show the failure samples via MC where $b = -850$ from
 483 50000 and 4800 samples respectively, while Fig.6(c) shows the failure samples via 4-level
 484 HFC under the same condition from 50000 raw samples. In fact, only 4800 simulations
 485 are used in HFC. Though some failure samples escape the detection, HFC still captures
 486 the failure regions accurately and computes the failure probability with good accuracy.
 487 Obviously, using the same number of simulations, MC captures fewer failure regions than
 488 HFC. It is therefore likely that SS, which uses MC or MCMC at each level, is also prone
 489 to a poor estimate of the failure probability. Tests on the performance of SS (not reported
 490 here for the sake of brevity) show no obvious improvement over MC for this example. The
 491 tests use both global and local proposal density functions. The results of HFC are listed
 492 in Tables 5-7, where all the c.o.vs. are calculated based on 100 runs.

Figure 6: Failure samples obtained by MC and HFC where $b=-850$

Table 5: Case 1 with threshold $b = -850$

Simulation method	Number of simulations	$\hat{P}(F)$	c.o.v.
Direct MC	8000	0.0023	0.2185
HFC (Method I)	≈ 850	0.0022	0.1983
HFC (Method II)	≈ 850	0.0022	0.2047

Table 6: Case 2 with threshold $b = -850$

Simulation method	Number of simulations	$\hat{P}(F)$	c.o.v.
Direct MC	50000	0.0024	0.0997
HFC (Method I)	≈ 4800	0.0023	0.1015
HFC (Method II)	≈ 4800	0.0023	0.1019

Table 7: Case 3 with threshold $b = -900$

Simulation method	Number of simulations	$\hat{P}(F)$	c.o.v.
Direct MC	50000	5.095×10^{-4}	0.2000
HFC (Method I)	≈ 5250	4.903×10^{-4}	0.1897
HFC (Method II)	≈ 5250	4.895×10^{-4}	0.1893

Figure 7: Beam on nonlinear springs

5.3. Example 3: Beam on Nonlinear Springs (Taflanidis, 2011)

Consider a simply supported beam shown in Fig.7, with length $L = 8m$, load P applied at distance a from the right end, modulus of elasticity E and moment of inertia I . The foundation support at both ends is modeled as a linear spring with coefficients K_A and K_B at ends A and B , respectively. The deflection at the middle of the beam according to the model is given by

$$z = d_m = \frac{PL^3}{12EI} \left(\frac{1}{2} - |k| \right) \left(\frac{1}{2} + |k| - k^2 \right) + \frac{P}{K_A} \left(1 - \frac{a}{L} \right) + \frac{Pa}{K_B L} \quad (31)$$

where

$$k = \frac{1}{2} - \frac{a}{L} \quad (32)$$

The system output is modeled by an additive prediction error e

$$y = z + e \quad (33)$$

where e is assumed to be a Gaussian variable with zero mean and variance σ_e^2 . Thus, this system has an 8-dimensional input random vector $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = [a \ P \ E \ I \ K_A \ K_B \ \sigma_e \ e] \quad (34)$$

503

The components of this vector have the following distributions

$$\begin{aligned}
\begin{bmatrix} a \\ P \end{bmatrix} &\sim \mathcal{N} \left(\begin{bmatrix} 4m \\ 120kN \end{bmatrix}, \begin{bmatrix} 1m^2 & -14m \cdot kN \\ -14m \cdot kN & 1225kN^2 \end{bmatrix} \right) \\
\ln(E) &\sim \mathcal{N}(\ln(200 \cdot 10^6 kN/m^2), 0.2^2) \\
\ln(I) &\sim \mathcal{N}(\ln(5 \cdot 10^{-3} m^4), 0.1^2) \\
\ln(\sigma_e) &\sim \mathcal{N}(\ln(1.5 \cdot 10^{-4} m), 1^2) \\
\begin{bmatrix} K_A \\ K_B \end{bmatrix} &\sim \mathcal{N} \left(\begin{bmatrix} 120000 \\ 120000 \end{bmatrix} kN/m, (120000 \cdot 0.2)^2 \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} (kN/m)^2 \right) \\
e &\sim \mathcal{N}(0, \sigma_e^2)
\end{aligned} \tag{35}$$

504 The failure region is defined by $F = \{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}} \mid y(\boldsymbol{\theta}) - b > 0\}$. The failure probabilities are
505 0.1, 0.01, 0.001 and 0.0001 given $b = 0.33$, $b = 0.45$, $b = 0.63$ and $b = 0.90$ respectively.

506 A comparison among MC, SS and HFC is conducted. This aims to compare the
507 number of simulations used in different methods given a prescribed b and c.o.v. In this
508 example, Gaussian density functions have been chosen as the global proposal density
509 function as a global proposal Gaussian density function unlike local proposal density
510 functions, involves no ad hoc hypothesis in the estimation. The global proposal density
511 functions are estimated respectively from the failure points at each level. For HFC,
512 K -medoids clustering was used instead of K -means because K -medoids result in better
513 representative points in high-dimensional cases. The results are all based on 100 runs and
514 shown in Fig.8. It is noted that the c.o.v. of MC grows exponentially with the number
515 of simulations, while the c.o.v.s. for SS and HFC both grow approximately in a linear
516 manner. However, given the same c.o.v., the number of simulations demanded in SS is
517 larger than HFC. For example, given $b = 0.90$, HFC employs 3×10^5 samples to construct
518 the tree, and uses only 642 of them to compute performance functions, while SS uses

Figure 8: Comparison of different simulation schemes

3308 samples and yields similar mean and c.o.v.. The performance of HFC can be further enhanced if more than 3×10^5 samples are employed to construct the tree.

6. Potential Future Extensions

The utility of HFC is not limited to accelerating the computation of the failure probability only, but also offers the advantage of being data-driven. With the extensive use of a cluster of sensors, the measured data are becoming increasingly large, whereas the distributions and complexities of these data are beyond ad hoc assumptions. Hence it is difficult to measure the estimation error between the data and distribution models. Unlike most of the methods discussed in the introduction, such as FORM/SORM, QMC, IS and SS, HFC requires no prior assumption on the distribution of the data. Instead of generating samples from an estimated distribution of the data, HFC is able to utilize data directly to build a hierarchical tree.

For high dimensional data, the concept of Euclidean distance becomes less precise as dimension increases, since the distance between any two points in a given dataset converges. The problem related to high dimensions is always challenging, but not exclusively for the clustering based methods. The estimation of high dimensional probability density function needed prior to a FORM/SORM, QMC, IS and SS based analysis, is equally

536 challenged. In order to address high dimensional problem, a reduction in dimension is
537 suggested to be carried out before clustering, e.g., PCA (principle component analysis),
538 SVD (singular value decomposition), ICA (independent component analysis) and NMF
539 (non-negative matrix factorization).

540 **7. Concluding Remarks**

541 In this paper, a new ‘hierarchical failure clustering’ (HFC) method for the computa-
542 tion of failure probabilities is developed. HFC can reduce the number of simulations by
543 several folds in comparison with the direct Monte Carlo simulation. In addition, HFC
544 distinguishes itself from conventional reliability analysis approaches because it does not
545 belong to the family of supervised learning methods. As a combination of unsupervised
546 and supervised learning approaches, HFC can utilize the unlabeled information in the
547 input to enhance the quality of the estimation of failure probability. The HFC presented
548 here is an initial effort and potentially holds the promise of ushering unsupervised learn-
549 ing approaches to reliability analysis. It is envisaged that the combination of HFC with
550 other effective simulation strategies would be very promising for future applications in
551 reliability estimates.

552 **Acknowledgment**

553 The support for this project was provided in part by the NSF Grants #CMMI 09-
554 28282 and CMMI 13-01008 which are gratefully acknowledged. The authors also wish
555 to thank Dr. Seymour M.J. Spence for his advice concerning subset simulation, and Dr.
556 Enrica Bernardini for her careful reading of the manuscript.

557 **References**

558 Ang, G. L., Ang, A. H.-S., Tang, W. H., 1992. Optimal importance-sampling density
559 estimator. Journal of engineering mechanics 118 (6), 1146–1163.

- 560 Au, S.-K., Beck, J. L., Oct. 2001. Estimation of small failure probabilities in high dimen-
 561 sions by subset simulation. Probabilistic Engineering Mechanics 16 (4), 263–277.
 562 URL <http://linkinghub.elsevier.com/retrieve/pii/S0266892001000194>
- 563 Bucher, C. G., 1988. Adaptive samplingan iterative fast Monte Carlo procedure. Struc-
 564 tural Safety 5 (2), 119–126.
- 565 Bucher, C. G., Bourgund, U., 1990. A fast and efficient response surface approach for
 566 structural reliability problems. Structural safety 7 (1), 57–66.
- 567 Ching, J., Au, S.-K., Beck, J. L., 2005. Reliability estimation for dynamical systems sub-
 568 ject to stochastic excitation using subset simulation with splitting. Computer methods
 569 in applied mechanics and engineering 194 (12), 1557–1579.
- 570 Der Kiureghian, A., Dakessian, T., 1998. Multiple design points in first and second-order
 571 reliability. Structural Safety 20 (1), 37–49.
- 572 Du, Q., Faber, V., Gunzburger, M., 1999. Centroidal Voronoi Tessellations : Applications
 573 and Algorithms 41 (4), 637–676.
- 574 Du, Q., Gunzburger, M. D., Ju, L., Jan. 2003. Constrained Centroidal Voronoi Tessella-
 575 tions for Surfaces. SIAM Journal on Scientific Computing 24 (5), 1488–1506.
 576 URL <http://epubs.siam.org/doi/abs/10.1137/S1064827501391576>
- 577 Fang, K. T., Wang, Y., 1994. Number-theoretic methods in statistics. Vol. 51. Chapman
 578 & Hall/CRC.
- 579 Faravelli, L., 1989. Response-surface approach for reliability analysis. Journal of Engi-
 580 neering Mechanics 115 (12), 2763–2781.
- 581 Gerstner, T., Griebel, M., 1998. Numerical integration using sparse grids. Numerical
 582 algorithms 18 (3-4), 209–232.

- 583 Hammersley, J. M., Handscomb, D. C., 1964. Monte carlo methods. Vol. 1. Springer.
- 584 Hammersley, J. M., Morton, K. W., 1956. A new Monte Carlo technique: antithetic
585 variates. In: Proceedings of the Cambridge Philosophical Society. Vol. 52. Cambridge
586 Univ Press, pp. 449–475.
- 587 Hurtado, J. E., 2004. Structural reliability: statistical learning perspectives. Vol. 17.
588 Springer.
- 589 Kareem, A., 1990. Reliability analysis of wind-sensitive structures. Journal of wind engi-
590 neering and industrial aerodynamics 33 (3), 495–514.
- 591 Kaufman, L., Rousseeuw, P. J., 2009. Finding groups in data: an introduction to cluster
592 analysis. Vol. 344. John Wiley & Sons.
- 593 Kurtz, N., Song, J., 2013. Cross-entropy-based adaptive importance sampling using Gaus-
594 sian mixture. Structural Safety 42, 35–44.
- 595 Li, J., Chen, J., 2008. The principle of preservation of probability and the generalized
596 density evolution equation. Structural Safety 30 (1), 65–77.
- 597 Macnaughton-Smith, P., Williams, W. T., Dale, M. B., Mockett, L. G., 1964. Dissimilarity
598 analysis: a new technique of hierarchical sub-division, 1034–1035.
- 599 MacQueen, J., 1967. Some methods for classification and analysis of multivariate observa-
600 tions. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and
601 probability. Vol. 1. California, USA, p. 14.
- 602 Murphy, K. P., 2012. Machine learning: a probabilistic perspective. The MIT Press.
- 603 Nie, J., Ellingwood, B. R., 2004. A new directional simulation method for system reliabil-
604 ity. Part I: application of deterministic point sets. Probabilistic engineering mechanics
605 19 (4), 425–436.

606 Rajashekhar, M. R., Ellingwood, B. R., 1993. A new look at the response surface approach
607 for reliability analysis. *Structural safety* 12 (3), 205–220.

608 Robert, C. P., Casella, G., 2004. *Monte Carlo statistical methods*. Vol. 319. Citeseer.

609 Rubinstein, R. Y., Kroese, D. P., 2004. *The cross-entropy method: a unified approach to*
610 *combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer.

611 Rubinstein, R. Y., Kroese, D. P., 2011. *Simulation and the Monte Carlo method*. Vol.
612 707. Wiley. com.

613 Shinozuka, M., 1983. Basic analysis of structural safety. *Journal of Structural Engineering*
614 109 (3), 721–740.

615 Smolyak, S. A., 1963. Quadrature and interpolation formulas for tensor products of certain
616 classes of functions. In: *Dokl. Akad. Nauk SSSR*. Vol. 4. p. 111.

617 Steinbach, M., Karypis, G., Kumar, V., Others, 2000. A comparison of document cluster-
618 ing techniques. In: *KDD workshop on text mining*. Vol. 400. Boston, pp. 525–526.

619 Taflanidis, A., 2011. *Class notes of Stochastic Systems*.

620 Thoft-Christensen, P., Baker, M. J., 1982. *Structural reliability theory and its applica-*
621 *tions*.

622 Ward Jr, J. H., 1963. Hierarchical grouping to optimize an objective function. *Journal of*
623 *the American statistical association* 58 (301), 236–244.

624 Wolpert, D. H., Macready, W. G., 1997. No free lunch theorems for optimization. *Evolu-*
625 *tionary Computation, IEEE Transactions on* 1 (1), 67–82.