On the Optimal Scaling of the Modified Metropolis-Hastings algorithm

K. M. Zuev & J. L. Beck
Division of Engineering and Applied Science
California Institute of Technology, MC 104-44, Pasadena, CA 91125, USA

L. S. Katafygiotis
Department of Civil and Environmental Engineering
Hong Kong University of Science and Technology, Hong Kong, China

ABSTRACT: Estimation of small failure probabilities is one of the most important and challenging problems in reliability engineering. In cases of practical interest, the failure probability is given by a high-dimensional integral. Since multivariate integration suffers from the curse of dimensionality, the usual numerical methods are inapplicable. Over the past decade, the civil engineering research community has increasingly realized the potential of advanced simulation methods for treating reliability problems. The Subset Simulation method, introduced by Au & Beck (2001a), is considered to be one of the most robust advanced simulation techniques for solving high-dimensional nonlinear problems. The Modified Metropolis-Hastings (MMH) algorithm, a variation of the original Metropolis-Hastings algorithm (Metropolis et al. 1953, Hastings 1970), is used in Subset Simulation for sampling from conditional high-dimensional distributions. The efficiency and accuracy of Subset Simulation directly depends on the ergodic properties of the Markov chain generated by MMH, in other words, on how fast the chain explores the parameter space. The latter is determined by the choice of one-dimensional proposal distributions, making this choice very important. It was noticed in Au & Beck (2001a) that the performance of MMH is not sensitive to the type of the proposal PDFs, however, it strongly depends on the variance of proposal PDFs. Nevertheless, in almost all real-life applications, the scaling of proposal PDFs is still largely an art. The issue of optimal scaling was realized in the original paper by Metropolis (Metropolis et al. 1953). Gelman, Roberts, and Gilks (Gelman et al. 1996) have been the first authors to publish theoretical results about the optimal scaling of the original Metropolis-Hastings algorithm. They proved that for optimal sampling from a high-dimensional Gaussian distribution, the Metropolis-Hastings algorithm should be tuned to accept approximately 25% of the proposed moves only. This came as an unexpected and counter-intuitive result. Since then a lot of papers has been published on the optimal scaling of the original Metropolis-Hastings algorithm. In this paper, in the spirit of Gelman et al. (1996), we address the following question which is of high practical importance: what are the optimal one-dimensional Gaussian proposal PDFs for simulating a high-dimensional conditional Gaussian distribution using the MMH algorithm? We present a collection of observations on the optimal scaling of the Modified Metropolis-Hastings algorithm for different numerical examples, and develop an optimal scaling strategy for MMH when it is employed within Subset Simulation for estimating small failure probabilities.

1 INTRODUCTION

One of the most important and challenging problems in reliability engineering is to estimate the failure probability p_F, that is, the probability of unacceptable system performance. This is usually expressed as an integral over a high-dimensional uncertain parameter space:

$$p_F = \int I_F(\theta)\pi(\theta)d\theta = \mathbb{E}_\pi[I_F(\theta)], \tag{1}$$

where $\theta \in \mathbb{R}^d$ represents the uncertain parameters needed to specify completely the excitation and dynamic model of the system; $\pi(\theta)$ is the joint probability density function (PDF) for θ; $F \subset \mathbb{R}^d$ is the failure domain in the parameter space (i.e. the set of parameters that lead to performance of the system that is considered to be unacceptable); and $I_F(\theta)$ stands for the indicator function, i.e. $I_F(\theta) = 1$ if $\theta \in F$ and $I_F(\theta) = 0$ if $\theta \notin F$.

Over the past decade, the engineering research community has realized the importance of advanced stochastic simulation methods for solving reliability problems. As a result, many different efficient algorithms have been recently developed, e.g. Subset Simulation (Au & Beck 2001a), Importance Sampling using Elementary Events (Au & Beck 2001b), Line Sampling (Koutsourelakis et al. 2004), Auxiliary domain method (Katafygiotis et al. 2007), Spherical Subset Simulation (Katafygiotis & Cheung 2007), Horseracing Simulation (Zuev & Katafygiotis, in press), to name but a few.

This paper focuses on the analysis of the Modified Metropolis-Hastings algorithm (MMH), a Markov chain Monte Carlo technique used in Subset Simulation (SS), which is presented in Section 2. The efficiency and accuracy of SS directly depends on the ergodic properties of the Markov chains generated by MMH. In Section 3, we examine the optimal scaling of MMH to tune the parameters of the algorithm to make the resulting Markov chain converge to stationarity as fast as possible. We present a collection of observations on the optimal scaling of MMH for different numerical examples, and develop an optimal scaling strategy for MMH when it is employed within SS for estimating small failure probabilities.

The rest of the paper is organized as follows. In Section 2, the original Subset Simulation method is described; the optimal scaling of the Modified Metropolis-Hastings algorithm is discussed in Section 3; Concluding remarks are given in Section 4.

2 SUBSET SIMULATION

The best known stochastic simulation algorithm for estimating high-dimensional integrals is Monte Carlo Simulation (MCS). In this method the failure probability p_F is estimated by approximating the mean of $I_F(\theta)$ in (1) by its sample mean:

$$p_F \approx \hat{p}_F^{MC} = \frac{1}{N}\sum_{i=1}^{N} I_F(\theta^{(i)}), \tag{2}$$

where samples $\theta^{(1)},\dots,\theta^{(N)}$ are independent and identically distributed (i.i.d.) samples from $\pi(\cdot)$, denoted $\theta^{(i)} \overset{i.i.d.}{\sim} \pi(\cdot)$. This estimate is just the fraction of samples that produce system failure. Notice that each evaluation of I_F requires a deterministic system analysis to be performed to check whether the sample implies failure. The main advantage of MCS is that its efficiency does not depend on the dimension d of the parameter space. Indeed, straightforward calculation shows that the coefficient of variation (c.o.v) of the Monte Carlo estimate (2), serving as a measure of accuracy, is given by:

$$\delta(\hat{p}_F^{MC}) = \sqrt{\frac{1-p_F}{Np_F}}. \tag{3}$$

However, MCS has a serious drawback: it is inefficient in estimating small failure probabilities. If p_F is very small, $p_F \ll 1$, then, it follows from (3) that the number of samples N (or, equivalently, number of system analyses) needed to achieve an acceptable level of accuracy is very large, $N \propto 1/p_F \gg 1$. This deficiency of MCS has motivated research to develop more efficient stochastic simulation algorithms for estimating small failure probabilities in high-dimensions.

The basic idea of Subset Simulation (Au & Beck 2001a) is the following: represent a very small failure probability p_F as a product of larger probabilities so $p_F = \prod_{j=1}^{m} p_j$, where the factors p_j are estimated sequentially, $p_j \approx \hat{p}_j$, to obtain an estimate $\hat{p}_F^{SS}$ for p_F as $\hat{p}_F^{SS} = \prod_{j=1}^{m} \hat{p}_j$. To reach this goal, let us consider a decreasing sequence of nested subsets of the parameter space, starting from the entire space and shrinking to the failure domain F:

$$\mathbb{R}^d = F_0 \supset F_1 \supset \ldots \supset F_{m-1} \supset F_m = F. \tag{4}$$

Subsets $F_1,\dots,F_{m-1}$ are called intermediate failure domains. As a result, the failure probability $p_F = P(F)$ can be rewritten in terms of conditional probabilities as follows:

$$p_F = \prod_{j=1}^{m} P(F_j|F_{j-1}) = \prod_{j=1}^{m} p_j, \tag{5}$$

where $p_j = P(F_j|F_{j-1})$ is the conditional probability at the $(j-1)^{th}$ conditional level. Clearly, by choosing the intermediate failure domains appropriately, all conditional probabilities p_j can be made large. Furthermore, they can be estimated, in principle, by the fraction of independent conditional samples that cause failure at the intermediate level:

$$p_j \approx \hat{p}_j^{MC} = \frac{1}{N}\sum_{i=1}^{N} I_{F_j}(\theta_{j-1}^{(i)}), \tag{6}$$

where $\theta_{j-1}^{(i)} \overset{i.i.d.}{\sim} \pi(\cdot|F_{j-1})$. Hence, the original problem (estimation of the small failure probability p_F) is replaced by a sequence of m intermediate problems (estimation of the larger failure probabilities p_j, $j = 1,\dots,m$).

The first probability $p_1 = P(F_1|F_0) = P(F_1)$ is straightforward to estimate by MCS, since (6) requires sampling from $\pi(\cdot)$ that is assumed to be readily sampled. However, if $j \geq 2$, to estimate p_j using (6) one needs to generate independent samples from conditional distribution $\pi(\cdot|F_{j-1})$, which, in general, is not a trivial task. It is not efficient to use MCS for this purpose, especially at higher levels, but it can be done by a specifically tailored Markov chain Monte Carlo technique at the expense of generating dependent samples.

Markov chain Monte Carlo (MCMC) (Liu 2001, Neal 1993, Robert & Casella 2004) is a class of algorithms for sampling from multi-dimensional target probability distributions that cannot be directly sampled, at least not efficiently. These methods are based on constructing a Markov chain that has the distribution of interest as its stationary distribution. By simulating samples from the Markov chain, these samples will eventually be draws from the target probability distribution but they will not be independent samples. In Subset Simulation, the Modified Metropolis-Hastings algorithm (MMH) (Au & Beck 2001a), an MCMC technique based on the original Metropolis-Hastings algorithm (Metropolis et al. 1953, Hastings 1970), is used for sampling from the conditional distributions $\pi(\cdot|F_{j-1})$.

Suppose we want to generate a Markov chain with stationary distribution

$$\pi(\theta|\mathbb{F})=\frac{\pi(\theta)I_{\mathbb{F}}(\theta)}{P(\mathbb{F})}=\frac{\prod_{k=1}^{d}\pi_k(\theta_k)I_{\mathbb{F}}(\theta)}{P(\mathbb{F})}, \tag{7}$$

where $\mathbb{F}\subset\mathbb{R}^d$ is a subset of the parameter space. Without significant loss of generality, we assume here that $\pi(\theta)=\prod_{k=1}^{d}\pi_k(\theta_k)$, i.e. components of θ are independent (but are not so when conditioned on F). MMH differs from the original Metropolis-Hastings algorithm algorithm in the way the candidate state $\xi=(\xi_1,\ldots,\xi_d)$ is generated. Instead of using a d-variate proposal PDF on $\mathbb{R}^d$ to directly obtain the candidate state, in MMH a sequence of univariate proposal PDFs is used. Namely, each coordinate ξ_k of the candidate state is generated separately using a univariate proposal distribution dependent on the coordinate θ_k of the current state. Then a check is made if the d-variate candidate generated in such a way belongs to the subset $\mathbb{F}$ in which case it is accepted as the next Markov chain state; otherwise it is rejected and the current MCMC sample is repeated. To summarize, the Modified Metropolis-Hastings algorithm proceeds as follows:

Modified Metropolis-Hastings algorithm

```
Input:
  ▷ $\theta^{(1)}\in\mathbb{F}$, initial state of a Markov chain;
  ▷ $N$, total number of states, i.e. samples;
  ▷ $\pi_1(\cdot),\ldots,\pi_d(\cdot)$, marginal PDFs of $\theta_1,\ldots,\theta_d$;
  ▷ $S_1(\cdot|\alpha),\ldots,S_d(\cdot|\alpha)$, univariate proposal
    PDFs depending on a parameter $\alpha\in\mathbb{R}$ and
    satisfying the symmetry property
    $S_k(\beta|\alpha)=S_k(\alpha|\beta)$, $k=1,\ldots,d$.
Algorithm:
  for $i=1,\ldots,N-1$ do
    % Generate a candidate state $\xi$:
    for $k=1,\ldots,d$ do
      Sample $\tilde{\xi}_k\sim S_k(\cdot|\theta_k^{(i)})$
      Compute the acceptance ratio
        $r=\frac{\pi_k(\tilde{\xi}_k)}{\pi_k(\theta_k^{(i)})}$    (8)
      Accept or reject $\tilde{\xi}_k$ by setting
        $\xi_k=\begin{cases}\tilde{\xi}_k, & \text{with probability } \min\{1,r\};\\ \theta_k^{(i)} & \text{with probability } 1-\min\{1,r\}.\end{cases}$    (9)
    end for
    Check whether $\xi\in\mathbb{F}$ by system analysis
    and accept or reject $\xi$ by setting
      $\theta^{(i+1)}=\begin{cases}\xi, & \text{if } \xi\in\mathbb{F};\\ \theta^{(i)}, & \text{if } \xi\notin\mathbb{F}.\end{cases}$    (10)
  end for
Output:
  ▶ $\theta^{(1)},\ldots,\theta^{(N)}$, $N$ states of a Markov chain
    with stationary distribution $\pi(\cdot|\mathbb{F})$.
```

Thus, if we run the Markov chain for sufficiently long (the burn-in period), starting from essentially any "seed" $\theta^{(1)}\in\mathbb{F}$, then for large N the distribution of $\theta^{(N)}$ will be approximately $\pi(\cdot|\mathbb{F})$. Note, however, that in any practical application it is very difficult to check whether the Markov chain has reached its stationary distribution. If the seed $\theta^{(1)}\sim\pi(\cdot|\mathbb{F})$, all states $\theta^{(i)}$ will be automatically distributed according to the target distribution, $\theta^{(i)}\sim\pi(\cdot|\mathbb{F})$, since it is the stationary distribution for the Markov chain. This is called perfect sampling (Robert & Casella 2004) and Subset Simulation has this property because of the way the seeds are chosen.

Let us assume now that we are given a seed $\theta_{j-1}^{(1)}\sim\pi(\cdot|F_{j-1})$, where $j=2,\ldots,m$. Then, using MMH, we can generate a Markov chain with N states starting from this seed and construct an estimate for p_j similar to (6), where MCS samples are replaced by MCMC samples:

$$p_j\approx\hat{p}_j^{MCMC}=\frac{1}{N}\sum_{i=1}^{N}I_{F_j}(\theta_{j-1}^{(i)}), \tag{11}$$

where $\theta_{j-1}^{(i)}\sim\pi(\cdot|F_{j-1})$. Note that all samples $\theta_{j-1}^{(1)},\ldots,\theta_{j-1}^{(N)}$ in (11) are identically distributed, but are not independent. Nevertheless, these MCMC samples can be used for statistical averaging as if they were i.i.d., although with some reduction in efficiency (Doob 1953). Namely, the more correlated $\theta_{j-1}^{(1)},\ldots,\theta_{j-1}^{(N)}$ are, the less efficient is the estimate (11). The correlation between successive samples is due to proposal PDFs S_k, which govern the generation of the next state of the Markov chain from the current one in MMH. Hence, the choice of the proposal PDFs S_k controls the efficiency of estimate (11), making this choice very important. It was observed in Au & Beck (2001a) that the efficiency of MMH depends on the spread of proposal distributions, rather than on their type. Both small and large spreads tend to increase the dependence between successive samples, slowing the convergence of the estimator. Large spreads may reduce the acceptance rate in (10), increasing the number of repeated MCMC samples. Small spreads, on the contrary, may lead to a reasonably high acceptance rate, but still produce very correlated samples due to their close proximity. Finding the optimal spread of proposal distributions for MMH is a non-trivial task which is discussed in Section 3.

Finally, combining estimates (6) for p_1 and (11) for p_j, $j \geq 2$, we obtain the estimate for the failure probability:

$$p_F \approx \hat{p}_F^{SS} = \hat{p}_1^{MC} \prod_{j=2}^{m} \hat{p}_j^{MCMC} \tag{12}$$

The last ingredient of Subset Simulation we have to specify is the choice of intermediate failure domains $F_1, \dots, F_{m-1}$. Usually, performance of a dynamical system is described by a certain positive-valued performance function $g : \mathbb{R}^d \to \mathbb{R}_+$, for instance, $g(\theta)$ may represent some peak (maximum) response quantity when the system model is subjected to the uncertain excitation θ. Then the failure region, i.e. unacceptable performance region, can be defined as a set of excitations that lead to the exceedance of some prescribed critical threshold b:

$$F = \{\theta \in \mathbb{R}^d : g(\theta) > b\}. \tag{13}$$

The sequence of intermediate failure domains can then be defined as

$$F_j = \{\theta \in \mathbb{R}^d : g(\theta) > b_j\}, \tag{14}$$

where $0 < b_1 < \ldots < b_{m-1} < b_m = b$. Intermediate threshold values b_j define the values of the conditional probabilities $p_j = P(F_j|F_{j-1})$ and, therefore, affect the efficiency of Subset Simulation. In practical cases it is difficult to make a rational choice of the b_j-values in advance, so the b_j are chosen adaptively so that the estimated conditional probabilities are equal to a fixed value $p_0 \in (0, 1)$. For a more detailed description of the Subset Simulation method and its implementation, see the original paper Au & Beck (2001a).

3 TUNING OF THE MODIFIED METROPOLIS-HASTINGS ALGORITHM

The efficiency and accuracy of Subset Simulation directly depends on the ergodic properties of the Markov chain generated by the Modified Metropolis-Hastings algorithm; in other words, on how fast the chain explores the parameter space and converges to its stationary distribution. The latter is determined by the choice of one-dimensional proposal distributions S_k, which makes this choice very important. In spite of this, the choice of proposal PDFs is still largely an art. It was observed in Au & Beck (2001a) that the efficiency of MMH is not sensitive to the type of the proposal PDFs; however, it strongly depends on their variance.

Optimal scaling refers to the need to tune the parameters of the algorithm to make the resulting Markov chain converge to stationarity as fast as possible. The issue of optimal scaling was recognized in the original paper by Metropolis (Metropolis et al. 1953). Gelman, Roberts, and Gilks (Gelman et al. 1996) were the first authors to obtain theoretical results on the optimal scaling of the original Metropolis-Hastings algorithm. They proved that for optimal sampling from a high-dimensional Gaussian distribution, the Metropolis-Hastings algorithm should be tuned to accept approximately 25% of the proposed moves only. This result gives rise to the useful heuristic strategy, which is easy to implement: tune the proposal variance so that the average acceptance rate is roughly 25%. In spite of the i.i.d. assumption for the target components, this result is believed to be robust and to hold under various perturbations of the target distribution. Being aware of practical difficulties of choosing the optimal σ^2, Gelman et al. (1996) provided a very useful observation: “Interestingly, if one cannot be optimal, it seems better to use too high a value of σ than too low.” Since then many papers have been published on optimal scaling of the original Metropolis-Hastings algorithm. In this section, in the spirit of Gelman et al. (1996), we address the following question which is of high practical importance: what is the optimal variance σ^2 of the univariate Gaussian proposal PDFs $S_k(\cdot|\mu) = \mathcal{N}(\cdot|\mu, \sigma^2)$, $k = 1, \dots, d$, for simulating a high-dimensional conditional Gaussian distribution $\pi(\cdot|F) = \mathcal{N}(\cdot|0, \mathbb{I}_d)I_F(\cdot)/P(F)$ using the Modified Metropolis-Hastings algorithm and what is the optimal scaling strategy for MMH when it is employed within Subset Simulation for estimating small failure probabilities?

Let us first define what we mean by “optimal” variance. Let $\theta_{j-1}^{(i),k}$ be the the i^{th} sample in the k^{th} Markov chain at simulation level $j-1$. The conditional probability $p_j = P(F_j|F_{j-1})$ is then estimated as follows:

$$p_j \approx \hat{p}_j = \frac{1}{N} \sum_{k=1}^{N_c} \sum_{i=1}^{N_s} I_{F_j}(\theta_{j-1}^{(i),k}), \tag{15}$$

where $\theta_{j-1}^{(i),k} \sim \pi(\cdot|F_{j-1})$, N_c is the number of Markov chains, and N_s is the total number of samples simulated from each of these chains, $N_s = N/N_c$, so that the total number of Markov chain samples is N. An expression for the coefficient of variation (c.o.v.) of $\hat{p}_j$, derived in Au & Beck (2001a), is given by:

$$\delta_j = \sqrt{\frac{1-p_j}{Np_j}(1+\gamma_j)}, \tag{16}$$

where

$$\gamma_j = 2\sum_{i=1}^{N_s-1}\left(1-\frac{i}{N_s}\right)\frac{R_j^{(i)}}{R_j^{(0)}}, \qquad (1$$

and

$$R_j^{(i)} = \mathbb{E}[I_{F_j}(\theta_{j-1}^{(1),k})I_{F_j}(\theta_{j-1}^{(1+i),k})] - p_j^2 \qquad (1$$

is the autocovariance of the stationary stocha tic process $X(i) = I_{F_j}(\theta_{j-1}^{(i),k})$ at lag i. The fact $\sqrt{(1-p_j)/Np_j}$ in (16) is the c.o.v. of the MCS e timator with N independent samples. The c.o.v. $\hat{p}_j$ can thus be considered as the one in MCS wi an effective number of independent samples $N/(1$ $\gamma_j)$. The efficiency of the estimator using depende MCMC samples ($\gamma_j > 0$) is therefore reduced con pared to the case when the samples are independe ($\gamma_j = 0$). Hence, γ_j given by (17) can be considered as a measure of correlation between the states of a Markov chain and smaller values of γ_j imply higher efficiency.

Formula (16) was derived assuming that the Markov chain generated according to MMH is ergodic and that the samples generated by different chains are uncorrelated through the indicator function, i.e. $\mathbb{E}[I_{F_j}(\theta)I_{F_j}(\theta')] - p_j^2 = 0$ if θ and θ' are from different chains. The latter, however, may not be always true, since the seeds for each chain may be dependent. Nevertheless, the expression in (16) provides a useful theoretical description of the c.o.v. of $\hat{p}_j$.

The autocovariance sequence $R_j^{(i)}$, $i = 0, \dots, N_s - 1$, needed for calculation of γ_j, can be estimated using the Markov chain samples at the $(j-1)^{\text{th}}$ level by:

$$R_j^{(i)} \approx \frac{\sum_{k=1,i'=1}^{N_c,N_s-i} I_{F_j}(\theta_{j-1}^{(i'),k})I_{F_j}(\theta_{j-1}^{(i'+i),k})}{N - iN_c} - \hat{p}_j^2 \qquad (19)$$

Note that in general, γ_j depends on the number of samples N_s in the Markov chain, the conditional probability p_j, the intermediate failure domains F_{j-1} and F_j, and the standard deviation σ_j of the proposal PDFs $S_k(\cdot|\mu) = \mathcal{N}(\cdot|\mu, \sigma_j^2)$. According to the "basic" description of the Subset Simulation algorithm, $p_j = p_0$ for all j and $N_s = 1/p_0$. The latter is not strictly necessary, yet convenient. In this subsection, the value p_0 is chosen to be 0.1, as in the original paper (Au & Beck 2001a). In this settings, γ_j depends only on the standard deviation σ_j and geometry of F_{j-1} and F_j. For a given reliability problem (i.e. for a given performance function g that defines domains F_j for all j), σ_j^{opt} is said to be the *optimal spread* of the proposal PDFs at level j, if it minimizes the value of γ_j:

$$\sigma_j^{\text{opt}} = \arg\min_{\sigma_j>0}\gamma_j(\sigma_j) \qquad (20)$$

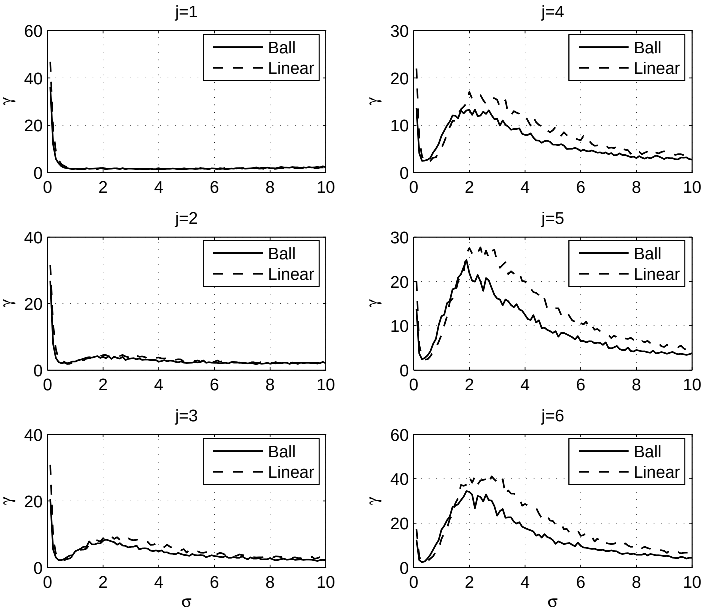

Figure 1: γ-efficiency of the Modified Metropolis-Hastings algorithm as a function of spread σ, for simulation levels $j = 1, \dots, 6$

We will refer to $\gamma_j = \gamma_j(\sigma_j)$ as γ-*efficiency* of the Modified Metropolis-Hastings algorithm with proposal PDFs $\mathcal{N}(\cdot|\mu, \sigma_j^2)$ at level j.

Consider two examples of the sequence of intermediate failure domains.

Example 1 (Exterior of a ball). Let $\theta = re \in \mathbb{R}^d$, where e is a unit vector and $r = \|\theta\|$. For many reasonable performance functions g, if r is large enough, then $\theta \in F = \{\theta \in \mathbb{R}^d : g(\theta) > b\}$, i.e. θ is a failure point, regardless of e. Therefore, an exterior of a ball, $\bar{B}_r = \{\theta \in \mathbb{R}^d : \|\theta\| \geq r\}$, can serve as an idealized model of many failure domains. Define the intermediate failure domains as follows:

$$F_j = \bar{B}_{r_j}, \qquad (21)$$

where the radii r_j are chosen such that $P(F_j|F_{j-1}) = p_0$, i.e $r_j^2 = F_{\chi_d^2}^{-1}(1-p_0^j)$, where $F_{\chi_d^2}$ denotes the cumulative distribution function (CDF) of the chi-square distribution with d degrees of freedom. The dimension d is chosen to be 10^3.

Example 2 (Linear case). Consider a linear reliability problem with performance function $g(\theta) = a^T\theta + b$, where $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are fixed coefficients. The corresponding intermediate failure domains F_j are half-spaces defined as follows:

$$F_j = \{\theta \in \mathbb{R}^d : \langle\theta, e_a\rangle \geq \beta_j\}, \qquad (22)$$

where $e_a = \frac{a}{\|a\|}$ is the unit normal to the hyperplane specified by g, and the values of β_j are chosen such that $P(F_j|F_{j-1}) = p_0$, i.e $\beta_j = \Phi^{-1}(1-p_0^j)$, where Φ denotes the CDF of the standard normal distribution. The dimension d is chosen to be 10^3.

For both examples, γ_j as a function of σ_j is plotted in Fig. 1 and the approximate values of the optimal

Table 1: Approximate values of the optimal spread for different simulation levels

Simulation Level j	1	2	3	4	5	6
Example 1, σ_j^{opt}	0.9	0.7	0.4	0.3	0.3	0.3
Example 2, σ_j^{opt}	1.1	0.8	0.6	0.4	0.4	0.4

spread σ_j^{opt} are given in Table 1, for simulation levels $j = 1, \ldots, 6$. As expected, the optimal spread σ_j^{opt} decreases, when j increases, and, as it follows from the numerical values in Table 1, σ_j^{opt} seems to converge to approximately 0.3 and 0.4 in Example 1 and 2, respectively. The following properties of the function $\gamma_j = \gamma_j(\sigma_j)$ are worth mentioning:

(i) γ_j increases very rapidly, when σ_j goes to zero;

(ii) γ_j has a deep trough around the optimal value σ_j^{opt}, when j is large (e.g., $j \geq 4$).

Interestingly, these observations are consistent with the statement given in Gelman et al. (1996) and cited above: if one cannot be optimal (due to (ii), it is indeed difficult to achieve optimality), it is better to use too high a value of σ_j than too low.

The question of interest now is what gain in efficiency can we achieve for a proper scaling of the Modified Metropolis-Hastings algorithm when calculating small failure probabilities? We consider the following values of failure probability: $p_F = 10^{-k}$, $k = 2, \ldots, 6$. The c.o.v. of the failure probability estimates obtained by Subset Simulation are given in Fig. 2 and Fig. 3 for Examples 1 and 2, respectively. The dashed (solid) curves correspond to the case when $N = 300$ ($N = 1000$) samples are used per each intermediate failure region. For estimation of each value of the failure probability, two different MMH algorithms are used within SS: the optimal algorithm with $\sigma_j = \sigma_j^{\text{opt}}$ (marked with stars); and the reference algorithm with $\sigma_j = 1$ (marked with squares). The corresponding c.o.v's are denoted by δ_{opt} and δ_1, respectively. From Fig. 2 and Fig. 3 it follows that the smaller p_F, the more important to scale MMH optimally. When $p_F = 10^{-6}$, the optimal c.o.v δ_{opt} is approximately 1.2 times smaller than the reference c.o.v. δ_1 for both examples, when $N = 1000$. Another important, yet expected, observation is that the significance of the optimal scaling of MMH diminishes when the number of samples per subset N grows. Indeed, when $N \rightarrow \infty$, the Markov chain converges to its stationarity regardless of the spread of the proposal PDF.

Despite its obvious usefulness, the optimal scaling of the Modified Metropolis-Hastings algorithm is difficult to achieve in practice. First, as it follows from Table 1, the values of the optimal spread σ_j^{opt} are different for different reliability problems. Second, even for a given reliability problem to find σ_j^{opt} is computationally expensive because of (ii); and our simulation results show that the qualitative properties (i) and (ii) generally hold for different reliability problems, not only for Examples 1 and 2.

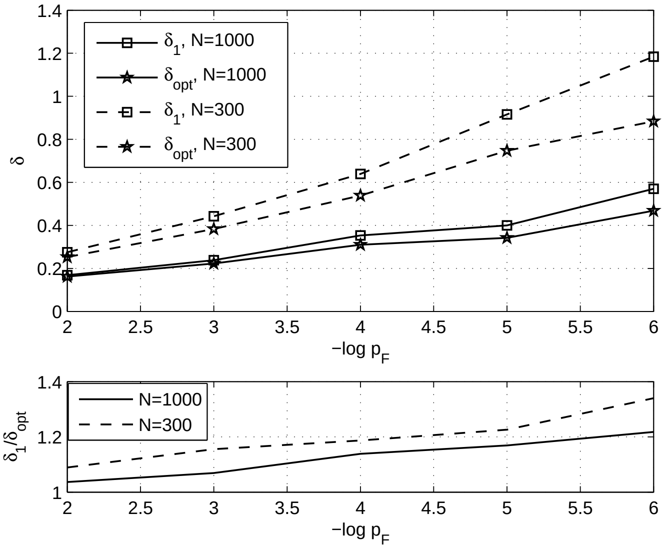

Figure 2: The c.o.v. of p_F estimates obtained by Subset Simulation for Example 1

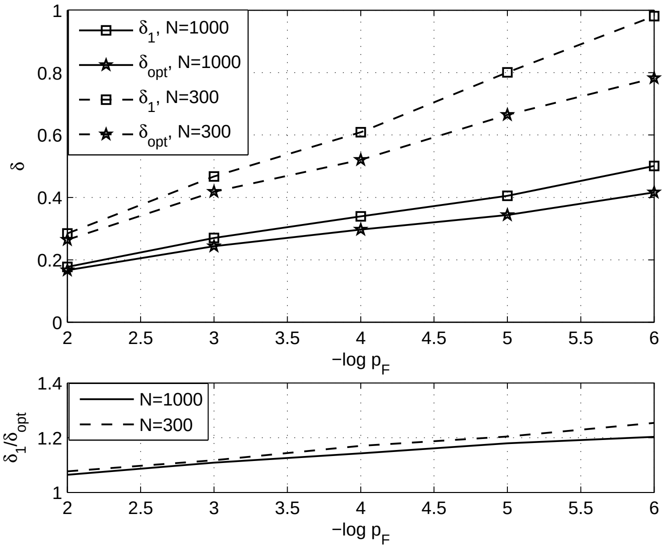

Figure 3: The c.o.v. of p_F estimates obtained by Subset Simulation for Example 2

It has been recognized for a long time that, when using an MCMC algorithm, it is useful to monitor its acceptance rate. Both γ-efficiency γ_j and the acceptance rate ρ_j at level j depend on σ_j. For Examples 1 and 2, γ_j as a function of ρ_j is plotted in Fig. 4, for simulation levels $j = 1, \ldots 6$. A key observation is that, contrary to (ii), γ_j is very flat around the optimal acceptance rate ρ_j^{opt}, which is defined as the acceptance rate that corresponds to the optimal spread, i.e. $\rho_j^{\text{opt}} = \rho_j(\sigma_j^{\text{opt}})$. Furthermore, according to our simulation results this behavior is typical, and not specific just for the considered examples. This observation gives rise to the following heuristic scaling strategy:

> *At simulation level j select σ_j such that the corresponding acceptance rate ρ_j is between 40% and 60% if $j = 1$ and between 30% and 50% if $j \geq 2$.*

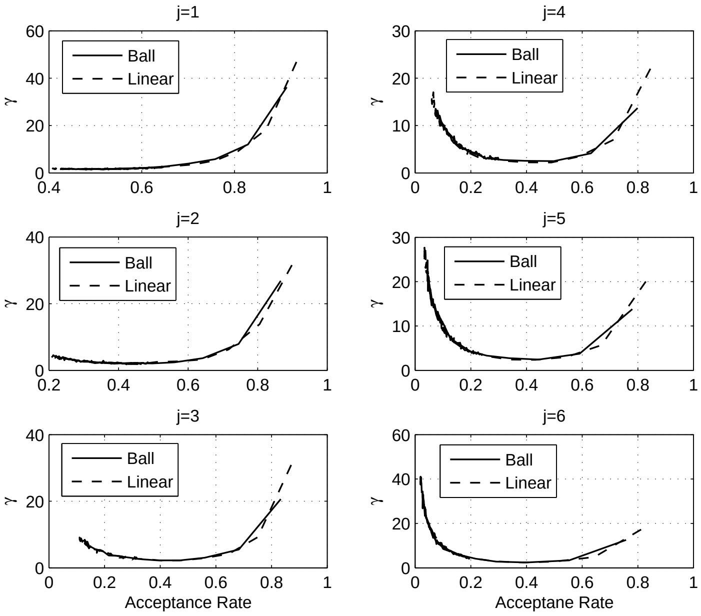

Figure 4: γ-efficiency of the Modified Metropolis-Hastings algorithm as a function of the acceptance rate, for simulation levels $j = 1,\ldots,6$

This strategy is easy to implement in the context of Subset Simulation. At each simulation level j, N_c Markov chains are generated. Suppose, we do not know the optimal spread σ_j^{opt} for our problem. We start with a reference value, say $\sigma_j^{1:n} = 1$, for the first n chains. Based only on these n chains, we calculate the corresponding acceptance rate $\rho_j^{1:n}$. If $\rho_j^{1:n}$ is too low (i.e. it is smaller than 40% and 30% for $j = 1$ and $j \geq 2$, respectively) we decrease the spread and use $\sigma_j^{n+1:2n} < \sigma_j^{1:n}$ for the next n chains. If $\rho_j^{1:n}$ is too large (i.e. it is larger than 60% and 50% for $j = 1$ and $j \geq 2$, respectively) we increase the spread and use $\sigma_j^{n+1:2n} > \sigma_j^{1:n}$ for the next n chains. We proceed like this until all N_c Markov chains have been generated. Note that according to this procedure, σ_j is kept constant within a single chain and it is changed only between chains. Hence the Markovian property is not destroyed. The described strategy guarantees that the corresponding scaling on the Modified Metropolis-Hastings algorithm is nearly optimal.

4 CONCLUSIONS

This paper explores the optimal scaling of the Modified Metropolis-Hastings algorithm, an MCMC technique employed within the Subset Simulation method proposed by Au & Beck (2001a). This exploration leads to the following nearly optimal scaling strategy for MMH: at the first simulation level select the spread (variance) of the proposal PDFs such that the corresponding acceptance rate is between 40% and 60%, while at higher levels so it is between 30% and 50%.

ACKNOWLEDGEMENTS

This work was supported by the Hong Kong Research Grants Council, under grant HKUST 614008 and by the National Science Foundation, under award number EAR-0941374. This support is gratefully acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

REFERENCES

Au, S. K. & J. L. Beck (2001a). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics 16(4)*, 263–277.

Au, S. K. & J. L. Beck (2001b). First-excursion probabilities for linear systems by very efficient importance sampling. *Probabilistic Engineering Mechanics 16(3)*, 193–207.

Doob, J. L. (1953). *Stochastic processes*. New York: Wiley.

Gelman, A., G. O. Roberts, & W. R. Gilks (1996). Efficient metropolis jumping rules. *Bayesian Statistics 5*, 599–607.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika 57(1)*, 97–109.

Katafygiotis, L. S. & S. H. Cheung (2007). Application of spherical subset simulation method and auxiliary domain method on a benchmark reliability study. *Structural Safety 3*, 194–207.

Katafygiotis, L. S., T. Moan, & S. H. Cheung (2007). Auxiliary domain method for solving multi-objective dynamic reliability problems for nonlinear structures. *Structural Engineering and Mechanics 25(3)*, 347–363.

Koutsourelakis, P., H. J. Pradlwarter, & G. I. Schuëller (2004). Reliability of structures in high dimensions, part i: algorithms and applications. *Probabilistic Engineering Mechanics 19(4)*, 409–417.

Liu, J. S. (2001). *Monte Carlo Strategies is Scientific Computing*. New York: Springer-Verlag.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, & E. Teller (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys. 21(6)*, 1087–1092.

Neal, R. M. (1993). Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto.

Robert, C. P. & G. Casella (2004). *Monte Carlo Statistical Methods*. New York: Springer-Verlag.

Zuev, K. M. & L. S. Katafygiotis (in press). Horseracing simulation algorithm for evaluation of small failure probabilities. *Probabilistic Engineering Mechanics*.