

Modified Metropolis-Hastings algorithm with delayed rejection

K.M. Zuev & L.S. Katafygiotis

Department of Civil Engineering, Hong Kong University of Science and Technology, Hong Kong, China

ABSTRACT:

The development of an efficient MCMC strategy for sampling from complex distributions is a difficult task that needs to be solved for calculating small failure probabilities encountered in high-dimensional reliability analysis of engineering systems. Usually different variations of the Metropolis-Hastings algorithm (MH) are used. However, the standard MH algorithm does generally not work in high dimensions, since it leads to very frequent repeated samples. In order to overcome this deficiency one can use the Modified Metropolis-Hastings algorithm (MMH) proposed in Au & Beck 2001. Another variation of the MH algorithm, called Metropolis-Hastings algorithm with delayed rejection (MHDR) has been proposed by Tierney & Mira 1999. The key idea behind the MHDR algorithm is to reduce the correlation between states of the Markov chain. In this paper we combine the ideas of MMH and MHDR and propose a novel modification of the MH algorithm, called Modified Metropolis-Hastings algorithm with delayed rejection (MMHDR). The efficiency of the new algorithm is demonstrated with a numerical example where MMHDR is used together with Subset simulation for computing small failure probabilities in high dimensions.

Keywords: Reliability; Subset simulation; Markov chain Monte Carlo; Metropolis-Hastings algorithm.

1 INTRODUCTION

One of the most important and challenging problems in reliability engineering is to compute the failure probability given by the following expression:

$$p_F = \int_{\mathbb{R}^N} I_F(x)\pi(x)dx \quad (1)$$

Here $\pi(\cdot)$ is the joint probability density function (PDF) of a random vector $x \in \mathbb{R}^N$, which represents the uncertain parameters of the problem; $F \subset \mathbb{R}^N$ is the failure domain that is usually defined as $F = \{x \in \mathbb{R}^N | G(x) < 0\}$, where G is called limit-state function; I_F is the indicator function of F , i.e. $I_F(x) = 1$ if $x \in F$ and $I_F(x) = 0$ if $x \notin F$. Throughout this work we assume that the parameter space \mathbb{R}^N is high-dimensional and the failure probability p_F is very small.

Each advanced stochastic simulation algorithm for computation of small failure probabilities (1) encountered in reliability analysis of engineering systems consists of two main steps. At first, we need to specify some artificial PDF(s) $\tilde{\pi}$ from which we are going to sample during the run of the algorithm. For example, in Importance Sampling, Subset Simulation (Au & Beck, 2001) and Adaptive Linked Importance Sampling (Katafygiotis &

Zuev, 2007) we sample from the important sampling density π_{is} , family of conditional distributions $\pi(\cdot|F)$ and family of intermediate distributions π_α respectively. The second step is the development of the Markov chain Monte Carlo (MCMC) strategy for sampling from the above specified artificial PDF(s). Usually different variations of the Metropolis-Hastings (MH) algorithm are used. Schematically the structure of advanced simulation algorithm is shown in Fig. 1.

The main objective of this paper is to develop a novel effective MCMC algorithm for sampling from complex high-dimensional distributions.

Our starting point is the standard MH algorithm (Metropolis et al. 1953, Hastings 1970). It has been shown by Au and Beck (2001) that the standard MH algorithm does generally not work in high dimensions, since it leads to extremely frequent repeated samples. A geometric understanding of why this is true is given in Katafygiotis & Zuev 2008. In order to overcome this deficiency of the MH algorithm one can use the modified Metropolis-Hastings algorithm (MMH) proposed in Au & Beck 2001 for sampling from high-dimensional distributions. It should be emphasized that MMH algorithm is suitable only for sampling from very specific distributions, namely, from the con-

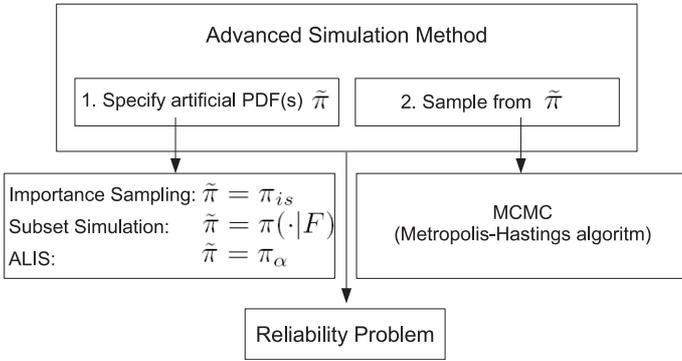


Figure 1: Structure of a general advanced simulation method.

ditional distributions $\pi(\cdot|F)$, where unconditional PDF π can be factorized into a product of easy to sample from one-dimensional distributions. To the best of our knowledge, at the moment there does not exist any efficient algorithm for sampling from an arbitrary high-dimensional distribution.

Another variation of the MH algorithm, called Metropolis-Hastings algorithm with delayed rejection (MHDR) has been proposed in Tierney & Mira 1999. The key idea behind the MHDR algorithm is that when a Markov chain remains in the same state for some time, the estimate obtained by averaging along the chain path becomes less efficient. For the MH algorithm this happens when a candidate generated from the proposal distribution is rejected. Therefore, we can improve the MH algorithm by reducing the number of rejected candidates. A way to achieve this goal is the following: whenever a candidate is rejected, instead of taking the current state of a Markov chain as its new state, as the case in the standard MH algorithm, we propose a new candidate. Of course, the acceptance probability of the new candidate has to be adjusted in order to keep the distribution invariant.

To address high-dimensional reliability problems, in this paper we combine the ideas of both the MMH and MHDR algorithms. As a result we obtain an efficient algorithm, called Modified Metropolis-Hastings algorithm with delayed rejection (MMHDR), for sampling from high-dimensional conditional distributions.

Different variations of the standard MH algorithm are schematically shown in Fig. 2.

2 MODIFICATIONS OF THE METROPOLIS-HASTINGS ALGORITHM

Throughout this work all the variations of the MH algorithm are discussed in the context of high-dimensional conditional distributions.

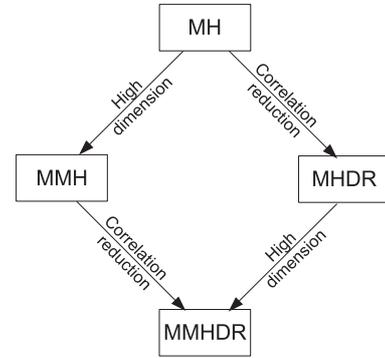


Figure 2: Modifications of the standard Metropolis-Hastings algorithm.

2.1 Standard Metropolis-Hastings algorithm

The MH algorithm is the most common MCMC method for sampling from a probability distribution that is difficult to sample from directly. The algorithm is named after Nicholas Metropolis, who proposed it in 1953 for the specific case of the Boltzmann distribution, and W. Keith Hastings, who generalized it in 1970.

In this method samples are simulated as the states of a Markov chain, which has the target distribution, i.e., the distribution we want to sample from, as its equilibrium distribution. Let the target distribution be $\pi(\cdot|F) = \pi(\cdot)I_F(\cdot)/Z$, where $Z = P(F)$ is a normalizing constant; x_0 be the current state of the Markov chain; and $S(\cdot|x_0)$, called proposed PDF, be an N -dimensional PDF depended on x_0 . Then the MH update $x_0 \rightarrow x_1$ of the Markov chain works as follows:

- 1) Simulate ξ according to $S(\cdot|x_0)$,
- 2) Compute the acceptance probability

$$a(x_0, \xi) = \min \left\{ 1, \frac{\pi(\xi)S(x_0|\xi)}{\pi(x_0)S(\xi|x_0)} I_F(\xi) \right\}, \quad (2)$$

- 3) Accept or reject ξ by setting

$$x_1 = \begin{cases} \xi, & \text{with prob. } a(x_0, \xi); \\ x_0, & \text{with prob. } 1 - a(x_0, \xi). \end{cases} \quad (3)$$

One can show that such update leaves $\pi(\cdot|F)$ invariant, i.e. if x_0 is distributed according to $\pi(\cdot|F)$, then so is x_1 :

$$x_0 \sim \pi(\cdot|F) \Rightarrow x_1 \sim \pi(\cdot|F). \quad (4)$$

Hence the chain will eventually converge to $\pi(\cdot|F)$ as its equilibrium distribution. Note, that the MH algorithm does not require information about the normalizing constant Z . Assuming a symmetric

proposal distribution, i.e. $S(x|y) = S(y|x)$, one obtains the original Metropolis algorithm (Metropolis et al. 1953). The MH update is schematically shown in the Fig. 3.

2.2 Metropolis-Hastings algorithm with delayed rejection

Rejecting the candidate state ξ with probability $1 - a(x_0, \xi)$ in (3) is necessary for keeping the target distribution $\pi(\cdot|F)$ invariant under the MH update. However, remaining in the current state x_0 for some time affects the quality of the corresponding Markov chain by increasing the autocorrelation between its states and, therefore, it reduces the efficiency of any simulation method that uses the standard MH algorithm. Thus, reducing the number of rejected candidate states will improve the standard MH algorithm.

Metropolis-Hastings algorithm with delayed rejection (MHDR), proposed in Tierney & Mira 1999, allows to achieve this goal: when a reject decision in (3) is taken, instead of getting a repeated sample, we generate a second candidate state using a different proposal distribution and accept or reject it based on a suitably computed probability. So, the Markov chain update $x_0 \rightarrow x_1$ in the MHDR algorithm when dealing with conditional distribution $\pi(\cdot|F)$ works as follows:

1) Simulate ξ_1 according to $S_1(\cdot|x_0)$,

2) Compute the acceptance probability

$$a_1(x_0, \xi_1) = \min \left\{ 1, \frac{\pi(\xi_1)S_1(x_0|\xi_1)}{\pi(x_0)S_1(\xi_1|x_0)} I_F(\xi_1) \right\}, \quad (5)$$

3) Accept or reject ξ_1 by setting

$$x_1 = \begin{cases} \xi_1, & \text{with prob. } a_1(x_0, \xi_1); \\ \text{go to step 4) with prob. } 1 - a_1(x_0, \xi_1). \end{cases} \quad (6)$$

4) Simulate ξ_2 according to $S_2(\cdot|x_0, \xi_1)$,

5) Compute the acceptance probability

$$a_2(x_0, \xi_1, \xi_2) = \min \left\{ 1, \frac{\pi(\xi_2)S_1(\xi_1|\xi_2)}{\pi(x_0)S_1(\xi_1|x_0)} \times \frac{S_2(x_0|\xi_2, \xi_1)(1 - a_1(\xi_2, \xi_1))}{S_2(\xi_2|x_0, \xi_1)(1 - a_1(x_0, \xi_1))} I_F(\xi_2) \right\}, \quad (7)$$

6) Accept or reject ξ_2 by setting

$$x_1 = \begin{cases} \xi_2, & \text{with prob. } a_2(x_0, \xi_1, \xi_2); \\ x_0, & \text{with prob. } 1 - a_2(x_0, \xi_1, \xi_2). \end{cases} \quad (8)$$

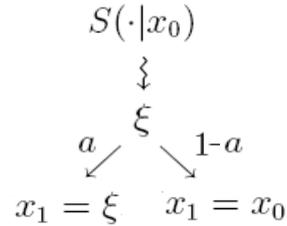


Figure 3: Standard Metropolis-Hastings algorithm.

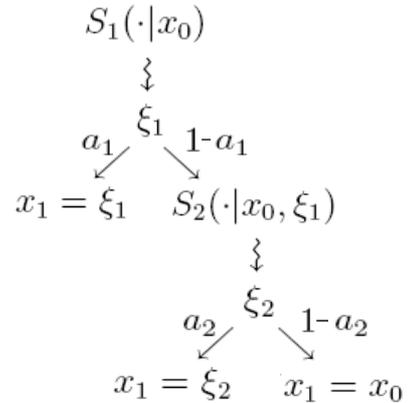


Figure 4: Metropolis-Hastings algorithm with delayed rejection.

An interesting feature of this algorithm is that the proposal distribution S_2 at the second stage is allowed to depend on the rejected candidate ξ_1 as well as on the current state x_0 of the chain. Allowing the proposal PDF S_2 to use information about previously rejected candidate does not destroy the Markovian property of the sampler. So all the asymptotic Markov chain theory used for the standard MH algorithm can be used for the MHDR method as well. The MHDR update is schematically shown in Fig. 4.

Whether MHDR algorithm is useful depends on whether the reduction in variance achieved compensates for the additional computational cost.

2.3 Modified Metropolis-Hastings algorithm

The standard MH algorithm does not generally work in high dimensions meaning that with extremely high probability the update of the Markov chain leads to the repeated sample, $x_1 = x_0$ (Au & Beck 2001, Katafygiotis & Zuev 2008). Clearly, the MHDR update has the same problem. Thus, a Markov chain of practically meaningful length constructed in high dimensions having applied either MH or MHDR algorithm may consist of as few as a single sample. This renders simulation methods, such as Subset Simulation, practically inapplicable.

The modified Metropolis-Hastings algorithm (MMH) was developed in Au & Beck 2001 espe-

cially for sampling from high-dimensional conditional distributions. The MMH algorithm differs from the standard MH algorithm in the way the candidate state ξ is generated. Instead of using an N -dimensional proposal PDF S to directly obtain the candidate state ξ , in the MMH algorithm a sequence of one-dimensional proposals $S^j(\cdot|x_0^j), j = 1 \dots N$ is used. Namely, each coordinate ξ^j of the candidate state is generated separately using a one-dimensional proposal distribution $S^j(\cdot|x_0^j)$ depended on the j th coordinate x_0^j of the current state. Finally, we check whether the generated candidate belongs to the failure domain or not. Thus, the MMH update of the Markov chain works as follows:

1) Generate candidate state ξ .

For each $j = 1 \dots N$:

1a) Simulate $\hat{\xi}^j$ according to $S^j(\cdot|x_0^j)$

1b) Compute the acceptance probability

$$a^j(x_0^j, \hat{\xi}^j) = \min \left\{ 1, \frac{\pi_j(\hat{\xi}^j) S^j(x_0^j | \hat{\xi}^j)}{\pi_j(x_0^j) S^j(\hat{\xi}^j | x_0^j)} \right\}, \quad (9)$$

1c) Accept or reject $\hat{\xi}^j$ by setting

$$\xi^j = \begin{cases} \hat{\xi}^j, & \text{with prob. } a^j(x_0^j, \hat{\xi}^j); \\ x_0^j, & \text{with prob. } 1 - a^j(x_0^j, \hat{\xi}^j). \end{cases} \quad (10)$$

2) Accept or reject ξ by setting

$$x_1 = \begin{cases} \xi, & \text{if } \xi \in F; \\ x_0, & \text{if } \xi \notin F. \end{cases} \quad (11)$$

It can be easily seen that the MMH algorithm overcomes the deficiency of standard MH algorithm by producing distinct Markov chain states, rather than repeated samples (Au & Beck 2001, Katafygiotis & Zuev 2008). The MMH update is schematically shown in Fig. 5.

2.4 Modified Metropolis-Hastings algorithm with delayed rejection

In this paper we propose a new MCMC method, called modified Metropolis-Hastings algorithm with delayed rejection (MMHDR), which combines the ideas of both MMH and MHDR algorithms.

Let $\xi_1 = (\xi_1^1, \dots, \xi_1^N)$ be a candidate state generated during the MMH update. Divide the set of all indexes $I = \{1, \dots, N\}$ into two disjoint subsets: $I = T \sqcup \bar{T}$, where $T = \{j \in I : \xi_1^j = \hat{\xi}_1^j\}$ and $\bar{T} = \{j \in I : \xi_1^j = x_0^j\}$. So, T is a set of all

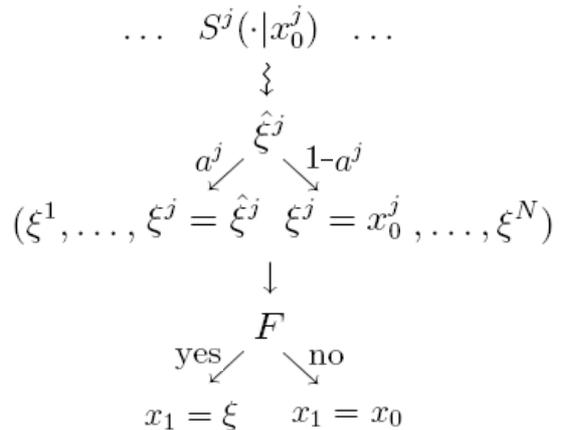


Figure 5: Modified Metropolis-Hastings algorithm.

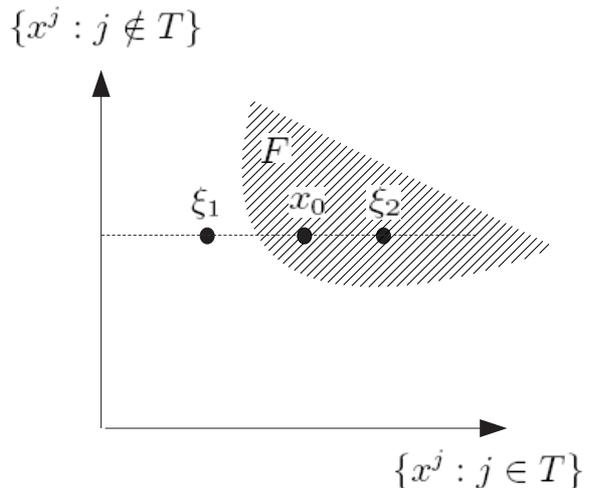


Figure 6: MMHDR update at the second stage.

indexes such that the corresponding coordinates of the candidate state were really transformed and \bar{T} is a set of all the remaining indexes.

Following the MMH algorithm after the candidate is generated we need to check whether it belongs to the failure domain or not. In the former case we accept the candidate as a new state of the Markov chain, in the latter case we reject the candidate and get a repeated sample. In the MMHDR algorithm when a reject decision in (11) is taken, instead of getting a repeated sample, we generate a second candidate state ξ_2 using a different one-dimensional proposal distribution S_2^j for each $j \in T$ and take $\xi_2^j = x_0^j$ for all $j \in \bar{T}$. In other words, at the second stage we try to update only those coordinates of the current state x_0 that have been transformed at the first stage already. Schematically this is shown in Fig. 6. Finally, when the second candidate is generated, we check whether it belongs to the failure domain, in which case we obtain a really new state of the Markov chain, or not, in which case we still get a repeated sample.

So, the MMHDR update $x_0 \rightarrow x_1$ of the Markov chain works as follows:

1) Generate candidate state ξ_1 .

For each $j = 1 \dots N$:

1a) Simulate $\hat{\xi}_1^j$ according to $S_1^j(\cdot|x_0^j)$

1b) Compute the acceptance probability

$$a_1^j(x_0^j, \hat{\xi}_1^j) = \min \left\{ 1, \frac{\pi_j(\hat{\xi}_1^j) S_1^j(x_0^j | \hat{\xi}_1^j)}{\pi_j(x_0^j) S_1^j(\hat{\xi}_1^j | x_0^j)} \right\}, \quad (12)$$

1c) Accept or reject $\hat{\xi}_1^j$ by setting

$$\xi_1^j = \begin{cases} \hat{\xi}_1^j, & j \in T \quad \text{with prob. } a_1^j(x_0^j, \hat{\xi}_1^j); \\ x_0^j, & j \in \bar{T} \quad \text{with prob. } 1 - a_1^j(x_0^j, \hat{\xi}_1^j). \end{cases} \quad (13)$$

2) Accept or reject ξ_1 by setting

$$x_1 = \begin{cases} \xi_1, & \text{if } \xi \in F; \\ \text{go to step 3)} & \text{if } \xi \notin F. \end{cases} \quad (14)$$

3) Generate candidate state ξ_2 .

For each $j = 1 \dots N$:

If $j \in \bar{T}$, set $\xi_2^j = x_0^j$; if $j \in T$, do

3a) Simulate $\hat{\xi}_2^j$ according to $S_2^j(\cdot|x_0^j, \xi_1^j)$

3b) Compute the acceptance probability

$$a_2^j(x_0^j, \xi_1^j, \hat{\xi}_2^j) = \min \left\{ 1, \frac{\pi_j(\hat{\xi}_2^j) S_1^j(\xi_1^j | \hat{\xi}_2^j)}{\pi_j(x_0^j) S_1^j(\xi_1^j | x_0^j)} \times \frac{S_2^j(x_0^j | \hat{\xi}_2^j, \xi_1^j) a_1^j(\hat{\xi}_2^j, \xi_1^j)}{S_2^j(\hat{\xi}_2^j | x_0^j, \xi_1^j) a_1^j(x_0^j, \xi_1^j)} \right\}, \quad (15)$$

3c) Accept or reject $\hat{\xi}_2^j$ by setting

$$\xi_2^j = \begin{cases} \hat{\xi}_2^j, & \text{with prob. } a_2^j(x_0^j, \xi_1^j, \hat{\xi}_2^j); \\ x_0^j, & \text{with prob. } 1 - a_2^j(x_0^j, \xi_1^j, \hat{\xi}_2^j). \end{cases} \quad (16)$$

4) Accept or reject ξ_2 by setting

$$x_1 = \begin{cases} \xi_2, & \text{if } \xi_2 \in F; \\ x_0, & \text{if } \xi_2 \notin F. \end{cases} \quad (17)$$

The MMHDR update is schematically shown in the Fig. 7. It can be proved that if x_0 is distributed according to $\pi(\cdot|F)$, then so is x_1 , i.e. the MMHDR update leaves distribution $\pi(\cdot|F)$ invariant. Because of the limited space we omit here the formal proof and it will be given in the corresponding journal paper.

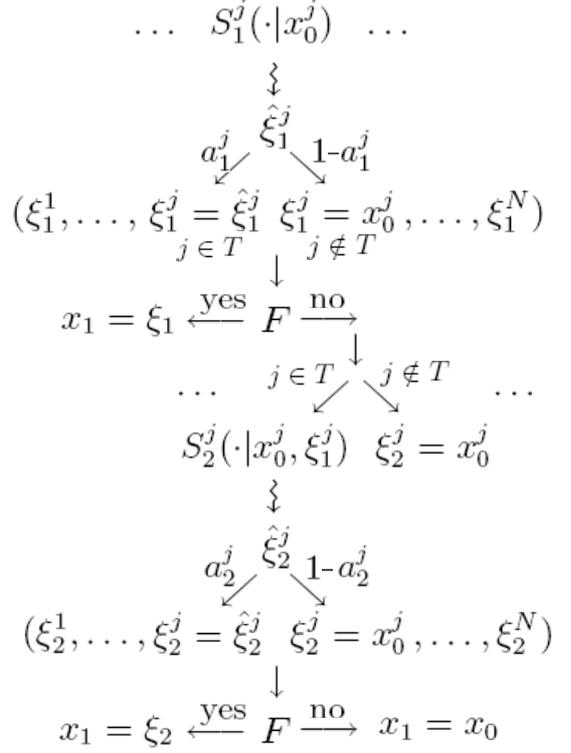


Figure 7: Modified Metropolis-Hastings algorithms with delayed rejection.

The MMHDR algorithm preserves an attractive feature of the MHDR algorithm. Namely, one-dimensional proposal distributions at the second stage are allowed to depend on the corresponding coordinates of the first rejected candidate. The usage of information on the previously rejected candidate potentially can help us to generate a better candidate at the second stage that will be accepted as a new state of the Markov chain. This certainly reduces the overall probability of remaining in the current state if compare with the MMH algorithm, and therefore leads to an improved sampler. However, this improvement is achieved at the expense of an additional computational cost. Whether the MMHDR algorithm is useful for solving reliability problems depends on whether the gained reduction in variance compensates for the additional computational effort.

3 EXAMPLE

To demonstrate the advantage of the MMHDR algorithm over the MMH algorithm we apply Subset Simulation (SS), Au & Beck 2001, with both MMHDR and MMH algorithms for evaluating the small failure probability of linear problem in high dimensions.

Let $N = 1000$ be the dimension of the linear problem, $p_F = 10^{-5}$ be the failure probability and the failure domain F is defined as

$$F = \{x \in \mathbb{R}^N : \langle x, e \rangle \geq \beta\}, \quad (18)$$

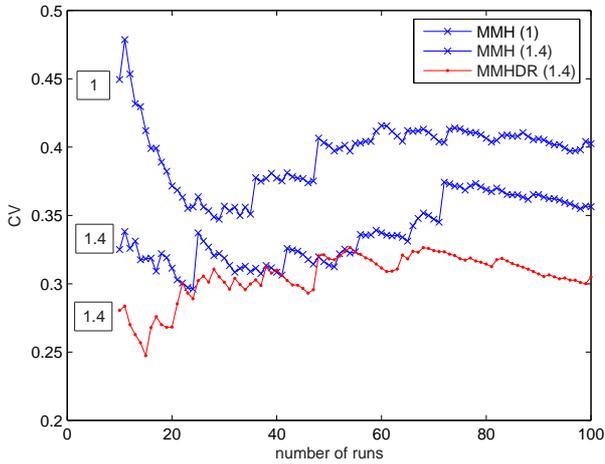


Figure 8: The CV of the estimates obtained by Subset Simulation with MMH and MMHADR algorithms.

where $e \in \mathbb{R}^N$ is a random unit vector drawn from uniform distribution and $\beta = \Phi^{-1}(1 - p_F) = 4.265$ is the reliability index. Here Φ denotes the CDF of the standard normal distribution. Note, that $x^* = e\beta$ is the design point of the failure domain F .

All one-dimensional proposal distributions in both MMH and MMHADR algorithms are set to be normal distributions with unit variance and centered at the corresponding coordinates of the current state:

$$S^j(\cdot|x_0^j) = S_1^j(\cdot|x_0^j) = S_2^j(\cdot|x_0^j, \xi_1^j) = \mathcal{N}_{x_{0,1}^j}(\cdot) \quad (19)$$

Here, when MMHADR algorithm is used, we do not change the second stage proposal distributions. How to generate a better candidate at the second stage based on the information on the first rejected candidate is in need of additional research. In this example we want just to check which one of the following two strategies is more effective: to have more Markov chains with more correlated states (MMH) or to have fewer Markov chains with less correlated states (MMHADR).

The coefficient of variation (CV) of the failure probability estimates obtained by SS method against the number of runs are given in Fig. 8. The curve denoted as MMH(1) corresponds to the SS with MMH algorithm where for each intermediate subset $n = 1000$ samples are used. We refer to the total computational cost of this method, i.e., the mean of the total number of samples used, as 1. The curve denoted as MMHADR(1.4) corresponds to the SS with MMHADR algorithm where 1000 of MMHADR updates are performed per each intermediate subset. It turns out that the total computational cost of MMHADR(1.4) is 40% higher

than MMH(1) and the reduction in CV achieved is about 25% (based on 100 of runs). Finally, the curve MMH(1.4) corresponds to the SS with MMH algorithm where for each intermediate subset $n = 1450$ samples are used. The total computational cost of MMH(1.4) is the same as for MMHADR(1.4), i.e. 40% higher than for MMH(1). However, the the reduction in CV achieved with MMH(1.4) compared to MMH(1) is about 11% only.

So, in this example SS with MMHADR algorithm clearly outperforms SS with MMH algorithm.

4 CONCLUSIONS

In this paper a novel modification of the MH algorithm, called Modified Metropolis-Hastings algorithm with delayed rejection (MMHADR), is proposed. Based on the well-known sampling techniques: modified Metropolis-Hastings algorithm (Au & Beck 2001) and Metropolis-Hastings algorithm with delayed rejection (Tierney & Mira 1999) the new algorithm is designed specially for sampling from high dimensional conditional distributions. The efficiency of the MMHADR algorithm is demonstrated with a numerical example where MMHADR is used together with Subset simulation for computing small failure probabilities in high dimensions.

5 ACKNOWLEDGMENTS

This research has been supported by the Hong Kong Research Grants Council under grant 614305. This support is gratefully acknowledged.

REFERENCES

- Au S.K. & Beck J.L. 2001. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*; 16(4), 263-277.
- Hastings W.K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97-109.
- Katafygiotis L.S. & Zuev K.M. 2008. Geometric insight into the challenges of solving high-dimensional reliability. *Probabilistic Engineering Mechanics*, 23, 208-218.
- Katafygiotis L.S. & Zuev K.M. 2007. Estimation of small failure probabilities in high dimensions by Adaptive Linked Importance Sampling, COMPDYN 2007, Rethymno Crete Greece, 13-16 June 2007.
- Metropolis N. et al. 1953. Equation of State Calculations by Fast Computing Machines. *J. Chemical Physics*, vol. 21, pp. 1087-1092.
- Tierney L. & Mira A. 1999. Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in Medicine*; 18, 2507-2515.