

MARKOV CHAIN MONTE CARLO METHODS FOR UNCERTAINTY PROPAGATION AND RELIABILITY ANALYSIS

Carsten Proppe¹

¹Chair of Engineering Mechanics, Karlsruhe Institute of Technology
Kaiserstr. 10, Bdg. 10.23, 76131 Karlsruhe, Germany
e-mail: proppe@kit.edu

Keywords: Reliability Estimation, Markov Chain Monte Carlo Simulation, Subset Simulation, Moving Particles Algorithm.

Abstract. *Two Markov chain Monte Carlo simulation methods for reliability estimation, subset simulation and the moving particles algorithm, are compared. To this end, both low-dimensional and high-dimensional test cases are considered. The investigation discusses differences in the efficiency of both algorithms and sheds light on the parameter settings as well as the avoidance of correlated samples in both algorithms, notably on the necessity of a burn-in period and the influence of a metamodel.*

1 INTRODUCTION

In direct Monte Carlo Simulation (MCS), independent and identically distributed samples are drawn in order to obtain an unbiased estimate of a function of random variables. For MCS, the strong law of large numbers yields the P -almost sure convergence of the estimator, and from the central limit theorem, one can deduce that the variance of the estimator decreases with $1/N$, where N is the number of samples. In particular, the decrease does not depend on the number of random variables. Unfortunately, direct MCS is not well suited for reliability analysis, where low failure probabilities must be estimated, because a huge number of samples is needed in order to obtain accurate estimates. This leads in turn to a prohibitively large number of calls to the performance function; and if the performance function itself requires a rather large computational time (which is in general the case in structural analysis), the total computational time will be excessive.

Generating samples from a fixed importance sampling density aims to reduce the variance of the MCS estimator while retaining the advantages of independent and identically distributed samples. Thus, the strong law of large numbers and the central limit theorem still apply, yielding the same convergence properties as for direct MCS, but with a reduced variance of the estimator due to a judicious choice of the importance sampling density. For reliability analysis, heuristics to obtain an appropriate importance sampling density have been discussed in the literature, see e.g. [1]. However, as has been shown in [2], sampling with a *fixed* importance sampling density becomes inefficient for reliability estimation involving a high-dimensional random vector and thus a large number of random variables.

Adaptive importance sampling densities may be constructed by Markov Chain Monte Carlo (MCMC) algorithms. In this case, the samples are not independent (but still identically distributed, if the Markov chain is in stationary state), but the ergodic theorem and the central limit theorem for reversible Markov chains yield the convergence and the asymptotic unbiasedness of the estimator. An MCMC based algorithm that is widely used for reliability estimation is subset simulation [3], which is based on the estimation of conditional probabilities for a nested sequence of sets. In contrast to many other MCMC algorithms, subset simulation does not require a burn-in of the Markov chain, because the seeds of the Markov chains are already distributed according to the target distribution [4]. On the other hand, the classical Metropolis-Hastings (MH) algorithm suffers from a high rejection rate in conjunction with subset simulation, especially for high-dimensional reliability estimation, cf. [2]. Therefore, a componentwise MH algorithm has been introduced in [3] and more recently, direct sampling from a normal transition kernel has been proposed which yields candidates that always differ from the current state [5] and leads to an improved efficiency for high dimensional problems, if the proposal distribution is adapted for each subset.

In subset simulation, the size of the nested sets is usually chosen such that the conditional probabilities are equal to 0.1, i.e. 90% of the samples are discarded and need to be recomputed from the remaining 10% that serve as seeds. This leads in general to less than ten nested sets. Having too few nested sets would lead to a similar inefficiency as for direct MCS. In [3], it is argued that having much more nested sets would lead to an increase in the total number of samples, which again decreases the efficiency of subset simulation. However, it is worthwhile to consider the limit case, where only a single sample is discarded and recomputed. This leads to a maximum number of nested sets. Such an MCMC based algorithm, called moving particles algorithm, has been introduced in [6] and [7].

The objective of this contribution is to compare subset simulation and the moving particles

algorithm for reliability estimation on low- as well as on high-dimensional test cases with emphasis on the accuracy, efficiency and the acceptance rate and thus to shed some light on specific features of these two MCMC based algorithms. The paper is organized as follows: in the next section, subset simulation and the moving particles algorithm are introduced. Following this, the test cases and the test methodology are presented. The test results are discussed and explanations for the observed phenomena are given. Finally, a summary of the observed phenomena and recommendations for off-the-shelf application of the simulation algorithms are given.

2 MARKOV CHAIN MONTE CARLO SIMULATION

2.1 Subset simulation

Subset Simulation is based on nested sets, $F_1 \supset F_2 \supset \dots \supset F_M$, where $F_M = \{\theta \in \mathbb{R}^n | g(\theta) < 0\}$ denotes the failure region and $g(\theta)$ is the performance function. If θ is a vector of random variables, the failure probability is given by

$$P_f = P_{F_1} \prod_{i=1}^{M-1} P(F_{i+1}|F_i). \quad (1)$$

The rather small failure probability is written as the product of larger probabilities that can be estimated with less effort. However, the estimation of the conditional probabilities requires the application of MCMC simulations, because the corresponding conditional probability density function is not known explicitly. In subset simulation, N_s parallel Markov chains are started from seeds that for step $i - 1$ lie in F_i . Generating the Markov chains with the classical MH algorithm may result in low acceptance rates. Alternatives are the componentwise MH algorithm [3], a single repetition of the candidate sample (delayed rejection) or the direct sampling from a suitable transition kernel such that rejection is limited to the case where the candidate sample is not an element of F_i .

The most influential parameters of subset simulation are the transition kernel of the MCMC algorithm and the sets F_i , for which sets of equal conditional probability $p_0 = P(F_{i+1}|F_i)$ are preferred. Given p_0 , the sets are obtained from a percentile estimation for the performance function. The failure probability is then given by

$$p_0^{M-1} \hat{P}_M, \quad (2)$$

where \hat{P}_M is the estimate for $P(F_M|F_{M-1})$.

The coefficient of variation of the estimator for the conditional probability $P_{i+1} = P(F_{i+1}|F_i)$ is given by [3]

$$\sqrt{\frac{1 - P_i}{NP_i}} (1 + \gamma_i), \quad (3)$$

where the additional term γ_i is

$$\gamma_i = 2 \sum_{k=1}^{N/N_s-1} \left(1 - \frac{kN_s}{N}\right) \rho_i(k). \quad (4)$$

$N_s = p_0 N$ is the number of samples that for step $i - 1$ lie in F_i and constitute the seeds of the Markov chains. $\rho_i(k)$ is the correlation coefficient of the series $I_{F_i}(\theta_{jk}^{(i-1)})$, where $\theta_{jk}^{(i-1)}$ is the k th sample of the j Markov chain that is generated in step $i - 1$. A weak correlation of the

samples produced by the Markov chain is thus necessary for a reduction of the coefficient of variation.

The coefficient of variation for the failure probability estimated with subset simulation can be approximated by

$$\delta_{sub} \approx \sqrt{\frac{\log p_F (1 - p_0)(1 + \bar{\gamma})}{\log p_0 N p_0}}, \quad (5)$$

where $\bar{\gamma}$ is the average value of γ_i (averaged over the number of subsets) and $\frac{\log p_F}{\log p_0}$ represents the number of subsets. A typical value for $\bar{\gamma}$ is $\frac{1}{1+\bar{\gamma}} = 0.4$, cf. [5]. The average number of function evaluation is thus approximated by

$$N_{sub} = N \left(1 + (1 - p_0) \frac{\log p_F}{\log p_0} \right), \quad (6)$$

and is composed by the initial Monte Carlo samples and the Markov chain samples (without burn-in) at each step.

2.2 Moving particles algorithm

The moving particles algorithm can be considered as subset simulation with a maximum number of steps. Thus, in each step, only one sample is discarded and resampled by a Markov chain that takes as seed one of the retained samples. As for subset simulation, the algorithm starts with a direct MCS. For each step, the values $g(\theta_i)$, $i = 1, \dots, N_m$, of the N_m samples are ranked. The sample with the maximum value of the performance function is moved: MCMC is carried out starting from one of the remaining samples and the final state of the Markov chain is accepted, if the value of the performance function could be reduced. Otherwise, the sample is simply replaced by the seed of the Markov chain. The Markov chain can be generated either by application of the MH algorithm or by direct sampling from a normal transition kernel. As the classical MH algorithm suffers from low acceptance rates for high-dimensional problems, the componentwise MH algorithm of [3] is applied in the following. Thus, the same Markov chain simulation technique is applied in this study for the moving particles algorithm and subset simulation.

However, instead of computing the probability of failure from eq. (2), each initial sample is moved until it reaches the failure region and the number of moves is counted. As has been shown in [6], the number of moves to get an initial sample into the failure region follows a Poisson distribution with parameter $\lambda = \log \frac{1}{p_F}$. The estimator for the parameter of the Poisson distribution is

$$\hat{\lambda} = \frac{\sum_{i=1}^{N_m} M_i}{N_m}, \quad (7)$$

where M_i denotes the number of moves until seed i reaches the failure state.

In order to obtain an unbiased estimate, it is mandatory that the trajectories of the Poisson process generated from the initial samples remain independent until the samples finally reach the failure domain. In [7], two means are proposed to maintain the independence:

- Burn-in: The Markov chain simulation is carried out with a burn-in period. The burn-in should ensure the independence of the candidate and the seed of the Markov chain.
- Seed avoidance: Repeated use of the same seed for the Markov chain should be avoided. Once a sample has been used as seed, the sample and its offspring should not be used as seed again.

The coefficient of variation for the failure probability estimated with the moving particles algorithm is given by

$$\delta_{mp} = \sqrt{\frac{-\log p_F}{N_m}}, \quad (8)$$

cf. [6], and the average number of function evaluations is

$$N_{mp} = N_m(1 - T \log p_F), \quad (9)$$

where the first term accounts for the initial Monte Carlo simulation and the second term for the Markov chain samples (with burn-in period T).

2.3 Comparison of the two algorithms

While the moving particles algorithm can be considered as a special case of subset simulation, there are several differences with respect to the original subset simulation algorithm:

- In subset simulation, the number of steps is rather small; however, the number of steps in the moving particles algorithm is maximal.
- In subset simulation, only p_0 (usually 10%) of the samples are retained in each step and serve as seed for the Markov chains. In contrast, only one sample is resampled in each step of the moving particles algorithm, and the seed can be selected among the other samples.
- The moving particles algorithm has a clear interpretation from Poisson process theory, but requires that all initial samples finally reach the failure region.

3 TEST CASES

Test cases have been considered with standardized parameters for the algorithms in order to investigate the robustness, accuracy and efficiency of both simulation methods. The aim was to find out whether the simulation methods can be applied off-the-shelf without fine tuning to a specific problem. The low-dimensional test cases were those summarized in [9], Table 1. They allowed to carry out parameter studies for the number of initial samples and the Markov chain transition kernel. Moreover, for the moving particles algorithm, the influence of a burn-in period, a seed selection strategy and a low-rank metamodel have been investigated. Following this, both algorithms were applied to high-dimensional examples, a paraboloid with discretely varying principal curvatures, [10], example 4, and a single-degree-of-freedom oscillator with 1500 random variables, [3], example 1, with a threshold value $b = 1.5$.

For each test case and each set of parameters, 100 simulation runs were carried out from which the mean probability of failure, the coefficient of variation, the mean number of performance function evaluations and the mean acceptance rate has been observed.

For the MCMC with MH algorithm, two acceptance rates can be defined:

- Level 1: Acceptance of the pre-candidate by the MH algorithm.
- Level 2: Acceptance of the candidate sample. For subset simulation, the candidate sample at step i must lie in F_i , while for the moving particles algorithm, the value of the performance function must decrease.

In this investigation, the level 2 acceptance rate has been monitored only.

The simulation platform was Matlab with FERUM [11]. To this end, the moving particles simulation algorithm has been added to FERUM and extensions to the subset simulation algorithm were made.

4 RESULTS

4.1 Number of initial samples

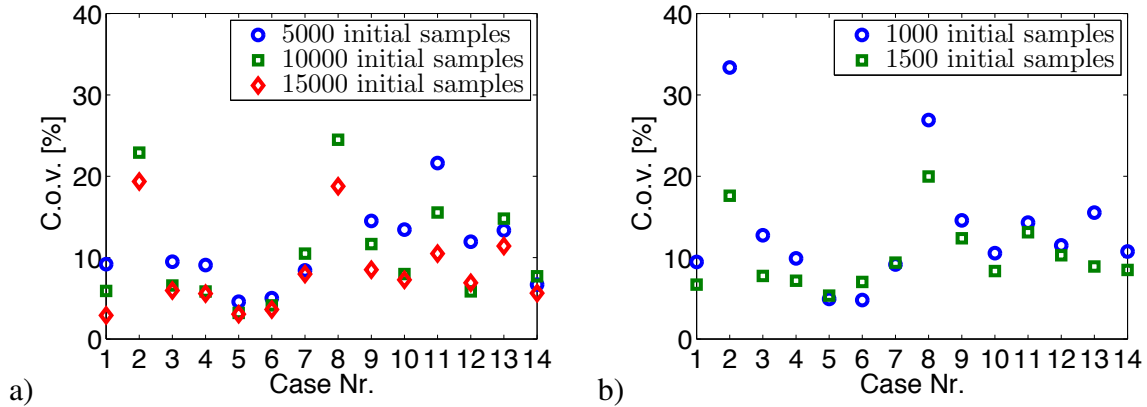


Figure 1: Influence of the number of initial samples. a) Subset simulation. b) Moving particles algorithm.

Figure 1 displays the influence of the number of initial samples on the coefficient of variation of the simulation algorithms for the low-dimensional test cases. It can be seen that the number of initial samples needed to obtain the same level for the coefficient of variation differs by nearly a factor of ten, which is due to the fact that for subset simulation, 90% of the samples are discarded in each simulation step. In fact, considering equations (5) and (8) with $\delta_{sub} = \delta_{mp}$, $\frac{1}{1+\gamma} = 0.4$ and $p_0 = 0.1$, one obtains $N_m \approx 0.1N$.

While increasing the number of initial samples increases the accuracy, the efficiency is decreased. The number of initial samples does not influence the mean level 2 acceptance rate very much.

4.2 Influence of the proposal density

As proposal density for the componentwise MH algorithm, a normal probability density function has been chosen. Figure 2 a) and b) displays the influence of the standard deviation on the coefficient of variation of the simulation algorithm. The figure shows the relative coefficient of variation that is computed by taking as reference the coefficient of variation obtained for a normal density with standard deviation of 0.7. It can be seen that increasing the standard deviation decreases the coefficient of variation for subset simulation, but increases the coefficient of variation for the moving particles algorithm. Thus, in comparison to subset simulation, a smaller standard deviation for the proposal density is required for the moving particles algorithm. This can be explained by the fact that the Markov chain simulations in the moving particles algorithm require a burn-in period and thus a repeated application of the proposal density, which is not the case for subset simulation.

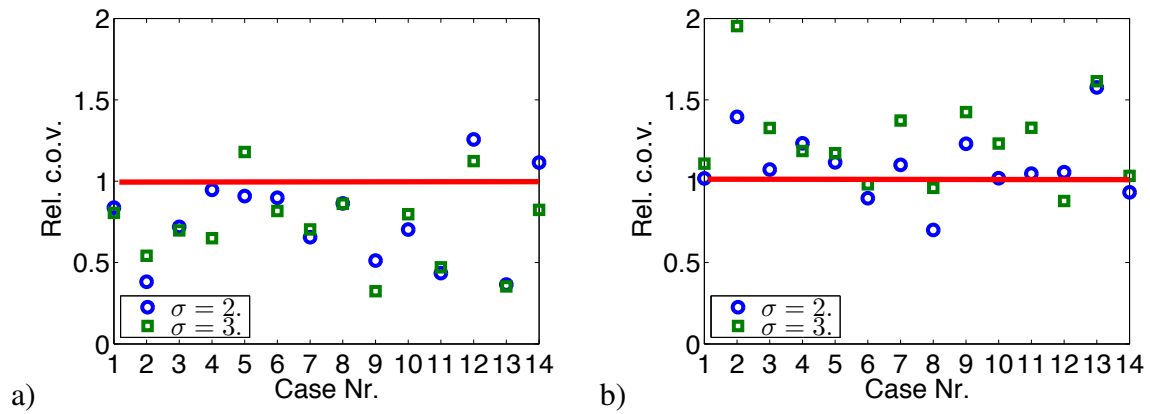


Figure 2: Influence of the standard deviation σ of the proposal density. a) Subset simulation. b) Moving particles algorithm.

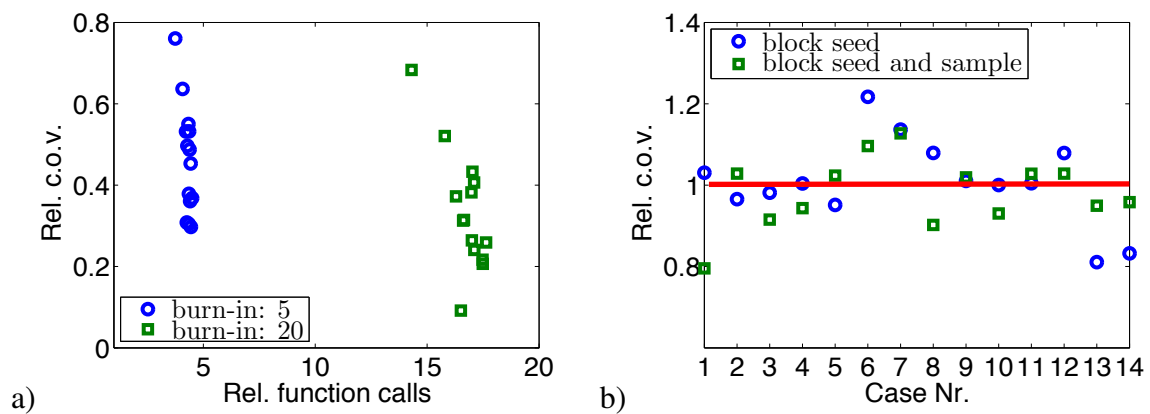


Figure 3: Moving particles algorithm. a) Influence of the burn-in period. b) Influence of seed selection strategies.

4.3 Moving particles algorithm: burn-in period and seed selection strategy

Figure 3 summarizes the influence of the burn-in period on the efficiency and various seed selection strategies on the accuracy of the moving particles algorithm. For the investigation of the burn-in period, simulations without burn-in have been taken as reference and the relative coefficient of variation is shown together with the increase in the number of function calls. The results underline that a burn-in period is necessary for a reduction of the coefficient of variation. However, a burn-in period of 20 samples as proposed in the literature leads to a high number of performance function calls. It was found that a shorter burn-in period (about 5 samples) is a good compromise between accuracy and effort.

Various seed selection strategies have been tested. Figure 3 b) states the results for two seed selection strategies. Simulations without a seed selection strategy were taken as reference. The results indicate that seed avoidance might be beneficial for the accuracy of the simulation algorithm and that blocking the seed of the Markov chain and the generated sample (the offspring) for further use as seed of a Markov chain might be a viable strategy to increase the efficiency. However, the effect is not very pronounced.

Neither the length of the burn-in period nor the seed selection strategy had a significant influence on the level 2 acceptance rate.

4.4 Moving particles algorithm: application of a low rank metamodel

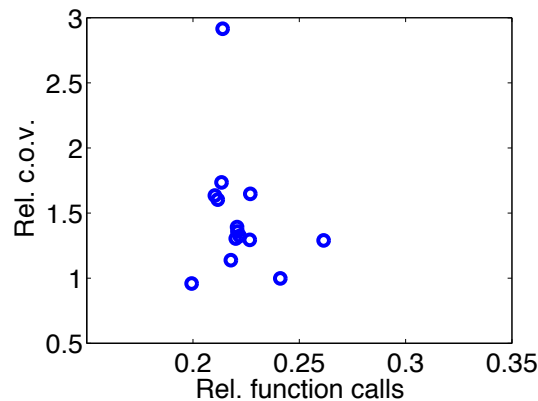


Figure 4: Moving particles algorithm. Application of a metamodel.

In order to reduce the number of function calls during the burn-in period of the Markov chains, the application of a metamodel has been investigated. A low rank separated representation with at most 50 polynomial terms of order 5 has been calibrated from the initial samples and continuously updated (as described in [7]) during the simulations. Results have been compared to computations without a metamodel. The relative coefficient of variation and the relative number of function calls have been collected for the test cases and are shown in Figure 4.

As can be seen from Figure 4, the application of a metamodel during the burn-in period of the Markov chains for the moving particles algorithm decreases the number of performance function calls, but increases at the same time the coefficient of variation. Moreover, there is a considerable scatter in the increase of the coefficient of variation.

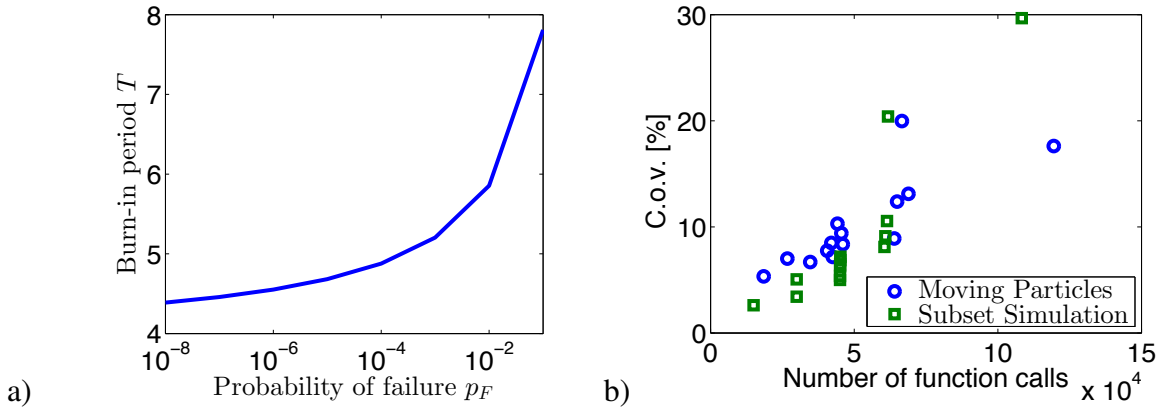


Figure 5: Comparison of subset simulation algorithm and moving particles algorithm. a) Burn-in period, such that the number of performance function evaluations of subset simulation and the moving particles algorithm is approximately the same. b) Comparison of the efficiency of subset simulation and the moving particles algorithm.

4.5 Comparison of subset simulation and moving particles algorithm

The efficiency of both algorithms can be compared by setting $N_m = 0.1N$ for the number of initial samples and $p_0 = 0.1$ as before. In this case, the coefficient of variation of both algorithms will be nearly the same. From equations (6) and (9), one obtains the relationship

$$\frac{N_{sub}}{N_{mp}} = \frac{10(1 - 0.39 \log p_F)}{1 - T \log p_F} \quad (10)$$

By setting this expression to one, a burn-in period T can be obtained as a function of the failure probability for which both algorithms would require approximately the same amount of function evaluations. This relationship is depicted in Fig. 5 a). It can be seen that the obtained burn-in period is in the range of values that has been found to be sufficient in the previous section and thus the number of function evaluations for both algorithms is of the same order of magnitude. For small failure probabilities, subset simulation becomes slightly more efficient than the moving particles algorithm, while for larger probabilities of failure, the opposite is the case.

The comparison of the two variants of the moving particles algorithm and subset simulation, Figure 5 b), underlines that both algorithms lead to a similar efficiency. However, while the number of function evaluations is approximately the same, the coefficient of variation for the results from the moving particles algorithm is slightly higher than for subset simulation. As a burn-in period of $T = 5$ has been applied for the moving particles algorithm and the failure probabilities are in the range from 10^{-4} to 10^{-6} , this confirms the results of Figure 5 a).

Finally, both algorithms have been applied to high dimensional test cases, a single-degree-of-freedom oscillator with 1500 random variables and a paraboloid with varying curvatures. Both algorithms yielded results of same accuracy and efficiency. The level 2 acceptance rates were similar as for the low-dimensional test cases.

5 CONCLUSIONS

The aim of the simulation study was to investigate the suitability of two MCMC simulation algorithms, subset simulation and the moving particles algorithm, for off-the-shelf reliability computations.

The following conclusions can be drawn: A large initial sample size is beneficial for the accuracy of both algorithms, but decreases the efficiency in terms of performance function evaluations. Comparing the subset simulation algorithm to the moving particles algorithm, it was found that the subset simulation algorithm needs about a factor of ten more initial samples, as most of the initial samples are discarded during the move to the next subset simulation step. For the componentwise MH transition kernel, both algorithms tolerate a wide range of values for the standard deviation of the normal distribution. In order to obtain accurate results, the standard deviation for subset simulation must be higher than for the moving particles algorithm. A burn-in period is necessary in order to obtain a good accuracy of the moving particles algorithm. A seed selection strategy might increase the accuracy of the moving particles algorithm, but has nearly no influence on the level 2 acceptance rate.

For the efficiency, the differences between the moving particles and the subset simulation algorithm were rather small for the low- as well as the high-dimensional test cases. In conclusion, subset simulation and the moving particles algorithm seem to be well suited for off-the-shelf reliability estimations, but with a different setting for the most important parameters (σ and the initial sample size).

REFERENCES

- [1] G. I. Schuëller, R. Stix, A critical appraisal of methods to determine failure probabilities, *Structural Safety* 4 (1987) 293–309.
- [2] L. S. Katafygiotis, K. M. Zuev, Geometric insight into the challenges of solving high-dimensional reliability problems, *Probabilistic Engineering Mechanics* 23 (2008) 208–218.
- [3] S. K. Au, J. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, *Probabilistic Engineering Mechanics* 16 (2001) 263–277.
- [4] S. K. Au, J. Beck, K. M. Zuev, L. S. Katafygiotis, Discussion of paper by F. Miao and M. Ghosn "Modified subset simulation method for reliability analysis of structural systems", *structural safety*, 33:251260, 2011, *Structural Safety* 34 (2012) 379–380.
- [5] I. Papaioannou, W. Betz, K. Zwirgmaier, D. Straub, MCMC algorithms for subset simulation, *Probabilistic Engineering Mechanics* 41 (2015) 89–103.
- [6] A. Guyader, N. Hengartner, E. Matzner-Løber, Simulation and estimation of extreme quantiles and extreme probabilities, *Applied Mathematics & Optimization* 64 (2011) 171–196.
- [7] C. Walter, Moving particles: A parallel optimal multilevel splitting method with application in quantiles estimation and meta-model based algorithms, *Structural Safety* 55 (2015) 10–25.
- [8] K. M. Zuev, L. S. Katafygiotis, Estimation of small failure probabilities in high dimensions by adaptive linked importance sampling, in: *Proceedings Computational Methods in Structural Dynamics and Earthquake Engineering COMPDYN-2007*, Rethymno, Crete, Greece, 2007.

- [9] F. Grootemann, An adaptive directional importance sampling method for structural reliability, *Probabilistic Engineering Mechanics* 26 (2011) 134–141.
- [10] A. Der Kiureghian, H.-Z. Lin, S.-J. Hwang, Second-order reliability approximations, *Journal of Engineering Mechanics* 113 (1987) 1208–1225.
- [11] J.-M. Bourinet, C. Mattrand, V. Dubourg, A review of recent features and improvements added to FERUM software, in: *Proceedings of the 10th International Conference on Structural Safety and Reliability (ICOSSAR09)*, Osaka, Japan, 2007.
- [12] K. M. Zuev, L. S. Katafygiotis, Modified Metropolis-Hastings algorithm with delayed rejection, *Probabilistic Engineering Mechanics* 26 (2011) 405–412.