Efficient Monte Carlo Algorithm For Rare Failure Event Simulation

Edoardo Patelli
Lecturer, Institute for Risk and Uncertainty, University of Liverpool, United Kingdom

Siu Kui Au
Professor, Institute for Risk and Uncertainty, University of Liverpool, United Kingdom

ABSTRACT: Studying failure scenarios allows one to gain insights into their cause and consequence, providing information for effective mitigation, contingency planning and improving system resilience. A new efficient algorithm is here proposed to solve applications where an expensive-to-evaluate computer model is involved. The algorithms allows to generate the conditional samples for the Subset simulation by representing each random variable by an arbitrary number of hidden variables. The resulting algorithm is very simple yet powerful and it does not required the use of the Markov Chain Monte Carlo method. The proposed algorithm has been implemented in a open source general purpose software, OpenCossan allowing the solution of large scale problems of industrial interest by taking advantages of High Performance Computing facilities. The applicability and flexibility of the proposed approach is shown by solving a number of different problems.

1. INTRODUCTION

Rare failure events of safety critical systems (such as in Nuclear and Aviation industry) can have huge impacts as shown by recent accidents (e.g. Tohoku Earthquake and the consequent Fukushima-Daiichi accident).

Assessing risk quantitatively requires the quantification of the probability of occurrence of a specific event by properly propagating the uncertainty through the model that predicts the quantities of interest. In principle, rare failure events can be investigated through Monte Carlo simulation (see e.g. Liu (2001)). However, this is computationally prohibitive for complex systems because it requires a large number of samples to obtain one failure sample. Checking whether each sample fails requires evaluating of the model and the calculation of output quantities, which is generally computationally expensive for complex systems.

Advanced Monte Carlo methods aim at estimating rare failure probabilities more efficiently than direct Monte Carlo (see e.g. Schuëller (2009)). Unfortunately, high dimension and model complexity make it extremely difficult to improve the efficiency of Monte Carlo algorithms purely based on prior knowledge, leaving algorithms that adapt the generation of samples during simulation the only choice. The estimation of small probabilities of failure from computer simulations is a classical problem in engineering, and the Subset Simulation algorithm proposed by Au and Beck (2001) has become one of the most popular method to solve it thanks to its significant savings in the number of simulations to achieve a given accuracy of estimation, with respect to many other Monte Carlo approaches.

In this paper, a new efficient algorithm for Subset simulation is proposed to generate conditional samples without using Markov Chain Monte Carlo. This allows to simplify the Subset simulation and to remove one of the main sequential components of Subset simulation. The proposed algorithm has been implemented in a open source general purpose software, OpenCossan (Patelli et al., 2014). The computational framework allows the application of the novel algorithm to solve large scale examples of practical interest and taking advantages of High Performance Computing facilities. A number of academic examples and real applications of industrial interest are solved and presented to show the applicability of the proposed approach.

2. SUBSET SIMULATION

In this section a short description of the original Subset algorithm and the proposed algorithm is presented.

2.1. *Original algorithm: Subset-MCMC*

Subset simulation (Au and Beck, 2001) is an advanced Monte Carlo method aimed at estimating rare failure probabilities more efficiently than direct Monte Carlo. The method has been already applied efficiently to a wide range of applications (e.g. Alvarez et al. (2014); Chiachio et al. (2014); Ching and Hsieh (2007)) since it is not based on any geometrical assumption about the topology of the failure domain.

The key idea of Subset simulation is to decompose the failure event F into a sequence of nested failure events: $F = F_m \in F_{m-1} \in \cdots \in F_1 : F = \cap_i F_i$. The probability of failure is expressed as the product of $P(F_1)$ and the conditional probabilities $P(F_{k+1}/F_k), k = \{1,\ldots,m-1\}$:

$$P(F) = P(\cap_i F_i) = P(F_1)\prod_{i=1}^{m-1} P(F_{k+1}/F_k). \quad (1)$$

During subset simulation the threshold values, $\delta_1,\ldots,\delta_m$, are adaptively generated so that the conditional failure probabilities F_k are sufficiently large. Hence, by choosing m and F_k appropriately, the conditional probabilities can be evaluated efficiently by direct simulation.

The challenging part of the Subset algorithm is the realization of the conditional samples $\mathbf{X}$ that are distributed according the conditional probabilities $P(\mathbf{x}|F_i)$. In the original implementation of Subset, the generation of conditional samples are obtained adopting a independent-component Markov Chain Monte Carlo (MCMC) algorithm. Using a modified Metropolis algorithm Metropolis and Ulam (1949), the chains are generated in two steps. First a next state is sampled from a proposal distribution $\mathbf{x}' \leftarrow \pi(\mathbf{x})$. Then, the candidate solution is accepted if its associated performance function is greater than the intermediate threshold level δ_k otherwise the candidate sample is rejected.

In general, controlling the MCMC algorithm is not a trivial task. Furthermore, the generation of the conditional samples based on Markov Chains represents the bottleneck for the parallelization and scalability of the algorithms on cluster and grid computing.

2.2. *Proposed algorithm: Subset-∞*

The main idea of the proposed algorithm (hereafter indicated as Subset-∞) is to define an equivalent problem where each random variable X_i is represented by an arbitrary number of hidden variables. In fact, Gaussian variables can be represented by an infinite number of Gaussian variables since a linear combination of Gaussian variables is still Gaussian. In addition, any problem can be represented in the so-called Standard Normal Space by means of a transformation (see e.g. (Nataf, 1962)) where each input variable is represented by an independent Gaussian distribution with 0 mean and unitary standard deviation.

Studying the limiting behaviour of the independent-component MCMC, Au (2015) has demonstrated that the conditional distribution of the candidate samples is a Gaussian distribution with mean and variance that depends on the proposal distribution. Thanks to this result, conditional samples can be directly sampled from an appropriated Gaussian distribution without resorting to the MCMC algorithm and the selection of the proposal distribution.

2.2.1. *Numerical implementation*

The main advantage of the Subset-∞ algorithm relies on its simplicity and scalability. In fact, the Subset-∞ does not require the construction of Markov Chains but only the capability to generate realizations in the standard normal space from normal distributions.

The following pseudo-algorithm generates samples distributed as the target conditional distribution of failure $F(x|F_k)$ in the Standard Normal space:

1. assume a sample $\mathbf{X}^{(i)}$ is distributed as $F(x|F_k)$
2. Calculate $\mathbf{a} = \sqrt{1-\mathbf{s}^2}$ where $\mathbf{s} = [s_1,\ldots,s_n]$ represents the vector of chosen variances for each component of the candidate X_n
3. Generate the candidate $\mathbf{X}' \sim N(\mathbf{a}\cdot\mathbf{X}^{(i)},\mathbf{s})$
4. The new sample $\mathbf{X}^{(i+1)} = \mathbf{X}'$ if $\mathbf{X}' \in F_k$ otherwise $\mathbf{X}^{(i+1)} = \mathbf{X}^{(i)}$ if $\mathbf{X}' \notin F_k$

5. Repeat for all samples
6. Repeat for all Subset level

The proposed algorithm is generally applicable for any finite dimensional problem that can be represented in standard normal space.

The proposed algorithm has been implemented in OPENCOSSAN software (Patelli et al., 2014). OPENCOSSAN is a collection of open source algorithms, methods and tools released under the LGPL license (Free Software Foundation, 2007), and under continuous development at the Institute for Risk and Uncertainty at the University of Liverpool, UK. The source code is freely available upon request at the web address `http://www.cossan.co.uk`. Thanks to the modularity and structure of the software organized in classes, i.e. data structures consisting of data fields and methods together with their interactions and interfaces, the implementation of the Subset-∞ algorithm has required the implementation of only few lines of code.

2.2.2. *Parallelization and scalability*

Generally, reliability analysis requires to evaluate of the model a large number of time. Although Subset simulation allows a significant reduction of the number of model calls, the wall clock time required by the analysis can only be further reduced resorting to the parallel execution of the code. Multiple independent instances of the solver are executed simultaneously for different realizations of the input, allowing for a reduction of the analyses time without any loss of accuracy.

MCMC is an inherently serial algorithm and hence it requires complex algorithms to parallelize the samples generation such as e.g. the speculative computing approaches (Pellissetti, 2009) or partitioning and weight estimation scheme (VanDerwerken and Schmidler, 2013). In contrast, the proposed ∞-algorithm is easily paralellelizable and two types of parallelization can be used. The first type of parallelization is used to speed-up the analysis in case the solver (i.e. the model to be evaluated) is a coded in Matlab. In this case, a special job on a pool of MATLAB workers is created on each multi-core machine, connecting the MATLAB client to the parallel pool (e.g. using the command `parpool`). Features from the MATLAB parallel toolbox e.g., `parfor`, can be used to distribute the tasks on the MATLAB clients. This type of parallelism can be implemented on each single computational node. In case the analysis requires the call of an external solver (such as a FE/CFD analysis) the multi-thread, shared memory parallelism capabilities of the external software need to be adopted. The second level of parallelization exploits *cluster and grid computing*, i.e. the availability of machines connected in an heterogeneous network. In this case, the total number of simulations is slitted in a multiple number of independent batch jobs. The jobs are then submitted to the job scheduler/manager and distributed efficiently on the available machines of the grid/cluster.

Using OPENCOSSAN framework, the parallelization of the analysis is straightforward. Independent multiple stream and sub-stream are generated by combined multiple recursive generator (`mrg32k3a`). Then, OPENCOSSAN creates independent jobs by compiling portion of OPENCOSSAN using `mcc` and then distribute the compiled code to the node of the cluster (workers). Hence, it is possible to execute jobs in parallel without the necessity to install MATLAB on each computational node of the cluster, but only accessing the MATLAB runtime libraries.

3. NUMERICAL EXAMPLES

In this Section different numerical examples are presented to show the applicability of the Subset-∞ algorithm.

3.1. *Simple synthetic model*

The purpose of this first numerical example is to show the applicability of the new proposed algorithm using a very simple problem. Although, it is a very simple example and consider only one input, it represents an extreme case for the proposed approach since it has been designed to be efficient for high dimensional problems. Nevertheless, it allows to show the applicability of the methods and clearly visualize and compare the conditional samples obtained using the Subset-∞ algorithm with samples obtained by means of the standard Subset-MCMC approach.

In this example the reliability problem is defined as the probability of the random variable X_1 to exceed a constant threshold. The limit state function is defined as:

$$g_1(X) = X_1 - 3 \quad (2)$$

where X_1 is a normally distributed variable with 0 mean and 1 standard deviation. The estimation of the probability of failure has been computed by means of the Subset simulation adopting the standard MCMC implementation and the proposed ∞-algorithm, respectively. The results are summarized in Table 2. The Subset-MCMC simulations have been performed using an target conditional failure probability of 0.1, a uniform proposal distribution with width 0.4 while Subset-∞ simulations have been performed using a variance $s = 0.5$. The main Subset parameters are summarized in Table 1.

Table 1: Parameters of the Subset algorithm used for solving the numerical example simple synthetic model.

Parameter	Value
Initial samples size	100
Target cond. failure probability	0.1
Proposal dist. (Subset-MCMC)	U(-0.2,0.2)
Variance s (Subset-∞)	0.5

Table 2: Estimated failure probability of the simple synthetic problem.

	Analytical	MCMC	Subset-∞
$\hat{p_f}$	$1.3 \cdot 10^{-3}$	$1.32 \cdot 10^{-3}$	$1.6\dot{1}0^{-3}$
Std	-	$9.53 \cdot 10^{-4}$	$1.08 \cdot 10^{-3}$
Samples	-	440	160

Figures 1-2 show the values of the computed performance function g_1 as a function of the subset level using MCMC and ∞ algorithm, respectively. The horizontal lines represent the identified threshold values, $\delta_1, \ldots, \delta_m$; the stars represent rejected samples while circles show the accepted conditional samples.

Figure 3 shows the effect of the proposed variance s for the Subset-∞ algorithm on the estimation of the failure probability. For each value of s 20

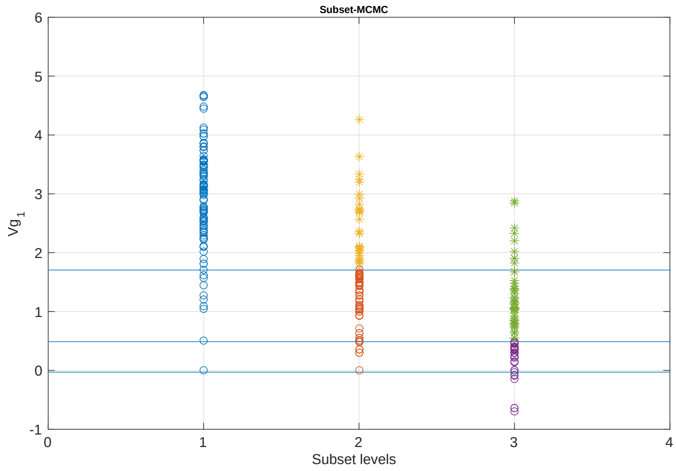

Figure 1: Performance function estimated by means of the Subset-MCMC algorithm.

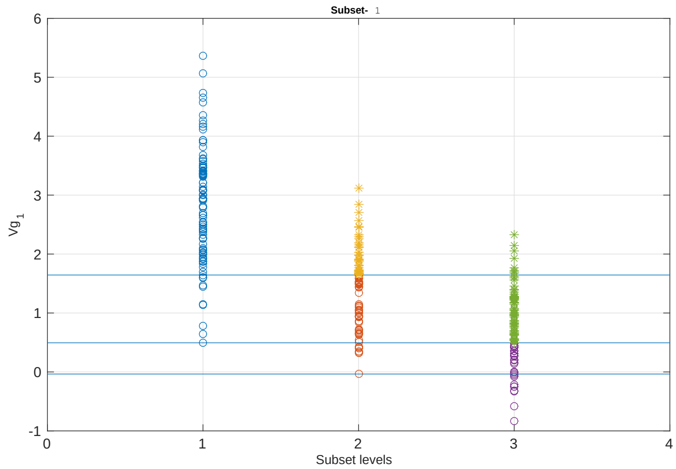

Figure 2: Performance function estimated by means of the Subset-∞ algorithm.

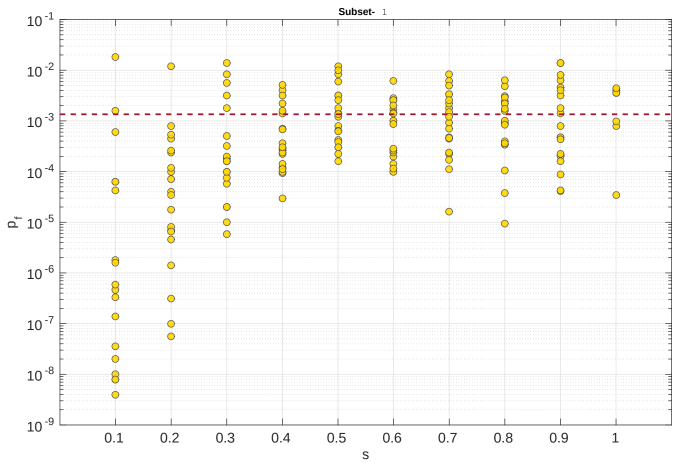

Figure 3: Effect of the proposed variance s of the estimation of $\hat{P}_f$ using the Subset-∞ algorithm.

independent runs of Subset simulation have been performed. The red dashed line represents the analytical solution, the yellow dots the $\hat{P}_f$ estimation performed by the Subset-∞.

3.2. Simple synthetic model in high-dimension

In order to test the performance of the proposed approach, the dimensionality of the simple problem presented in Section 3.1 has been incremented up to 2500 variables. The same limit state function of the previous model has been used (i.e. Eq. (2)). Hence, only the input X_1 controls the failure probability.

Subset simulations have been repeated 10 times and the minimum, median and maximum of the failure estimation is summarized in Table 3 using MCMC algorithm with a uniform proposed distribution of width 0.4 (in standard normal space) and in Table 4 using the canonical algorithm with a proposed variance of $s = 0.5$, respectively. Figure 4 shows the estimation of the failure probability for different number of input variables using the Subset-MCMC and Subset-∞, respectively. As expected, both algorithms show similar results. They seem to be insensitive to the dimensionality of the problem.

Table 3: Estimation of the failure probability computed using Subset-MCMC for different number of input variables. Subset simulation has been repeated 10 times.

Dimension	Min	Median	Max
10	$2.37 \cdot 10^{-5}$	$8.13 \cdot 10^{-4}$	$3.1 \cdot 10^{-3}$
100	$2.88 \cdot 10^{-4}$	$2.17 \cdot 10^{-3}$	$6.3 \cdot 10^{-3}$
1000	$8.48 \cdot 10^{-5}$	$1.20 \cdot 10^{-3}$	$4.5 \cdot 10^{-3}$

Table 4: Estimation of the failure probability computed using Subset-∞ algorithm for different number of input variables. Subset simulation has been repeated 10 times.

Dimension	Min	Median	Max
10	$2.80 \cdot 10^{-4}$	$1.35 \cdot 10^{-3}$	$4.5 \cdot 10^{-3}$
100	$1.50 \cdot 10^{-4}$	$1.15 \cdot 10^{-3}$	$5.6 \cdot 10^{-3}$
1000	$2.4 \cdot 10^{-4}$	$5.45 \cdot 10^{-4}$	$4.7 \cdot 10^{-3}$

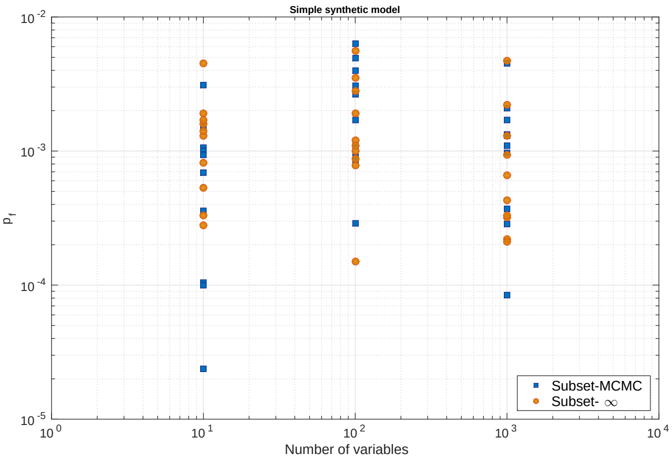

Figure 4: Estimation of the failure probability, $\hat{P}_f$, for the simple synthetic model estimated by means of Subset-MCMC and Subset-∞ simulation for different number of input variables, respectively.

Figure 5 shows the computational time (wall clock time) required by the Subset simulations as a function of the number of variables. The computational time has been calculated using OpenCossan revision 769 on a quad-core Intel Core i5-4590T CPU @2.00GHz. As shown in Figure 5, the Subset-∞ algorithm is one order of magnitude faster than the standard Subset-MCMC procedure.

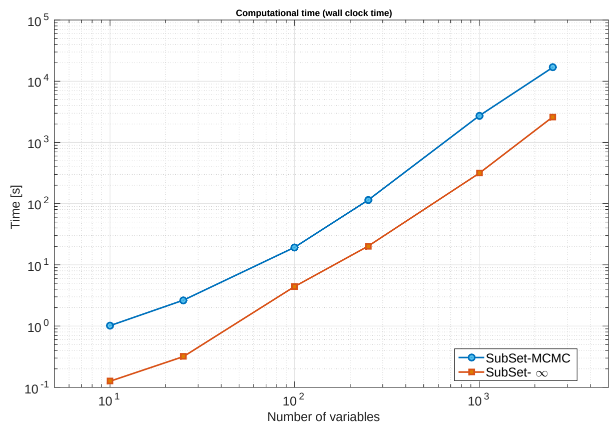

Figure 5: Computational time required by the Subset-MCMC and Subset-∞ as a function of the number of input variables, respectively.

3.3. Simple frame

In this numerical example applies the reliability analysis of a simple structural frame system sketched in Figure 6 and originally presented in

Schuëller et al. (1989) is presented. The structural parameters are summarized in Table 5.

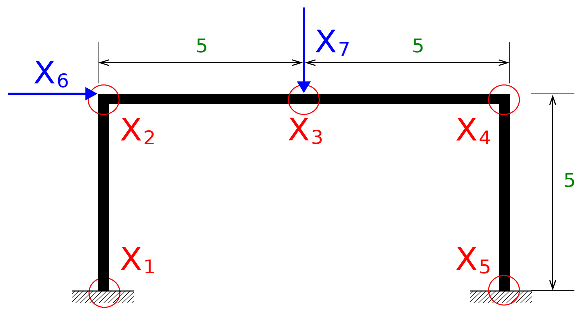

Figure 6: Simple structural frame.

The structural failure criterion is defined by the development of any of the 3 possible collapse mechanisms as illustrated in Figure 7. Hence, the limit state function for the structure is the combination of the 3 limit state functions that describe each collapse mechanism. This can be represented as a systems with components in parallel where the failure of a single component produce the failure of the entire system. Hence the performance function of the system is:

$$g_2(\mathbf{X}) = min(g_{2a}, g_{2b}, g_{2c}) \tag{3}$$

where

$$g_{2a}(\mathbf{X}) = X_1 + 2X_3 + 2X_4 + X_5 - 5X_6 - 5X_7 \tag{4}$$
$$g_{2b}(\mathbf{X}) = X_1 + 2X_2 + X_4 + X_5 - 5X_6 \tag{5}$$
$$g_{2c}(\mathbf{X}) = X_2 + 2X_3 + X_4 - 5X_6 \tag{6}$$

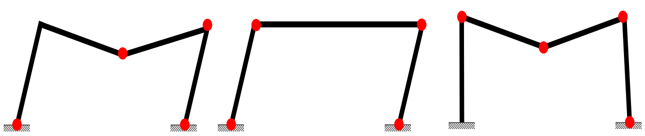

Figure 7: Possible failure modes of the simple structural frame.

Subset simulation has been used to estimate the failure probability using the parameters shown in Table 1. The results of the reliability analysis are summarized in Table 6. The results show that the proposed Subset-∞ approach is able to handle multi non linear limit state functions (in Standard Normal Space).

Table 5: Input variables of the simple structural frame. All the inputs are considered uncorrelated.

Variable	Distribution	Mean	Std
$X_{i=\{1:5\}}$	Lognormal	60	6.0
X_6	Type I - largest	20	6.0
X_7	Type I - largest	20	7.5

Table 6: Estimated failure probability of the simple structural frame.

	MC	SS-MCMC	Subset-∞
$\hat{p}_f$	$1.27 \cdot 10^{-4}$	$3.18 \cdot 10^{-5}$	$1.26 \cdot 10^{-4}$
CoV	$8.87 \cdot 10^{-2}$	$6.43 \cdot 10^{-1}$	$4.87 \cdot 10^{-1}$
Samples	$1 \cdot 10^{6}$	$2.07 \cdot 10^{3}$	$1.8 \cdot 10^{3}$

3.4. *A multi-storey building model*

The applicability of the Subset simulation for solving problems of industrial interest is showing by performing the reliability analysis of a multi-storey building modelled with ABAQUS. The FE-model of the structure as shown in Fig. 8, involves approximately 8,200 elements and 66,300 DOFs, where solid elements (C3D8I) are used for the foundation, the mesh of the floors consists solely of quadrilateral elements (S4) and each of the 16 columns of all floors are modelled with 2-node beam elements (B31).

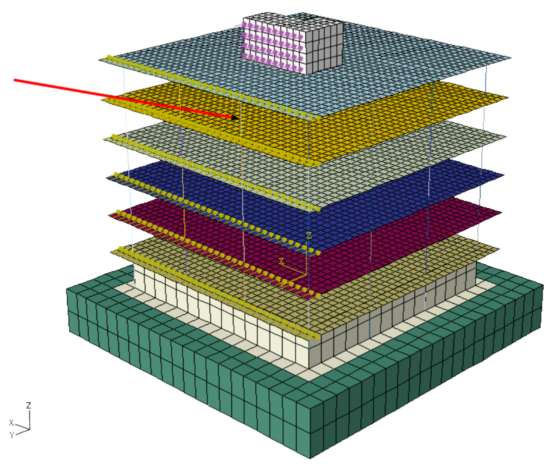

Figure 8: FE-model of a multi-storey building with indication of the loading and the observed column of interest. The FE model has been adapted from Ref. **?**.

The load case under investigation is the combination of self weight and simplified lateral wind load modelled by deterministic concentrated static forces acting on the nodes of one edge of each floor and on the upper part of the staircase where the magnitudes increase with the height of the building. Failure is defined as the exceedance of the yield stress in a bar element of the column of the 5th floor indicated by the arrow in Fig. 8. Hence, the performance function is defined by

$$g(\theta) = \sigma_{\max} - \sigma(\theta), \tag{7}$$

where σ_{max} defines the maximum stress level tolerated by the column (i.e. the resistance) and $\sigma(\theta)$ is the element stress as extracted from the output file of the FE-analysis (i.e. the demand). θ represents the vector of 244 uncertainty structural parameters summarized in Table 7.

Table 7: Random variables used for modelling the uncertainties within the multi-storey building.

Parameter	Value
Column Resistance	$N(7.5e7, 1.0e7)$ Pa
Columns section	$U[0.36, 0.44]$ m
E-modulus	$LN(3.5e10, 3.5e9)$ Pa
Density	$LN(2500, 250)$ kg/m^3
Poisson ratio	$LN(0.25, 0.025)$

The reliability analysis has been performed using Subset simulation. The estimation by means of Monte Carlo simulation is infeasible due to the large sample size needed to trustworthy identify the failure region and the computational cost associated to the analysis of this large FE-model. Subset simulations have been run with 100 initial samples, intermediated failure probability of 0.1 and a uniform proposal distribution with a width of 0.4 for standard Subset-MCMC and a proposal variance of $s = 0.5$ for Subset-∞. The results are summarized in Table 8.

Figures 9 and 10 show the computed column resistance versus the maximum of the element stress in the column as extracted from the output file of the FE-analysis for different levels during Subset simulation. The Subset-MCMC and Subset-∞ provide similar results. They are both applicable to solve

Table 8: Reliability analysis of a FE multi-story building by means of Subset simulation.

	Subset-MCMC	Subset-∞
$\hat{p}_f$	$8.58 \cdot 10^{-6}$	$2.4 \cdot 10^{-7}$
CoV	1.24	1.06
Samples	550	600

large scale problems of practical interest. As already pointed out, the Subset-∞ algorithm is computational faster and more parallelizable than the classic Subset-MCMC and hence it allows to reduce the wall-clock time required by the analysis.

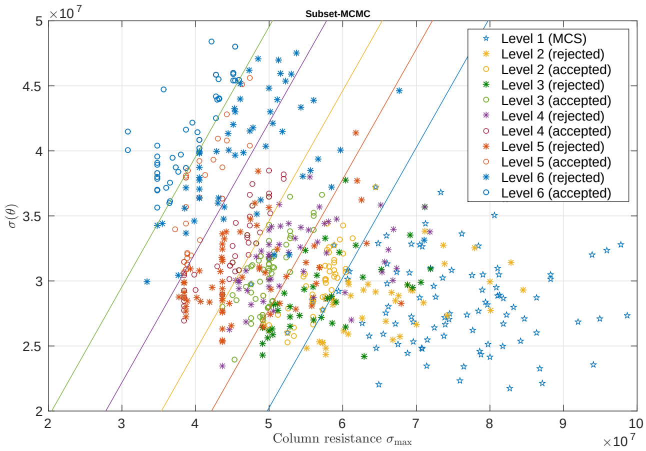

Figure 9: Samples generated by the Subset-MCMC simulation. The lines represent the identified threshold levels.

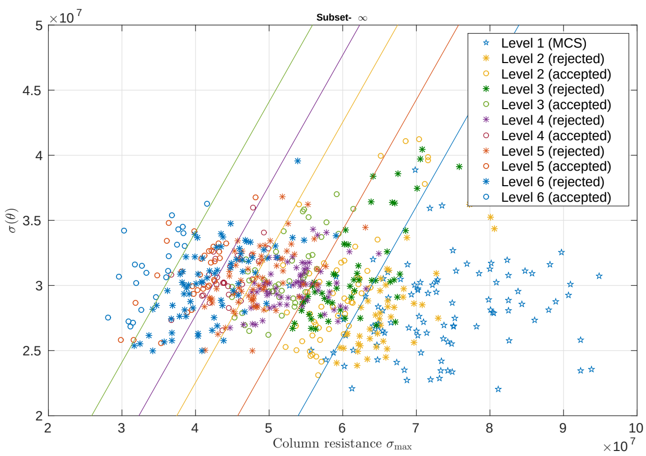

Figure 10: Samples generated by the Subset-∞ simulation. The lines represent the identified threshold levels.

4. CONCLUSIONS

In this paper, a new efficient algorithm for rare failure event simulation has been presented. In the proposed algorithm, denoted as Subset-∞, each random variable is represented by an large number of hidden (Gaussian) variables. As a consequence, the conditional samples can be obtained directly form an appropriate Gaussian distribution without resorting to the Markov Chain Monte Carlo method.

The proposed algorithm has been implemented in OPENCOSSAN. A number of different numerical examples have been presented to show the flexibility and applicability of the approach. In fact, Subset-∞ shows the same accuracy and efficiency of the classic Subset-MCMC. However, Subset-∞ is much simple and faster. In addition, the approach is fully scalable on parallel machines since it does not have the drawback of the sequential Markov Chain Monte Carlo.

5. REFERENCES

Alvarez, D. A., Hurtado, J. E., and Uribe, F. (2014). "Estimation of the lower and upper probabilities of failure using random sets and subset simulation." *Vulnerability, Uncertainty, and Risk*, American Society of Civil Engineers, 905–914.

Au, I. (2015). "Subset simulation in finite-infinite dimensional space. part i: Theory.

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

Chiachio, M., Beck, J. L., Chiachio, J., and Rus, G. (2014). "Approximate bayesian computation by subset simulation." *arXiv preprint arXiv:1404.6225*.

Ching, J. and Hsieh, Y. (2007). "Approximate reliability-based optimization using a three-step approach based on subset simulation." *Journal of Engineering Mechanics*, 133(4), 481–493.

Free Software Foundation (2007). "Free software foundation, gnu lesser general public license, version 3, <http://www.gnu.org/licenses/lgpl.html>.

Liu, J. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer.

Metropolis, N. and Ulam, S. (1949). "The monte carlo method." *Journal of the American Statistical Association*, 44, 335–341.

Nataf, A. (1962). "Détermination des distribution dont les marges sont données." *Comptes rendus de l'academie des sciences*, 225, 42–43.

Patelli, E., Broggi, M., Angelis, M., and Beer, M. (2014). "Opencossan: An efficient open tool for dealing with epistemic and aleatory uncertainties." *Vulnerability, Uncertainty, and Risk*, American Society of Civil Engineers, 2564–2573.

Patelli, E., H.M.Panayirci, Broggi, M., Goller, B., Beaurepaire, P., Pradlwarter, H. J., and Schuëller, G. I. (2012). "General purpose software for efficient uncertainty management of large finite element models." *Finite Elements in Analysis and Design*, 51, 31–48.

Pellissetti, M. (2009). "Parallel processing in structural reliability." *Journal of Structural Engineering and Mechanics*, 32(1), 95–126.

Schuëller, G. (2009). "Efficient Monte Carlo simulation procedures in structural uncertainty and reliability analysis - recent advances." *Journal of Structural Engineering and Mechanics*, 32(1), 1–20.

Schuëller, G. I., Bucher, C., Bourgund, U., and Ouypornprasert, W. (1989). "On efficient computational schemes to calculate structural failure probabilities." *Journal of Probabilistic Engineering Mechanics*, 4(1), 10–18.

VanDerwerken, D. N. and Schmidler, S. C. (2013). "Parallel markov chain monte carlo." *ArXiv e-prints*.