Structural Safety 55 (2015) 10–25

Contents lists available at ScienceDirect

Structural Safety

journal homepage: www.elsevier.com/locate/strusafe

Moving particles: A parallel optimal Multilevel Splitting method with application in quantiles estimation and meta-model based algorithms

Clément Walter *

CEA, DAM, DIF, F-91297 Arpajon, France
Université Paris Diderot, 5 rue Thomas Mann, 75013 Paris, France

ARTICLE INFO

Article history:
Received 12 May 2014
Received in revised form 11 February 2015
Accepted 11 February 2015
Available online 19 March 2015

Keywords:
Rare event simulation
MCMC
Sequential Monte-Carlo
Subset Simulation
Multilevel Splitting
Extreme probabilities and quantiles

ABSTRACT

Considering the issue of estimating *small* probabilities p, *i.e.* measuring a *rare* domain $F = \{\mathbf{x} | g(\mathbf{x}) > q\}$ with respect to the distribution of a random vector $\mathbf{X}$, Multilevel Splitting strategies (also called *Subset Simulation*) aim at writing F as an intersection of *less rare* events (nested subsets) such that their measures are conditionally easily computable. However the definition of an *appropriate* sequence of nested subsets remains an open issue.

We introduce here a new approach to Multilevel Splitting methods in terms of a move of particles in the input space. This allows us to derive two main results: (1) the number of samples required to get a realisation of $\mathbf{X}$ in F is drastically reduced, following a Poisson law with parameter $\log 1/p$ (to be compared with a Geometric law with parameter p for naive Monte-Carlo); and (2) we get a parallel optimal Multilevel Splitting algorithm where there is indeed no subset to define any more.

We also apply result (1) in quantile estimation producing a new parallel algorithm and derive a new strategy for the construction of first Design of Experiments in meta-model based algorithms.

1. Introduction

Context. Extreme events simulation and quantification come from the need to insure that undesirable events will not appear. Typically such events are failure of industrial critical systems, *i.e.* systems for which failure is regarded as a massive catastrophic situation, found in sectors like nuclear safety, aerospace, etc. In this context one could either want to estimate a probability of failure or to define a threshold to insure security with a given confidence. Usually the system is a “black box” whose output determines safety/failure domains.

Formally, let $\mathbf{X}$ be a random vector with values in $\mathbb{R}^d$ and g be a measurable function from $\mathbb{R}^d$ to $\mathbb{R}$ defining the failure domain $F = \{\mathbf{x} \in \mathbb{R}^d \,|\, g(\mathbf{x}) > q\}$, we seek to measure F with respect to the distribution of $\mathbf{X}$: $\mathrm{P}[\mathbf{X} \in F] = \mu^X(F) = p$ or to find back q given p.

Two reasons why this calculation is not obvious is the order of magnitude of the probability (say $p < 10^{-5}$) and the computational time of g output, from couples of hours to several months. In this framework efficiency of the algorithm (precision, global computational time) with respect to good statistical properties of the estimator are of great interest.

Let us first introduce several techniques used so far. Then we will present our new algorithm and its applications in probability and quantile estimation.

Modified Monte-Carlo algorithms. A comprehensive review of Monte-Carlo methods can be found in [1]. On the one hand Importance Sampling [2,3] modifies the distribution of $\mathbf{X}$ to lower the variance of the naive Monte-Carlo estimator; unfortunately the search for an appropriate change of probability is not obvious. In particular it is known that the optimal change depends on the quantity of interest and is thus not directly available.

On the other hand Multilevel Splitting methods consider the failure domain as a finite intersection of nested subsets for which conditional probabilities are *not too small* and so more easily computable: let $(F_k)_k$ be a finite sequence of nested subsets ($F_0 = \mathbb{R}^d$) such that $F = \bigcap_k F_k$, one can write:

$$\mathrm{P}[g(\mathbf{X}) > q] = \mu^X(F) = \prod_k \mu^X(F_k | F_{k-1})$$

In these algorithms the two mains issues are the conditional sampling and the subsets definition. The idea of splitting an event

[*] Tel.: +33 1 69 26 66 75; Cel.: +33 6 65 16 53 97.
E-mail address: walter@math.univ-paris-diderot.fr

http://dx.doi.org/10.1016/j.strusafe.2015.02.002
0167-4730/© 2015 Elsevier Ltd. All rights reserved.

$F = \{g(\mathbf{X}) > q\}$ with a sequence of $(q_m)_m$ such as F can be written as an intersection of nested subsets appeared in the mid 1950's (from Kahn & Harris [4] and Rosenbluth & Rosenbluth [5]). Then Au&Beck [6] brought it to rare event estimation. An in-depth review of these techniques can be found in Glasserman et al. [7]. This algorithm was further improved by Cérou et al. [8] linking it with Feynman-Kac formulae and Céou and Guyader [9] who proposed a method to adaptively select the conditional probabilities. Concerning the conditional simulations, Del Moral et al. [10] introduced reversible transition kernels. Finally Cérou et al. [11] showed that these algorithms were optimal when all conditional probabilities were equal and that an adaptive choice of levels leads to bias in the estimators. Recently Guyader et al. [12] proposed a limit case where the conditional probabilities are fixed to $1 - 1/N$, given N the size of the working population. While they showed that this choice is optimal in terms of computational efficiency, it also disables parallel computation possibility and eventually makes this algorithm longer in practice if multicore computers are available.

Some others practical sampling strategies such as Line Sampling (LS) reduce the problem to the measure of a one-dimensional failure domain [13]. However it is well-known that this methodology performs well only for linear of weakly nonlinear systems [14] and that general methods such as Multilevel Splitting are still required in some strongly nonlinear problems [15].

Meta-model based algorithms. As modified Monte-Carlo methods seen above still require an important number of samples and do not always allow for full parallelisation, meta-model based algorithms propose to spend the computational budget in fitting a surrogate model to the expensive-to-evaluate function g and then to use it instead of the *true* function to compute probability estimation with usual methods [16–18]. Thus theses strategies highly depend on the *quality* of the Design of Experiments (DoE) and especially on their ability to predict the boundary between safety and failure domains, *i.e.* to explore the input space close to the boundary. While space-filling strategies recommend to sample uniformly in the input space (see [19] or [18] chapter 2 for a review of these methods) and thus depend a lot on the dimension of the input space, DoE generated according to the distribution of $\mathbf{X}$ are unlikely to visit the failure domain because of the order of magnitude of the failure probability.

Effective computing time. As mentioned previously, the function of interest g is assumed to be highly demanding in computational time and the number of calls to g, *i.e.* the number of samples used for an estimator, is limited. In this context parallel algorithms (see for examples reference books [20] or [21]) are of great interest. Basically they allow for generating samples for an estimator in a parallel way, *i.e.* that one can get as many samples as available "computers" for the time of one. Thus, to increase the precision of an estimator one can simply use more computers, *i.e.* multi-core processors, to get more samples without making the estimation longer. We will refer to the number of calls to the limit-state function made by one computer as the effective computing time of an algorithm.

Main results. We introduce here a new approach to Multilevel Splitting in terms of a move of particles from an initial random state to the failure domain. This approach brings two main results: first the number of samples needed to get a realisation of $\mathbf{X}$ in the failure domain follows a Poisson law with parameter $\log 1/p$, this is to be compared with a classical Geometric law with parameter p for naive Monte-Carlo; then we get the full parallelisation of the optimal sequential algorithm described by Guyader et al. [12], which turns it into the best Multilevel Splitting algorithm in terms of effective computing time, resolving the issues of choosing a sequence of $(q_m)_m$ or selecting a cut-off probability for the adaptive construction. This new point of view also allows us to propose a modified version of Guyader et al. quantile estimator with a reduced bias and parallel computation.

In the context of meta-model based algorithms (that require DoE with points close to the boundary between safety and failure domains), we use this approach to get a first DoE embedding failing samples while limiting drastically the number of calls to the limit-state function, which depends linearly on the dimension:

$$N_{DoE} = d + 1 + N_{\text{fail}} \log(1/p)$$

with N_{DoE} the size of the first DoE, d the dimension of the input space and N_{fail} the final number of points in the failure domain.

2. Getting into the failure domain

2.1. Introduction

The idea of trying to go *as fast as possible* into the failure domain comes from the need to get failing samples in several methods, from Importance Sampling [22] to meta-model based algorithms. In this latter case it is well noticed [16] that without a first DoE embedding failing samples, the learning of the failure domain is complicated and the final probability estimator rather poor. On the other hand it was tried to merge Multilevel Splitting methods and meta-model based algorithms to increase the precision of conditional probabilities estimation while making easier the learning of the failure domain [17,23]. The paradox was that the final DoE was indeed far too dense in *a posteriori* useless regions.

Then we came up with the idea of *stopping* to try computing the probability estimation on-the-go but only keeping the *moving* part of these algorithms. Finally, Guyader et al. work [12] brings to us the theoretical framework to derive this algorithm.

The problem can be defined as follows: let $\mathbf{X}$ be a random variable with values in $\mathbb{R}^d, d \in \mathbb{N}^*, \mu^X$ its distribution and g a measurable function from $\mathbb{R}^d$ to $\mathbb{R}$ such that the *cdf* of $g(\mathbf{X})$ is continuous.

We first introduce the algorithm in the ideal case, *i.e.* the case where we know how to sample from any distribution when required, and we will then present two implementations to be used depending on the goal (probability and quantile estimation or building of first DoE).

2.2. Ideal case

In this section we consider that it is possible to sample from any distribution when required; thus it is said *ideal.*

2.2.1. Move of one particle

Algorithm 1. Move of one particle.

```
$q_0 = -\infty; m = 0$
loop
  Sample $\mathbf{X} \sim \mu^X(\cdot | g > q_m)$
  Evaluate $g : q_{m+1} = g(\mathbf{X})$
  $m = m + 1$
end loop
```

Thus, Algorithm 1 can be seen as a move of a particle from an initial random place along the levels of g and the sequence of $(q_m)_m$ has indeed an interesting statistical behaviour. Let F_g be the *cdf* of $g(\mathbf{X})$ and Λ the integrated hazard function: $\Lambda(y) = -\log(1 - F_g(y))$; the following results are based on [12]: