Multilevel Monte Carlo for Reliability Theory

Louis J. M. Aslett[a,*], Tigran Nagapetyan[a], Sebastian J. Vollmer[a]

[a]*Department of Statistics, University of Oxford, 24–29 St Giles', Oxford, OX1 3LB, United Kingdom*

Abstract

As the size of engineered systems grows, problems in reliability theory can become computationally challenging, often due to the combinatorial growth in the cut sets. In this paper we demonstrate how Multilevel Monte Carlo (MLMC) — a simulation approach which is typically used for stochastic differential equation models — can be applied in reliability problems by carefully controlling the bias-variance tradeoff in approximating large system behaviour. In this first exposition of MLMC methods in reliability problems we address the canonical problem of estimating the expectation of a functional of system lifetime and show the computational advantages compared to classical Monte Carlo methods. The difference in computational complexity can be orders of magnitude for very large or complicated system structures.

Keywords: reliability theory, multilevel Monte Carlo, cut sets, system lifetime estimation

1. Introduction

It can prove to be computationally intractable to perform classical reliability analysis of very large engineered systems when the number of cut (path) sets grows combinatorially. It is well understood that working instead with subsets of the cut (path) sets or bounding structural designs can provide probability bounds in many reliability problems [2], but such bounds can be crude or may not be well characterised at all.

Evaluation of the reliability of engineered systems is a crucial part of system design and often scenario planning may involve repeated evaluation of the reliability for changing system configurations or component types meaning rapid simulation is highly desirable. For simplicity of exposition we herein consider the canonical problem of estimating the expectation of a functional

*Corresponding author

of system lifetime, though the approach developed is easily generalised to other reliability problems which depend on cut (path) sets for the analysis.

Multilevel Monte Carlo (MLMC) methods — pioneered by Heinrich [10] and Giles [6] — are now standard for estimation of expectations of functionals of processes defined by stochastic differential equations (SDEs). Herein, we show how Multilevel Monte Carlo (MLMC) can be adapted to a reliability setting and develop an MLMC algorithm which can provide substantial speedup in cut set based evaluation of system reliability, extending the size of systems which are in reach for reliability evaluation.

In Section 2, we recap the traditional cut set method of simulating system lifetimes which does not scale well to large systems even when the cut sets are known. This motivates the approach taken in this work. In Section 3 we recap standard Monte Carlo theory and set out the error and computational cost metrics which will enable comparison with our MLMC based approach. The fundamental MLMC methodology and our adaptation to the reliability setting then follow in Section 4, before numerical results demonstrating the kind of substantial computational improvements which can be achieved are covered in Section 5.

2. Simulating system lifetimes

Consider a coherent system with n components. Let $x_1(t), \ldots, x_n(t)$ denote the operational status (1 = working, 0 = failed) of the components at time t and consider the random variable for the lifetime of component c to be $T_c \sim F_c(t)$, where $F_c(\cdot)$ are positively supported lifetime distributions which are not necessarily independent or identical. We will depict a system as an undirected network comprising a set of nodes (vertices) S, and a set of edges E, where nodes are considered unreliable and edges are perfectly reliable (note that any setting with imperfect edge reliability can be easily transformed to a corresponding representation where they are perfectly reliable [1]). The system is considered to be functional as long as there is a path from left to right which passes only though functioning nodes, see Figure 1. This is usually represented mathematically by the structure function, $\phi : \{0,1\}^n \to \{0,1\}$, which maps component status to system status.

Herein, our focus is on an equivalent means of evaluation based on cut sets. A set of components, C, is said to be a cut set of the system if the system is failed whenever all the components in C are failed. A cut set is said to be a minimal cut set if no subset of it is also a cut set. Then, the set of all minimal cut sets, $\mathcal{C}$, characterises the operational state of a system completely and is equivalent to knowledge of the structure function [4] . In addition to the cut sets characterising the operational state of the system

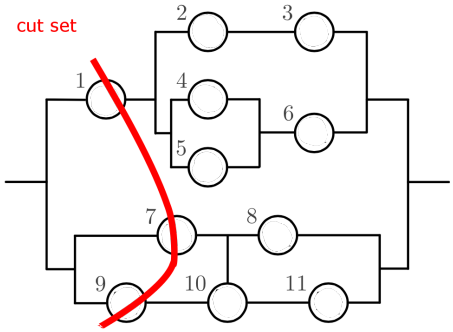

Figure 1: A sample network with a minimal cut set.

given the binary operational state of the components, they also immediately provide the system failure time if the individual component failure times are known [3]:

$$T_S = f_S(T_1, \ldots, T_n) := \min_{\mathrm{C} \in \mathcal{C}} \left\{ \max_{c \in \mathrm{C}} \{T_c\} \right\}.$$

Thus, the failure time for the system depends on the system structure and the failure time distributions for each node.

The traditional approach to estimating the expectation of a functional of the lifetime of a system given the lifetime distributions of the components is to perform a simple Monte Carlo simulation. That is,

$$\mathbb{E}[g(T_S)] = \frac{1}{N} \sum_{i=1}^{N} g(f_S(t_1^{(i)}, \ldots, t_n^{(i)})) \text{ where } t_j^{(i)} \sim F_j(\cdot).$$

The overall runtime for this approach depends on three quantities:

1. Variance of the estimator. Due to the random nature of component failure times, the estimator is a random variable: higher variance estimators will require more iterations to achieve an accurate estimate;

2. Target accuracy of the estimate. Naturally, the higher the desired accuracy, the longer the algorithm will take due to more iterations being required;

3. Number of cut sets. As the system size grows, the number of cut sets has a combinatorial growth, making the approach impractical for very large systems.

Less brute force approaches are possible with the restrictive assumption of iid components by making use of the system signature [13, 14]. More recent

work on the survival signature [5] has generalised the signature to multiple types of component, with the weaker assumption of exchangeability between components. However, if a large number of the components are of different types or there are highly dependent failures, then the survival signature will also grow exponentially in complexity. Hence, in this work, we first address the most general possible setting in which any form of component lifetime and dependence structure is allowed, requiring only knowledge of component lifetimes and the cut sets of the system. However, note that it should be possible to specialise this approach to work with the survival signature which we hope to address in future research.

3. Monte Carlo algorithms

To simplify presentation, hereinafter we only consider estimating expected failure time directly, rather than some functional of the failure time, though this is mostly without loss of generality (see the Section 4 for details). Therefore, assume that for a given system S, we want to estimate the expected failure time.

$$\mathbb{E}T_S = \mathbb{E}f_S(t_1^{(i)}, \ldots, t_n^{(i)}).$$

There are many approaches to simulation which may differ in terms of convergence to the true value as well as computational characteristics. In order to compare them, we present some useful cost and error expressions in the following subsection.

3.1. Performance measures: error and cost definitions

We start by defining the two main quantities, which will be used throughout this paper to compare methodologies. Given an estimator $\hat{T}_S$ of the quantity $\mathbb{E}T_S$, the Mean Squared Error (MSE) of any Monte Carlo based method is:

$$\text{error} = \mathbb{E}\left[\left((\hat{T}_S) - \mathbb{E}T_S\right)^2\right].$$

The classical decomposition of the MSE yields:

$$\begin{aligned}\mathbb{E}\left[\left(\hat{T}_S - \mathbb{E}T_S\right)^2\right] &= \mathbb{E}\left[\left(\hat{T}_S + \mathbb{E}\hat{T}_S - \mathbb{E}\hat{T}_S - \mathbb{E}T_S\right)^2\right] \\ &= \mathbb{E}\left[\left(\hat{T}_S - \mathbb{E}\hat{T}_S\right)^2\right] + \left(\mathbb{E}\hat{T}_S - \mathbb{E}T_S\right)^2 \qquad (1)\end{aligned}$$

where $\left(\mathbb{E}\hat{T}_S - \mathbb{E}T_S\right)^2$ is the squared bias error, while $\mathbb{E}\left[\left(\hat{T}_S - \mathbb{E}\hat{T}_S\right)^2\right]$ is the error due to Monte Carlo variance. The first is a systematic error arising

from the fact that we might not sample our random variable exactly, but rather use a suitable approximation, while the second error comes from the randomised nature of any Monte Carlo algorithm. For example, in traditional Monte Carlo applications, one samples exactly so that the first error is zero and only the Monte Carlo variance needs to be treated carefully.

The cost of any Monte Carlo based algorithm is typically taken to be the expected runtime in order to achieve a prescribed accuracy. A more convenient approach for theoretical comparison between different methods is to define

$$\text{cost} = \mathbb{E}(\#\text{random number generations and operations}).$$

We now recap traditional Monte Carlo and then introduce Multilevel Monte Carlo, in each case highlighting what results for these two measures of performance.

3.2. Traditional (or single-level) Monte Carlo algorithm

The traditional Monte Carlo estimator is based on N replications of simulating the system lifetime, via the minimal cutsets, by simulating the component lifetimes. That is, given system simulations $\tau_i = f_S(t_1^{(i)}, \ldots, t_n^{(i)})$ the traditional Monte Carlo estimator has the form

$$\hat{T}_S = \frac{1}{N} \sum_{i=1}^{N} f_S(t_1^{(i)}, \ldots, t_n^{(i)}). \tag{2}$$

For reasons that will become clear in the sequel, it is useful to refer to this as the single-level Monte Carlo algorithm because it emphasises the relationship to Multilevel Monte Carlo.

This single-level Monte Carlo estimate has variance proportional to N^{-1},

$$\text{Var}(\hat{T}_S) = \text{Var}\left(\frac{1}{N}\sum_{i=1}^{N} \tau_i\right) = \frac{1}{N^2}\,\text{Var}\left(\sum_{i=1}^{N} \tau_i\right) = \frac{1}{N}\,\text{Var}\left(\tau_i\right).$$

The estimator (2) is clearly unbiased, because there is no approximation involved in estimating the failure time, so the error measure introduced earlier only has this second variance term,

$$\text{error}_{\text{MC}} = N^{-1}\,\text{Var}\left(\tau_i\right).$$

Indeed, more generally the well known central limit result for standard Monte Carlo means that:

$$\mathbb{P}\left(|\hat{T}_S - \mathbb{E}T_S| > z\frac{\sqrt{\mathrm{Var}(\tau_i)}}{\sqrt{n}}\right) \approx \mathbb{P}(|Z| > z)$$

for $Z \sim \mathrm{N}(0,1)$.

Thus, for a desired level of accuracy $\varepsilon > 0$ with confidence level $1-\alpha$, we require

$$n = z_{\alpha/2}^2 \,\mathrm{Var}(\tau_i)\varepsilon^{-2}$$

Monte Carlo simulations, where the quantile $z_{\alpha/2}$ is chosen to ensure, that $\mathbb{P}(Z > z_{\alpha/2}) = \alpha/2$.

Naturally $z_{\alpha/2}$ is a constant for any fixed level of confidence, so the variable compute costs in simulation are

$$\mathrm{cost}_{\mathrm{MC}} = \mathrm{Var}(\tau_i) \cdot \varepsilon^{-2} \cdot \#\mathcal{C}, \tag{3}$$

where $\#\mathcal{C}$ denotes the number of minimal cut sets for the system.

4. Multilevel Monte Carlo

To simplify presentation we only consider estimating expected failure time directly, rather than some functional of the failure time. Note that there is no loss of generality, so long as the functional of interest is Lipschitz continuous (or bounded for discrete measures). The most common functional of interest that this would exclude is computing expectations of quantiles. However, this problem can be treated with the smoothing approach, discussed for the MLMC setting in [9]. In all other cases, the presentation below carries over in the natural fashion.

4.1. General MLMC

We first introduce MLMC in generality before specialising this to the reliability problem considered herein. Consider a sequence of estimators $T_0, T_1, \ldots$, which approximates T_L with increasing accuracy, but also increasing cost. By linearity of expectation, we have

$$\mathbb{E}(T_L) = \mathbb{E}(T_0) + \sum_{\ell=1}^{L} \mathbb{E}[T_\ell - T_{\ell-1}],$$

and therefore we can use the following unbiased estimator for $\mathbb{E}[T_L]$,

$$\frac{1}{N_0}\sum_{n=1}^{N_0} T_0^{(0,n)} \;+\; \sum_{\ell=1}^{L}\left\{\frac{1}{N_\ell}\sum_{n=1}^{N_\ell}\left(T_\ell^{(\ell,n)} - T_{\ell-1}^{(\ell,n)}\right)\right\}$$

The inclusion of the level ℓ in the superscript (ℓ, n) indicates that the samples used at each level of correction are independent, but crucially note that the differences themselves use common samples. Note the terminology 'correction' arises from the fact that each T_ℓ is generally *not* an unbiased estimate any more.

Let V_0 and cost_0 be the variance and the expected cost of one sample of T_0, and let V_ℓ, cost_ℓ be the variance and expected cost of one sample of $T_\ell - T_{\ell-1}$. Then the overall expected cost and variance of the multilevel estimator are $\sum_{\ell=0}^{L} N_\ell \cdot \text{cost}_\ell$ and $\sum_{\ell=0}^{L} N_\ell^{-1} \cdot V_\ell$, respectively.

More generally, the idea is that provided the product $V_\ell \cdot C_\ell$ decreases with ℓ (i.e. the cost increases with level slower than the variance decreases), then one can achieve significant computational savings, which can be formalised in Theorem 1 from [6].

Theorem 1. *Let T_S denote a random variable, and let T_ℓ denote the corresponding level ℓ numerical approximation.*

If there exist independent estimators Y_ℓ based on N_ℓ Monte Carlo samples, and positive constants $\alpha, \beta, \gamma, c_1, c_2, c_3$ such that $\alpha \geq \frac{1}{2}\min(\beta, \gamma)$ and

1. $|\mathbb{E}(T_\ell - T_S)| \leq c_1 2^{-\alpha \ell}$
2. $\mathbb{E}(Y_\ell) = \begin{cases} \mathbb{E}(T_0), & \ell = 0 \\ \mathbb{E}(T_\ell - T_{\ell-1}), & \ell > 0 \end{cases}$
3. $\text{Var}(Y_\ell) \leq c_2 N_\ell^{-1} 2^{-\beta \ell}$
4. $\text{cost}_\ell \leq c_3 2^{\gamma \ell}$, *where cost_ℓ is the expected computational complexity of Y_ℓ*

then there exists a positive constant c_4 such that for any $\varepsilon < e^{-1}$ there are values L and N_ℓ for which the multilevel estimator

$$Y = \sum_{\ell=0}^{L} Y_\ell,$$

has a mean-square-error with bound

$$MSE \equiv \mathbb{E}\left[(Y - \mathbb{E}[T_S])^2\right] < \varepsilon^2$$

with a computational complexity C with bound

$$\text{cost}_{MLMC} \leq \begin{cases} c_4 \varepsilon^{-2}, & \beta > \gamma, \\ c_4 \varepsilon^{-2} (\log \varepsilon)^2, & \beta = \gamma, \\ c_4 \varepsilon^{-2-(\gamma-\beta)/\alpha}, & \beta < \gamma. \end{cases}$$

Remark 2. We will informally illustrate the idea behind MLMC on a simple example with just two levels. Consider just two approximations T_k and T_L, where $k < L$, with sample costs $\text{cost}_k < \text{cost}_L$. It is clear, that the cost of one sample for $T_L - T_k$ is roughly cost_L. Now assume, that

$$V_1 = \operatorname{Var} T_k \approx \operatorname{Var} T_L, \text{ and } V_2 = \operatorname{Var}(T_L - T_k),$$

where $V_2 < V_1$. Then we have

$$\begin{aligned} &\mathbb{E}T_L = \mathbb{E}T_k + \mathbb{E}(T_L - T_k) \\ \Rightarrow \hat{T} &= \frac{1}{N}\sum_{n=1}^{N} T_L^{(2,n)} \\ &\approx \bar{T} = \frac{1}{N_1}\sum_{i=1}^{N_1} T_k^{(1,i)} + \frac{1}{N_2}\sum_{j=1}^{N_2}(T_L^{(1,i)} - T_k^{(1,i)}). \end{aligned}$$

We see that the overall cost of Monte Carlo estimators, according to (3), can be expressed as

$$\begin{aligned} \text{cost}(\hat{T}) &= \varepsilon^{-2} \cdot V_1 \cdot \text{cost}_L \\ \text{cost}(\bar{T}) &= \varepsilon^{-2} \cdot (\text{cost}_k \cdot V_1 + \text{cost}_L \cdot V_2)\,, \end{aligned}$$

which gives us a condition

$$\text{cost}(\hat{T}) > \text{cost}(\bar{T}) \Rightarrow 1 > \frac{\text{cost}_k}{\text{cost}_L} + \frac{V_2}{V_1}.$$

In other words, provided there exists a good coupling between estimators T_k and T_L, we have reductions cost even for two levels. The two-level Monte Carlo method in the context of Monte Carlo path simulation has been suggested and analysed in [11].

Remark 3. Multilevel Monte Carlo became popular after the seminal work of Mike Giles [6] for estimating expectations of functionals $\mathbb{E}(f(X_t))$, where X_t is the solution of a stochastic differential equation. In the general Multilevel Monte Carlo path simulation setting, T_ℓ from Theorem 1 is the functional value, evaluated via an approximation arising from a discretisation method, e.g. the Euler-Maruyama method [12].

4.2. MLMC for system reliability

Theorem 1 suggests that one may want to try getting a coarser Monte Carlo estimate of the system lifetime, perhaps by considering only a subset of the collection of minimal cutsets.

$$\mathcal{C}' \subset \mathcal{C} \implies \min_{C\in\mathcal{C}'} \Big\{ \max_{i\in C}\{t_i\} \Big\} = T_S' \geq T_S = \min_{C\in\mathcal{C}} \Big\{ \max_{i\in C}\{t_i\} \Big\}.$$

On its own T'_S is a biased estimator, so although a traditional single-level Monte Carlo estimator based on it may have lower computational cost, it will have increased MSE because the first error term in (1) can no longer be ignored. However, by using this coarse estimate inside an MLMC approach, we aim improve the overall performance. To this end we introduce the sequence of estimators $T_0, \ldots, T_L$ based on a nested sequence of minimal cutsets, $\mathcal{C}_0 \subset \cdots \subset \mathcal{C}_L = \mathcal{C}$. Note that here $T_L \equiv T_S$, which is not typically true in a general MLMC setting.

The crucial ingredient is the finite telescopic sum

$$\mathbb{E}(T_S) \equiv \mathbb{E}(T_0) + \sum_{\ell=1}^{L} \mathbb{E}(T_\ell - T_{\ell-1}) = \sum_{\ell=0}^{L} \mathbb{E}(Y_\ell)$$

As described above, we independently estimate each term, and within each term, T_ℓ and $T_{\ell-1}$ use the same random component simulations:

$$\mathbb{E}(Y_\ell) \approx N_\ell^{-1} \sum_{j=1}^{N_\ell} \left(\tau_\ell^{(j)} - \tau_{\ell-1}^{(j)} \right)$$

with each level having cost being bounded from above by $c \cdot \mathrm{Var}(Y_\ell) \cdot \#\mathcal{C}_\ell$. Here c is a constant independent of ℓ and the desired target accuracy. We choose $\#\mathcal{C}_\ell$ — the number of minimal cutsets at level ℓ — to be a proxy for the upper bound on the cost of each level, because for a fixed system the number of elements in each minimal cutset is independent of the target accuracy. In other words, as we double the number of minimal cutsets in each level, their number is a straightforward way to construct a meaningful and easy upper bound for the cost of one sample.

Thus, the overall MLMC variance is

$$\mathrm{Var}\left(\sum_{\ell=0}^{L} Y_\ell \right) = \sum_{\ell=0}^{L} N_\ell^{-1} \mathrm{Var}(Y_\ell)$$

at a cost of $\sum_{\ell=0}^{L} N_\ell \cdot \#\mathcal{C}_\ell$. Therefore, given a fixed target accuracy (variance), if we choose a sample size $N_\ell \approx \sqrt{\mathrm{Var}(Y_\ell)/\#\mathcal{C}_l}$ on each level, this will minimise the computational cost. That is, for a desired accuracy $\varepsilon > 0$, the overall cost is:

$$\mathrm{Cost}_{\mathrm{MLMC}} = \sum_{\ell=0}^{L} N_\ell \cdot \#\mathcal{C}_\ell = \varepsilon^{-2} \left(\sum_{\ell=0}^{L} \sqrt{\mathrm{Var}(Y_\ell) \cdot \#\mathcal{C}_\ell} \right)^2$$

Recall that,

$$\mathrm{cost}_{\mathrm{MC}} = \mathrm{Var}(\tau_i) \cdot \varepsilon^{-2} \cdot \#\mathcal{C},$$

This means that:

1. If we have a good coupling between the approximations, or equivalently $\mathrm{Var}(Y_\ell)$ decays rapidly, then we can achieve considerable savings compared to the single-level Monte Carlo algorithm.

2. Additional savings are possible if we do not calculate all the levels Y_ℓ, but rather stop the algorithm early. This introduces a (small) bias, but substantially decreases the overall computational cost. As long as the bias is quantified — and when combined with the estimator variance gives a MSE (1) below our target accuracy — then we can still solve the original problem at much lower cost.

   Our proposed application to reliability involves a nested sequence of minimal cutsets providing improving accuracy:

   $$\mathcal{C}_0 \subset \cdots \subset \mathcal{C}_L = \mathcal{C},$$

   so the possible gain from stopping early depends on the way the minimal cutsets are chosen at each level.

Level selection algorithm

The first point to note is that existing Multilevel Monte Carlo literature has shown that anything less than geometric decay in the cost of computation at each level leads to suboptimal gains, see [8]. Therefore, we pre-specify that level ℓ contain $\lceil \#\mathcal{C}/2^{L-\ell} \rceil$ minimal cutsets. The levels will be grown from $\ell = 0$ up, adding in those minimal cutsets which are in some sense optimal for the next level. Thus, we specify level selection in an inductive fashion.

$\ell = 0$

Level 0 will be simulated most frequently since it is the lowest cost. Therefore, optimal choice of this level is straightforward: it should contain the minimal cutsets which provide the best approximation to T_S. That is, we wish to assign to level 0 the minimal cutsets which have smallest expected failure time, since these will most frequently be the causes of system failure. To achieve this, we propose an initial highly crude estimate by performing a pilot standard Monte Carlo simulation of N' lifetimes of each component in the system, using these to generate N' realisations of the failure time associated with each minimal cutset,

$$\eta_i = \frac{1}{N'} \sum_{j=1}^{N'} \max_{c \in C_i} \left\{ t_c^{(j)} \right\}, \quad \forall\, C_i \in \mathcal{C}.$$

The cutsets corresponding to the smallest $\lceil \#\mathcal{C}/2^{L} \rceil$ of these η_i are then chosen to form $\mathcal{C}_0$.

$\ell > 0$

Given that we have chosen levels $0, \ldots, \ell - 1$ already (that is $\mathcal{C}_0 \subset \cdots \subset \mathcal{C}_{\ell-1}$ are now fixed), we need to select which cutsets to add from $\mathcal{C}_{\text{trial}} = \mathcal{C} \backslash \mathcal{C}_{\ell-1}$. To maximise the performance of MLMC we would like to select the cutsets such that

$$\mathbb{E}[T_{\ell-1} - T_\ell] \to \max .$$

In other words, choose

$$\mathcal{C}_\ell = \underset{\substack{\mathcal{C}_\ell \subseteq \mathcal{C} \\ \text{s.t. } \mathcal{C}_{\ell-1} \subset \mathcal{C}_\ell, \#\mathcal{C}_\ell = \lceil \#\mathcal{C}/2^{L-\ell} \rceil}}{\arg\max} \ \mathbb{E}[T_{\ell-1} - T_\ell],$$

so that the contribution from each level is as large as possible in the smallest levels, leading to a rapid decay in the size of the contribution in each level and hence the possibility This is the of terminating the algorithm early. In particular, note that if $\sum_{\ell=k}^{L} \mathbb{E}(T_\ell - T_{\ell-1}) \ll \varepsilon$, then levels $k, \ldots, L$ need not be simulated at all, so that ensuring all large differences occur in early levels is highly desirable.

Notice that:

$$\mathbb{E}[T_{\ell-1} - T_\ell] \leq \mathbb{E}\left[T_{\ell-1} - \min\left\{T_{\ell-1}, \max_{c \in C}\{t_c\}\right\}\right],$$

for any $C \in \mathcal{C}_{\text{trial}}$. So our choice for sorting cut sets is motivated by the minimisation of the upper bound for the increments at each level, which we can implement for any level ℓ in a simple way:

- use the N' samples and calculated failure times for all cut sets used in selecting $\ell = 0$,
- calculate the following estimates:

$$\delta_k = \frac{1}{N'} \sum_{j=1}^{N'} \left[T_{\ell-1} - \min\left\{T_{\ell-1}, \max_{c \in C_k}\{t_c^{(j)}\}\right\}\right],$$

  for each $C_k \in \mathcal{C}_{\text{trial}}$.
- Sort δ_k in a descending order and add cutsets corresponding to the largest values for δ_k to $\mathcal{C}_\ell$ until $\#\mathcal{C}_\ell = \lceil \#\mathcal{C}/2^{L-\ell} \rceil$.

This choice of number of minimal cut sets on each level guarantees the exponential increase in the cost with $\gamma = 1$ in Theorem 1.

4.3. Full Multilevel Monte Carlo algorithm for reliability

One of the key features of the Multilevel Monte Carlo algorithm is its ability to naturally provide stopping criteria for an optimal selection of the number of levels $\hat{L}$, which we illustrate now along with the full description of the algorithm. For more advanced approaches to implementation of Multilevel Monte Carlo we refer to [7] and [8].

According to the Theorem 1 and the first assumption in it, we have asymptotically as $\ell \to \infty$

$$\mathbb{E}(Y_\ell) \approx \mathbb{E}(T_S - \hat{T}_\ell),$$

so that a natural stopping criteria is to choose $\hat{L}$ minimal such that $|Y_{\hat{L}}| \leq \varepsilon/2$.

Input: Requested accuracy ε and level specification as per §4.2.

1. Set the initial number of levels to $\hat{L} := 2$. In order to define the optimal number of samples at each level we need to estimate the variances on levels $\ell = 0, 1, 2$

2. Compute $N_\ell := 100$ samples on levels $\ell = 0, 1, 2$

3. Estimate $\mathrm{Var}(Y_\ell)$ and update N_ℓ for each level $\ell = 0, \ldots, \hat{L}$:

$$N_\ell := \max\left\{N_\ell, \hat{N}_\ell\right\}, \text{ where}$$

$$\hat{N}_\ell = \left\lceil 4 \cdot \varepsilon^{-2} \cdot \sqrt{\mathrm{Var}(Y_\ell)2^{-\ell}} \cdot \sum_{k=0}^{\hat{L}} \sqrt{\mathrm{Var}(Y_k) \cdot 2^k} \right\rceil$$

   We take the maximum as it may happen that the numerical variance was initially overestimated and so more simulations were performed than necessary. If N_ℓ has increased less than 1% on all levels, then skip to step 5.

4. Compute the additional number of samples on each level $\ell = 0, \ldots, \hat{L}$ and return to step 3.

5. Upon reaching this step we have converged in terms of MSE due to variance, so we test whether the bias error term is sufficiently small to terminate or whether more levels and simulations are required. If $|Y_\ell| \geq \varepsilon$:
   Then: set $\hat{L} := \hat{L} + 1$, $\mathrm{Var}(\hat{Y}_{\hat{L}}) = \mathrm{Var}(Y_{\hat{L}-1})/2$ and return to step 3;
   Else: Terminate algorithm returning $\sum_{\ell=0}^{\hat{L}} Y_\ell$ as the estimate.

There are two extreme cases to bear in mind. In the first case, we have only a few minimal cut sets (or even only one), which influence the failure time. This case is treated with the initial choice of the cut sets and selection as prescribed in §4.2 should ensure the minimal number of levels is simulated. The second case is when all the cut sets have similar 'weight' in determining the failure time, such as with a fully connected system with independent and identically distributed failure times for all components. This case is treated with the doubling of the number of cut sets with respect to the previous level, which again assures the mean and variance decay between the levels.

5. Numerical experiments

5.1. Systems and component reliability distributions

We generated many random systems to test the MLMC reliability method proposed hereinbefore. These random systems are generated by starting from the trivial one component system and with fixed probabilities either:

- replacing a component with two components in series;
- replacing a component with two components in parallel;
- selecting two edges and inserting a 'bridging' component.

This allowed us to generate a wide range of different systems and in particular an increasing sequence of related systems with varying numbers of component.

For all systems we consider three test cases, where the components have Weibull distributed lifetimes with shape parameter $k = 0.5, 1$ or 3 and where the scale is chosen uniformly at random on an interval $[2, 10]$. Variety in shape parameters corresponds to different applications in industry (see e.g. [15]). The shape parameter has a substantial effect on the corresponding density function.

5.2. Numerical results

We ran our algorithm 100 times for systems of different sizes with independent but differently distributed components, whose reliability is described above. In each case we considered fixed target accuracies of $\varepsilon = 2^{-4}$ and $\varepsilon = 2^{-7}$, and computed the cost gains achieved for these fixed accuracies.

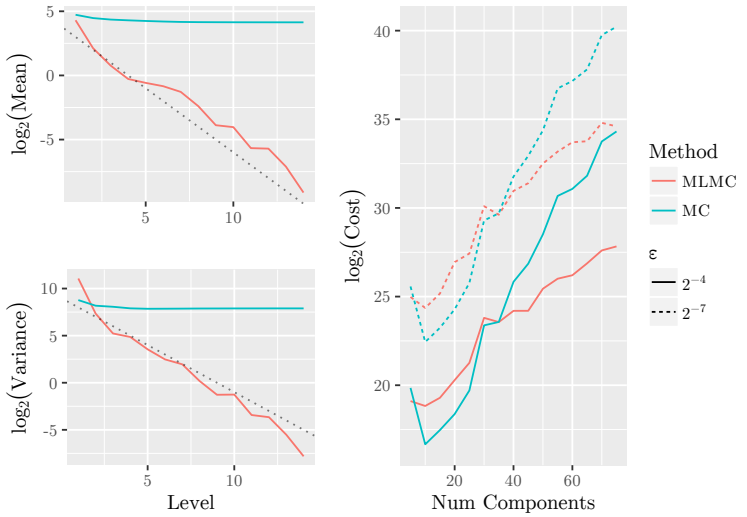

Figure 2: Left: Diagnostic tests for the largest considered system; Right: cost gains for nested randomly grown systems from 5 to 75 components, with Weibull distribution having shape parameter $k = 0.5$ and uniformly distributed scale.

5.2.1. Shape parameter $k = 0.5$.

The left top and bottom plots on Figure 2 show the result of diagnostic runs, where we tested the variance and mean decay, which correspond to assumptions (3) and (1) from Theorem 1 with $\beta = 1$ and $\alpha = 1$ respectively. This indicates, that Multilevel Monte Carlo achieves the same convergence rate as traditional Monte Carlo in terms of accuracy ε, but can offer computational savings, due to the fact that most of the samples are calculated for very small subset of minimal cut sets. The right plot compares the differences in averaged costs for Multilevel Monte Carlo and standard Monte Carlo algorithms, which shows good savings even including the costs for initially selecting the cutsets for each level.

5.2.2. Shape parameter $k = 1$.

The test case with $k = 1$ gives us almost the same mean and variance decays along with computational gains as in the case with $k = 0.5$. One can see that in the last levels we see even super linear decay for the mean and variance, which indicates that the cut sets being added at those levels have very weak impact on system lifetime compared to those already chosen, which indicates good performance for the level selection algorithm.

5.2.3. Shape parameter $k = 3$.

The case with $k = 3$ shows substantial savings for $\varepsilon = 2^{-7}$, as also seen in the previous examples, while still showing competitive results for $\varepsilon = 2^{-4}$.

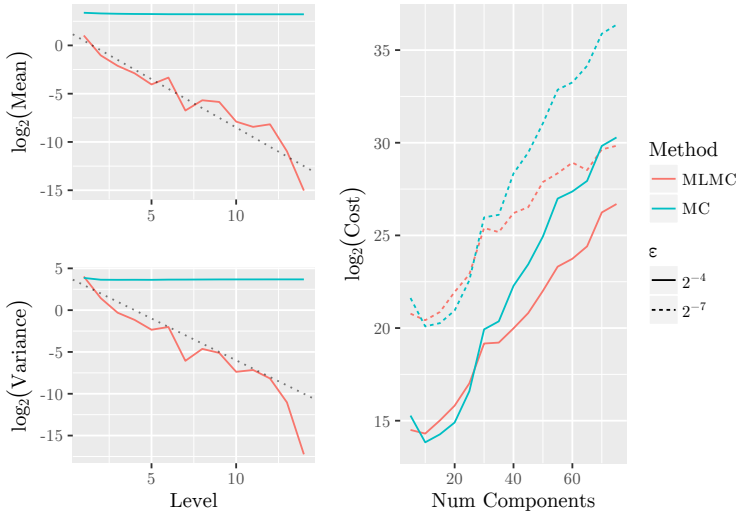

Figure 3: Left: Diagnostic tests for the largest considered system; Right: cost gains for nested randomly grown systems from 5 to 75 components, with Weibull distribution having shape parameter $k = 1$ and uniformly distributed scale.

The reason the gains are more modest here is that, as we had before, there is very small variance in the standard Monte Carlo and overall Multilevel Monte Carlo estimators, which puts more emphasis on the initial level selection costs which are not ε dependent, but are size dependent.

6. Conclusion

We have presented an exciting new application for the Multilevel Monte Carlo method, which speeds up traditional Monte Carlo estimation of system lifetimes and provides a approach which can easily generalise to other reliability problems which involve cut (path) sets. Unlike classic MLMC implementations, where one considers different approximations of a certain stochastic process wherein all of them are biased, here we introduce approximations based on sorting the minimal cut sets in a special way, which are biased, but less costly to simulate. The numerical experiments show substantial savings for large systems and are promising for further study of reliability and structural optimisation.

Acknowledgments

Louis Aslett is supported by the i-like programme grant (EPSRC grant reference number EP/K014463/1 http://www.i-like.org.uk). Tigran Nagapetyan and Sebastian Vollmer received funding from EPSRC Grant EP/N000188/1.

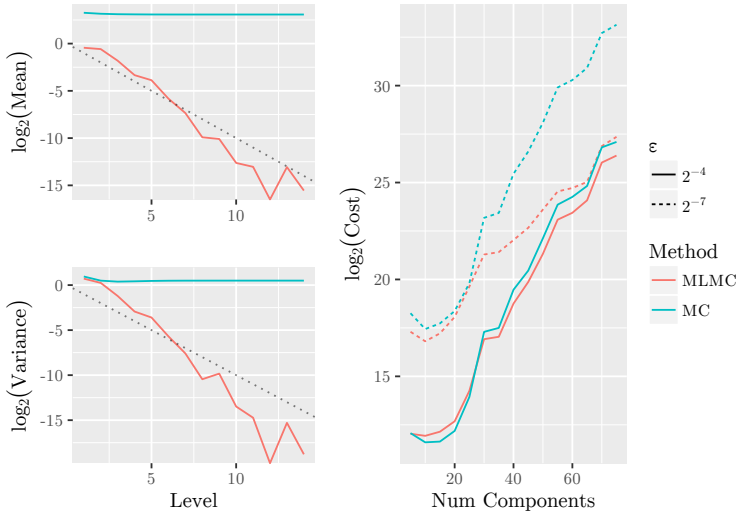

Figure 4: Left: Diagnostic tests for the largest considered system; Right: cost gains for nested randomly grown systems from 5 to 75 components, with Weibull distribution having shape parameter $k = 3$ and uniformly distributed scale.

[1] L. J. M. Aslett, F. P. A. Coolen, and S. P. Wilson. Bayesian inference for reliability of systems and networks using the survival signature. *Risk Analysis*, 35(9):1640–1651, 2015.

[2] Richard E. Barlow and Frank Proschan. *Mathematical Theory of Reliability*. Wiley, 1965.

[3] Richard E. Barlow and Frank Proschan. *Statistical Theory of Reliability and Life Testing*. To Begin With Press, 1981.

[4] Richard W. Butterworth. A set theoretic treatment of coherent systems. *SIAM Journal on Applied Mathematics*, 22(4):590–598, 1972.

[5] F. P. A. Coolen and T. Coolen-Maturi. Generalizing the signature to systems with multiple types of components. In W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, editors, *Complex Systems and Dependability*, volume 170 of *Advances in Intelligent and Soft Computing*, pages 115–130. Springer Berlin Heidelberg, 2012.

[6] M.B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[7] M.B. Giles. Matlab code for multilevel Monte Carlo computations. `http://people.maths.ox.ac.uk/gilesm/acta/`, 2014.

[8] M.B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

[9] M.B. Giles, T. Nagapetyan, and K. Ritter. Multilevel Monte Carlo approximation of distribution functions and densities. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):267–295, 2015.

[10] S. Heinrich. Monte Carlo complexity of global solution of integral equations. *Journal of Complexity*, 14(2):151–175, 1998.

[11] A. Kebaier. Statistical Romberg extrapolation: a new variance reduction method and applications to options pricing. *Annals of Applied Probability*, 14(4):2681–2705, 2005.

[12] Peter E Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer, 1992.

[13] Subhash Kochar, Hari Mukerjee, and Francisco J. Samaniego. The "signature" of a coherent system and its application to comparisons among systems. *Naval Research Logistics*, 46(5):507–523, 1999.

[14] Francisco J. Samaniego. On closure of the IFR class under formation of coherent systems. *IEEE Transactions on Reliability*, R-34(1):69–72, April 1985.

[15] Martin L Shooman. *Reliability of computer systems and networks: fault tolerance, analysis, and design*. John Wiley & Sons, 2003.