

Chapter 1

Analysis and Improvements of Path-Based Methods for Monte Carlo Reliability Evaluation of Static Models *

Héctor Cancela, Pierre L'Ecuyer, Matías Lee, Gerardo Rubino and Bruno Tuffin

Abstract Many dependability analyses are performed using static models, that is, models where time is not an explicit variable. In these models, the system and its components are considered at a fixed point in time, and the word “static” means that the past or future behavior is not relevant for the analysis. Examples of such models are reliability diagrams, or fault trees. The main difficulty when evaluating the dependability of these systems is the combinatorial explosion associated with exact solution techniques. For large and complex models, one may turn to Monte Carlo methods, but these methods have to be modified or adapted in the presence of rare important events, which are commonplace in reliability and dependability systems. This chapter examines a recently proposed method designed to deal with the problem of estimating reliability metrics for highly-dependable systems where the failure of the whole system is a rare event. We focus on the robustness properties of estimators. We also propose improvements to the original technique, including its combination with randomized quasi-Monte Carlo, for which we prove that the variance converges at a faster rate (asymptotically) than for standard Monte Carlo.

H. Cancela
U. of the Republic, Uruguay, e-mail: hcancela@fing.edu.uy

P. L'Ecuyer
U. of Montréal, Canada, e-mail: lecuyer@iro.umontreal.ca

M. Lee
Nac. U. of Córdoba, Argentina, e-mail: lee.matias@gmail.com

G. Rubino and B. Tuffin
INRIA, France, e-mail: rubino@irisa.fr, btuffin@irisa.fr

* Partially supported by the STIC AmSud 2009 Project “Performance evaluation and design of optical and wireless networks”.

1.1 Introduction

Dependability analysis of complex systems is sometimes performed using dynamic stochastic models. The system is represented by some type of stochastic process such as a Markov or a semi-Markov one, and different dependability metrics (reliability, point availability, interval availability, etc.) are evaluated as functions of the process at a fixed point in time (e.g., reliability or point availability), over a finite interval (e.g. interval availability), or in equilibrium (e.g. asymptotic availability). But in many cases, the system is considered in a context where the time variable plays no specific role. These models are called *static*, and are widely used in engineering, taking the form of specific mathematical objects such as reliability networks, reliability diagrams, fault trees, etc.

The basic scheme is the following. The system is composed of M components which typically are subsystems of the original one, and are considered as atoms in the modeling effort. Each component and the whole system can be in two different states, either operational or failed. The set of states of the M components is a *configuration* or *state-vector* of the system (hence, there are at most 2^M such configurations, since not all configurations are necessarily possible in the model of a specific system). We assume that the probability of each configuration is known. The main system dependability metric is the *reliability* R of the system, the probability that the whole system is operational, or equivalently, its *unreliability* $U = 1 - R$, the probability that the whole system fails. The reliability is the sum of the probabilities of all the configurations leading to an operational state for the whole system, and the unreliability is the corresponding sum of the probabilities of all the configurations leading to a failed system. In such a static context, R is sometimes also called the availability of the system. The function Φ mapping the configurations into one of the two possible system states is called the structure function of the system. It provides the information about the way the M components are organized from the dependability point of view, that is, the way the combination of operational and failed components lead to an operational or failed system. The different modeling frameworks (reliability networks, fault trees...) can be seen as different languages that allow for a compact representation of structure functions.

Suppose that the components behave independently, and that for each component we know the probability that it is in the operational state. We number the components from 1 to M , and r_i is the probability that component i is working. Coding by 1 the operational state (of a component, or of the whole system) and by 0 the failed state, we have that

$$R = \sum_{x:\Phi(x)=1} p(x),$$

where x denotes a configuration and $p(x)$ its probability, $x = (x_1, \dots, x_M)$, and x_i is the state of component i in configuration x . The independence assumption on the states of the components means that for any configuration x we have

$$p(x) = \prod_{i:x_i=1} r_i \prod_{j:x_j=0} (1 - r_j).$$

We are interested in the case where $R \approx 1$, or equivalently, $U \approx 0$, the usual situation in many areas, typically in the analysis of critical systems. These are systems where a failure may produce losses in human lives (transporting facilities, nuclear plants...) or huge losses in monetary terms (information systems, telecommunication networks...), so that system design is extremely conservative ensuring very low failure probability. This is a rare event context, the rare event being the system failure. If X is a random configuration (randomness coming from the fact that the component state is assumed to be a random variable), then, the rare event is “ $\Phi(X) = 0$ ”, and $\mathbb{P}(\Phi(X) = 0) = 1 - R = U$. Since these are binary random variables, $R = \mathbb{E}(\Phi(X))$ and $U = \mathbb{E}(1 - \Phi(X))$.

In this chapter, we address the problem of estimating U (or R) using Monte Carlo, where the structure function is given by means of a graph. Think of a communication network represented by an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, also referred to as links in this context. The graph is supposed to be connected and without loops. The components are, for instance, the edges, and recall they are assumed to operate independently. Associated with edge i we have its (elementary) reliability r_i (or equivalently, its unreliability u_i); if X_i is the binary random variable “state of component i ”, we have $r_i = \mathbb{P}(X_i = 1)$, and $u_i = \mathbb{P}(X_i = 0)$, $r_i + u_i = 1$. A configuration is a vector $x = (x_1, \dots, x_M)$, where x_i is the state of component (here, edge) i . We denote by $E(x)$ the subset of operational edges in configuration x , that is, $E(x) = \{i \in E : x_i = 1\}$, and by $G(x)$ the graph $G(x) = (V, E(x))$. It remains to specify when the whole system works, that is, to define the structure function. For this purpose, two nodes are selected in V , denoted by the letters s (as source) and t (as terminal). The system (the network) works under the configuration x if nodes s and t belong to the same connected component of the random graph $G(x)$. That is, $\Phi(x) = 1$ iff in $G(x)$, nodes s and t are connected. This model is the basic one in the network reliability area, and it corresponds to the typical model in reliability block diagrams. Computing $R = \mathbb{P}(\Phi(X) = 1)$ is an NP-complete problem, even in very restricted classes of graphs. More specifically, the use of the many combinatorial approaches for computing R or U cannot deal with models of moderate size (around, say, one hundred components) and simulation is the only available evaluation tool.

For general presentations about the computation of the reliability or the unreliability in these static contexts, or about bounding them, as well as complexity issues, see [1], [2], [3], [4]. In these references the reader can also find material about Monte Carlo estimation of these basic dependability metrics.

1.2 Standard Monte Carlo reliability evaluation

The standard estimation procedure for U (or R) simply consists in building a sequence $X^{(1)}, X^{(2)}, \dots, X^{(n)}$ of independent copies of the random configuration X , and checking in each graph of the corresponding sequence $G(X^{(1)}), \dots, G(X^{(n)})$ if s and t are connected. The ratio between the number of times s and t are not connected

and n , is then an unbiased estimator \widehat{U} of U . Formally,

$$\widehat{U} = \frac{1}{n} \sum_{i=1}^n 1(\Phi(X^{(i)}) = 0),$$

where $1(A)$ is the indicator function of event A . The variance of \widehat{U} being $\mathbb{V}(\widehat{U}) = \sigma_n^2 = U(1-U)/n$, a confidence interval for U , with level $\alpha \in [0, 1]$, is obtained from the central limit theorem (CLT):

$$U \in \left(\widehat{U} - z_{1-\alpha/2} \sqrt{U(1-U)/n}, \widehat{U} + z_{1-\alpha/2} \sqrt{U(1-U)/n} \right)$$

with probability $1 - \alpha$, where $z_{1-\alpha/2}$ is the $1 - \alpha/2$ quantile of the normal law with mean 0 and variance 1. In many interesting and important systems, the reliability of the components is close to one and the path redundancy in the graph makes that the probability of the existence of at least a path between the two selected nodes is extremely high. Both factors make that unreliability of the whole network is very small. This precludes the use of the standard estimation approach, since we have to wait for a long time (on average) before observing a system failure. In other words, the cost in time of the standard procedure is very high.

To formalize this situation, assume that the unreliability of link i is $u_i = a_i \varepsilon^{b_i}$ with $a_i, b_i > 0$ and $0 < \varepsilon \ll 1$. Recall that a cut in the graph (with respect to nodes s and t) is a set of edges such that if we delete them from the graph, s and t become unconnected. A *mincut* is a cut that does not contain strictly another cut. Nodes s and t are unconnected if and only if for at least one (min)cut in the graph, all the edges that compose it are down. If γ is a mincut, we can denote by C_γ the event “all the edges in γ are down”, and write

$$U = \mathbb{P} \left(\bigcup_{\text{all mincut } \gamma} C_\gamma \right).$$

Observing that, due to the independence of the components' states, $\mathbb{P}(C_\gamma) = \prod_{i \in \gamma} u_i$, we see that U is a polynomial in ε and that $U = \Theta(\varepsilon^c)$ for some $c > 0$ (recall that the graph is connected, so, there is at least one cut separating nodes s and t).

The real number ε is a way to parameterize rarity: as ε goes to zero, the system failure event becomes increasingly rare. The relative error [9, 11] when estimating U using \widehat{U} , defined as the ratio between the square root of the variance of the estimator and its mean, i.e. $\sqrt{U(1-U)/n}/U$ (also called *relative variance*, or *coefficient of variation*) is $\approx (nU)^{-1/2}$ when ε is small, and increases as ε decreases. We want this relative error to be small, but not at any price! This means that we would like to avoid using an important computing effort in order to obtain specific error levels. That is, the CPU time required to compute the estimator from a sample of size n must also be taken into account. For this purpose, we consider the work-normalized relative variance (WNRV) of the estimator \widehat{U} , defined by

$$\text{WNRV}(\hat{U}) = \frac{\sigma_n^2 \tau_n}{U^2},$$

where τ_n is the mean time needed to compute \hat{U} using a sample of size n . Here, this time is essentially linear in n . What we want now is that this ratio remains bounded when $\varepsilon \rightarrow 0$. In other words, no matter how rare the system failure is, we would like to be able to estimate it accurately with “reasonable” effort. This property is called bounded WNRV (BWNRV), and it does not hold for \hat{U} , because $\text{WNRV}(\hat{U})$ is proportional to $1/U$, and $1/U \rightarrow \infty$ when $\varepsilon \rightarrow 0$.

In this work we discuss efficient Monte Carlo methods for the estimation of the unreliability U of the network, by combining two approaches. First, as in other works, we use easy-to-get knowledge about the network, namely its path structure, to follow a conditional approach allowing to bound the target metrics (this is based on ideas presented in [5]). We show, in particular, how to derive methods having BWNRV in the homogeneous-components case. We also exhibit a counter-example in the heterogeneous case, that is, a case of unbounded WNRV. Secondly, we explore the randomized quasi-Monte Carlo (RQMC) technique in this context, in order to further reduce the variance of the estimators. These methods are usually effective mostly to estimate the integrals of smooth functions over the unit hypercube, when the function depends only or mostly on a few coordinates. They often perform poorly for discontinuous integrands. However, in our case, RQMC performs very nicely both theoretically (with a provably faster convergence rate) and empirically. Numerical results illustrate and compare the effectiveness of the different techniques considered, as well as their combination.

For general material about Monte Carlo approaches in this area, in addition to some general references [2, 3, 4] given earlier, the reader can see [8] where many different procedures are described. In the same book [7], completely devoted to rare event estimation using Monte Carlo techniques, other chapters contains related material focused on other aspects of the problems and the methods available to solve them.

1.3 A path-based approach

In [5] a technique for facing the problem of rarity is proposed. The idea is to start by building a set $\mathcal{P} = \{P_1, P_2, \dots, P_H\}$ of elementary paths (no node appears more than once in the path) connecting nodes s and t , such that any pair of paths does not share any link (that is, \mathcal{P} is a set of edge-disjoint paths between source and terminal). As we will recall later, this is not a computationally expensive task (compared to the cost of Monte Carlo procedures), that can be performed in polynomial time.

Let $p_h = \prod_{i \in P_h} r_i$ denote the probability that all links of path P_h work. Assume $X^{(1)}, X^{(2)}, \dots$ is a sequence of independent copies of the random configuration X , and that $G(X^{(1)}), G(X^{(2)}), \dots$ is the associated sequence of random partial graphs of G . The main idea of the method is to consider the random variable F equal to the

index of the first graph in this list where every path in \mathcal{P} has at least one link that does not work. Clearly, F is geometrically distributed with parameter $q = \prod_{h=1}^H (1 - p_h)$: that is, $\mathbb{P}(F > f) = (1 - q)^f$, $f \geq 1$. In particular, $\mathbb{E}(F) = 1/q$.

Let us write $P_h = (i_{h,1}, \dots, i_{h,M_h})$ for $P_h \in \mathcal{P}$, and let $b_h = \min_{1 \leq m \leq M_h} b_{i_{h,m}}$ be the order (in ε) of the most reliable edge of P_h . We then have $1 - p_h = \Theta(\varepsilon^{b_h})$ and $q = \Theta(\varepsilon^b)$ where $b = \sum_{h=1}^H b_h > 0$. Observe that $q \rightarrow 0$ as $\varepsilon \rightarrow 0$. The fact that $\mathbb{E}(F) = 1/q$ means that, on the average, we have to wait for $1/q$ samples to find a graph where at least a link is failed in each path of \mathcal{P} . This suggests to sample first from F . If the value f is obtained for F , then we assume that in a “virtual” sequence of copies of $G(X)$, in the first $f - 1$ elements nodes s and t are always connected. It remains to deal with the f th copy. Let Y be a binary random variable defined as follows: if C is the event “every path in \mathcal{P} has at least one link that does not work”, then $\mathbb{P}(Y = 1) = \mathbb{P}(\Phi(X) = 1 | C)$. According to this “interpretation” of the sampling of F , the state of the network in the f th graph is modeled by Y .

We need now a sampling procedure for Y . Consider a path $P_h = (i_{h,1}, i_{h,2}, \dots, i_{h,M_h})$ belonging to \mathcal{P} . Call W_h the r.v. giving the index of the first failed edge of P_h in the order of the links in the path, $W_h \in \{1, 2, \dots, M_h\}$. For each path P_h in \mathcal{P} , we have [5]

$$\Pr(W_h = w) = \frac{r_{i_{h,1}} r_{i_{h,2}} \cdots r_{i_{h,w-1}} (1 - r_{i_{h,w}})}{1 - r_{i_{h,1}} r_{i_{h,2}} \cdots r_{i_{h,M_h}}},$$

which simply translates the definition of W_h into a formula. Sampling Y consists in first sampling the state of every link in the model, and then checking by a standard procedure, typically a DFS (Depth First Search) or a BFS (Breadth First Search) method, if s and t are unconnected or not. Since we are assuming that in every path of \mathcal{P} , at least one link is failing, we first sample the states of the components of P_h for $h = 1, 2, \dots, H$, then the states of the remaining edges in the graph. To sample the states of the links in P_h , we first sample from the distribution of W_h . Assume we get value w . We set the states of edges $i_{h,1}, i_{h,2}, \dots, i_{h,w-1}$ (that is, random variables $X_{i_{h,1}}, \dots, X_{i_{h,w-1}}$) to 1 and that of edge $i_{h,w}$ to 0. The states of the remaining edges in P_h , if any, are sampled from their *a priori* original Bernoulli distributions, and the same for the edges not belonging to any path in \mathcal{P} . Then, we sample from Y , obtaining either 1 or 0 according to the fact that nodes s and t are respectively not connected or connected, and we interpret this as a sample of the state of a network where we know that in every path in \mathcal{P} at least one link is failed.

Resuming, we will build, say, K independent copies F_1, \dots, F_K of F together with K independent copies Y_1, \dots, Y_K of Y , and will use as an estimator of U the number

$$\tilde{U} = \frac{\sum_{k=1}^K Y_k}{\sum_{k=1}^K F_k}.$$

To illustrate the gain obtained with this algorithm, let us consider the “dodecahedron” shown in Figure 1.1, a structure often used as a benchmark for network reliability evaluation techniques. We consider the homogeneous case, where all links have the same unreliability ε . The source and the terminal are nodes 1 and 20.

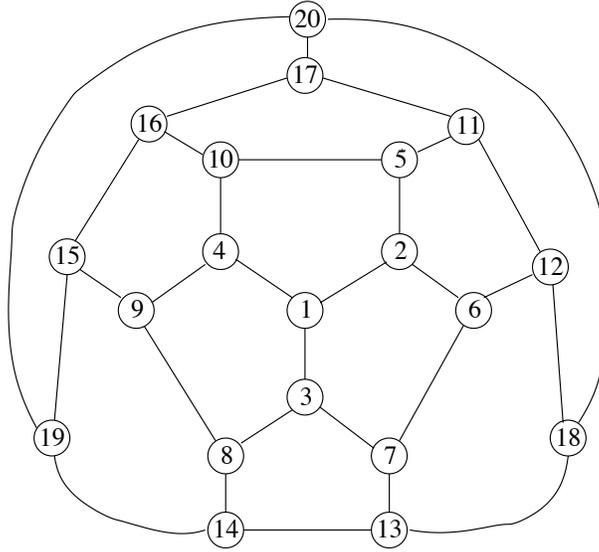


Fig. 1.1 A “dodecahedron” (20 nodes, 30 links). All links have reliability $1 - \varepsilon$.

The gain in efficiency with respect to the standard procedure is captured by the ratio between the WNRV values for \hat{U} and \tilde{U} . We call *relative efficiency* of \tilde{U} with respect to \hat{U} the ratio $\sigma_n^2 \tau_n / (\tilde{\sigma}_n^2 \tilde{\tau}_n^2)$ with $\tilde{\sigma}_n^2$ and $\tilde{\tau}_n^2$ the variance and the mean computation time of \tilde{U} for a sample of size n . We estimated the system unreliability for $n = 10^7$ replications, for three cases: $\varepsilon = 0.1, 0.01$ and 0.001 . The estimated relative efficiency was, respectively, of 18.9, 188.3 and 3800.2 respectively. This illustrates the power of the approach.

1.4 Robustness analysis of the algorithm

In [5], it is pointed out that we can still use a fixed number of samples n , by calling F a random number W of times, where $W = \max\{K \geq 1 : \sum_{k=1}^K F_k \leq n\}$, and using the unbiased estimator

$$U^* = \frac{1}{n} \sum_{k=1}^W Y_k.$$

In other words, we are “wasting” some results (the last ones) of the virtual sampling process associated with \tilde{U} . The variance of U^* is then $\mathbb{V}(U^*) = \sigma_n^2 = U(1-U)/n$, because this is simply an efficient way of implementing the standard estimator.

The point is that while we have not modified the variance with respect to the standard estimator, we did obtain an important gain in time. Let us denote by τ_n^* the average cost in time for the sampling process (that is, sampling W times from the geometric distribution and sampling W times r.v. Y). The WNRV of this procedure

is $\sigma_n^2 \tau_n^* / U^2$. Here, τ_n^* is proportional to $\mathbb{E}(W)$, that is, to nq , leading to

$$WNRV(U^*) = \Theta(\varepsilon^{b-c})$$

where we recall that $U \sim a\varepsilon^c$ for some constant $a > 0$, and that $b = b_1 + b_2 + \dots + b_H$ where the most reliable edge in path P_h has unreliability $\sim d\varepsilon^{b_h}$ for some constant $d > 0$.

Recall that the desirable property (BWNRV) is to have $WNRV(U^*)$ bounded when ε gets small. This means that the estimation remains “efficient” for a given computational time budget, no matter how small ε is. We see that the estimator U^* does not always have this property, and that a sufficient condition for BWNRV is then $b \geq c$, as pointed out in [6].

In Figure 1.2 we see a trivial example where a 3-node model is analyzed using the U^* estimator. We assume homogeneous edges, i.e., edges with reliabilities of the same order of magnitude. In this case, the BWNRV property holds. Indeed, the reader can check that $U = 2\varepsilon^2 - \varepsilon^3 \approx 2\varepsilon^2$ (we are setting all the unreliabilities to the same value ε) and that the variance for a single crude estimation is $U(1-U) = 2\varepsilon^2 - \varepsilon^3 - 4\varepsilon^4 + 4\varepsilon^5 - \varepsilon^6 \approx 2\varepsilon^2$. Calling P_1 the path (s, t) and P_2 the path (s, u, t) , the probabilities that all links of P_1 and P_2 work are $p_1 = 1 - \varepsilon$ and $p_2 = (1 - \varepsilon)^2$ respectively. Thus $q = (1 - p_1)(1 - p_2)$, which here is exactly equal to the target, the system unreliability U , and then $q \approx 2\varepsilon^2$. As a consequence, the BWNRV property is verified. We see that $c = 2$ and that $b = 2$ as well, so that the given sufficient condition is satisfied.

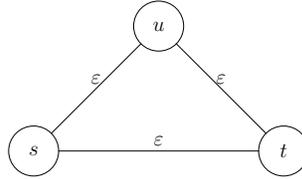


Fig. 1.2 A simple “triangle” illustrating the path-based method leading to bounded relative efficiency. The unreliabilities are all equal to ε . There are 2 paths between s and t , path $P_1 = (s, t)$ and path $P_2 = (s, u, t)$. The probability p_1 that all links in P_1 work is $1 - \varepsilon$ and, for P_2 , we have $p_2 = (1 - \varepsilon)^2$, dealing to $q = (1 - p_1)(1 - p_2) \approx 2\varepsilon^2$. We have $U = 2\varepsilon^2 - \varepsilon^3 \approx 2\varepsilon^2$. Finally, $WNRV = \sigma^2 \tau / U \approx 1$, thus bounded.

Consider now the “bridge” in Figure 1.3, where the links are no longer homogeneous with respect to their reliabilities (or unreliabilities). In the picture, the unreliabilities of the links are indicated.

The unreliability of the system is

$$U = \varepsilon^4(2 + \varepsilon^4 - 2\varepsilon^5 - 2\varepsilon^6 + 2\varepsilon^7) = 2\varepsilon^4 + o(\varepsilon^4).$$

The computations are longer here, but we can check that whatever the set of disjoint paths between s and t , we always have $b < 4$. So, in this case, the path-based method

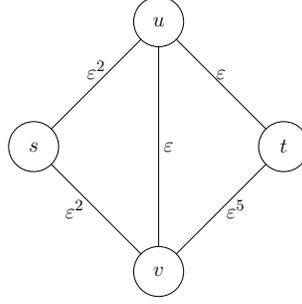


Fig. 1.3 A “bridge” illustrating the path-based method leading to unbounded WNRV

has not the BWNRV property. For the details, there are three possible sets of disjoint minpaths: $\mathcal{P}_1 = \{(s, u, v, t)\}$, $\mathcal{P}_2 = \{(s, v, u, t)\}$ and $\mathcal{P}_3 = \{(s, u, t), (s, v, t)\}$. For each set \mathcal{P}_i , let us denote by q_i the corresponding probability that at least one link in each path is not working. We have:

$$\begin{aligned}
 q_1 &= 1 - (1 - \varepsilon^2)(1 - \varepsilon)(1 - \varepsilon) = 2\varepsilon - 2\varepsilon^3 + \varepsilon^4 \approx 2\varepsilon \\
 q_2 &= 1 - (1 - \varepsilon^2)(1 - \varepsilon)(1 - \varepsilon^5) \approx \varepsilon \\
 q_3 &= (1 - (1 - \varepsilon^2)(1 - \varepsilon^5))(1 - (1 - \varepsilon^2)(1 - \varepsilon^1)) \approx \varepsilon^2 \varepsilon^1 = \varepsilon^3.
 \end{aligned}$$

Then, for the three cases, BWNRV is not verified because we respectively have for the three cases $\text{WNRV} = \Theta(\varepsilon^{-3})$ for \mathcal{P}_1 , $\text{WNRV} = \Theta(\varepsilon^{-3})$ for \mathcal{P}_2 and $\text{WNRV} = \Theta(\varepsilon^{-1})$ for \mathcal{P}_3 .

Coming back to the homogeneous case, illustrated by the elementary example of Figure 1.2, let us show that it is always possible to find a set of paths \mathcal{P} leading to the BWNRV property of the corresponding estimator U^* . This has been briefly stated in [6]. We provide a more detailed proof here.

Theorem 1. *Assume that the unreliabilities of the links are homogeneous in ε , that is, that for any link i in the graph, we have $u_i = a_i \varepsilon$. Then, it is always possible to find a set of minpaths \mathcal{P} such that the corresponding estimator U^* has the BWNRV property.*

Proof. First, observe that it is useless to put the same exponent, say β , to the ε factor in the link unreliabilities, since we can then rename ε^β as the new ε in the analysis.

The breadth of a graph is the size of a minimal size mincut. Let K be the number of mincuts in the graph, which we arbitrary order and number from 1 to K . Let C_k be the event “all links in the k th mincut are failed”. Writing

$$U = \Pr(C_1 \cup \dots \cup C_K),$$

and using Poincare’s formula for expanding this expression, we see that the term with the lowest power in ε is of the form $a\varepsilon^c$ where c is precisely the breadth of the graph. For this, just observe that for each mincut C_k of minimal size c , $\mathbb{P}(C_k) =$

$\Theta(\varepsilon^c)$, and that for any other $\mathbb{P}(C_j)$ and for all terms of the form $\mathbb{P}(C_i \cap C_j \cap \dots)$ we obtain $\Theta(\varepsilon^d)$, $d > c$.

The second observation comes from the theory of flows in graphs, where a basic result states that if c is the breadth, then there exist c disjoint paths from s to t . For an effective way to find them, they come for instance directly as a byproduct of the marking process in Ford-Fulkerson algorithm (for finding a maximal flow from s to t), which runs in time polynomial in the size of the graph [16]. Then, we just see that with the previous notation, for each of the $H = c$ minpaths, $b_h = 1$ and thus $b = c$, which is sufficient for having the BWNRV property. \square

1.5 Improvement

The estimator \tilde{U} does not have the same variance as U^* and is more difficult to analyze; it actually has a (slightly) smaller variance and the same computational cost. The goal in [5] is to point out that the standard estimator can still be very useful when dealing with rare events if an efficient implementation is possible. That means, in particular, to keep F as a geometric random variable.

Looking now for efficiency improvements, we can replace the random variable F by its mean (instead of sampling it). Let us look at what happens in this case. If F is replaced by its expected value, then exactly one in $1/q$ independent graphs will have at least one failed link on each path of \mathcal{P} . Recall that Y is a Bernoulli random variable that is 1 if the graph is failed and 0 otherwise, conditioned on the fact that at least one link on each selected path is failed. The random variable $Z = qY$ is then an (unbiased) estimator of U over such a block. This is known as a *conditional Monte Carlo* estimator [12]: the usual estimator has been replaced by its conditional expectation given Y . A confidence interval for U is obtained by considering independent copies of Z and applying standard procedures.

Define p as the probability that $Y = 1$. Obviously, $U = qp$, and $\mathbb{V}(Z) = q^2\mathbb{V}(Y) = q^2p(1-p)$. If we look at the ratio of the WNRV of Z (considering the expected value of F) over the WNRV of the estimator U^* (obtained by employing the geometric distribution) and if we neglect the time to generate the geometric r. v., we get the following relative efficiency:

$$\frac{\text{WNRV}(Z)}{\text{WNRV}(U^*)} = \frac{qU(1-U)}{q^2p(1-p)} = \frac{1-qp}{1-p} > 1. \quad (1.1)$$

This shows that the conditional Monte Carlo estimator always yields an efficiency improvement that we are able to characterize, by reducing the WNRV. The cost (in CPU time) is also reduced because there is no longer a need for sampling from a geometric law. Note that in general, conditional Monte Carlo *always* reduce the variance.

Let us illustrate this improvement on a few examples. Consider first the bridge shown in Figure 1.3, but with all its links identical. For the path-based method, we

use the two symmetric paths between s and t of size 2: $\mathcal{P} = \{(s, u, t), (s, v, t)\}$. In Table 1.1 we see that the improvement roughly doubles the efficiency of the original approach.

u_i , for all link i	estimation	variance	rel. efficiency (relation (1.1))
0.1	2.1 e-2	3.1 e-11	2.4
0.01	2.0 e-4	3.9 e-14	2.0
0.001	2.0 e-6	4.0 e-19	2.0

Table 1.1 For three cases, where the system failure event becomes rarer (ϵ going from 0.1 to 0.001), we show the result of the estimation, the variance of the estimator and the relative efficiency with respect to the original method. The model is the “bridge” described in Figure 1.3.

Now, we evaluate the unreliability in the case of the topology given in Figure 1.4 with homogeneous links, where $s = 1$ and $t = 14$. The breadth of the graph is $c = 3$, so, to use an estimation procedure having the BWNRV property, we need three disjoint elementary paths between s and t . The three paths chosen are $P_1 = (1, 2, 6, 8, 9, 13, 14)$, $P_2 = (1, 3, 7, 10, 14)$ and $P_3 = (1, 4, 7, 11, 12, 14)$.

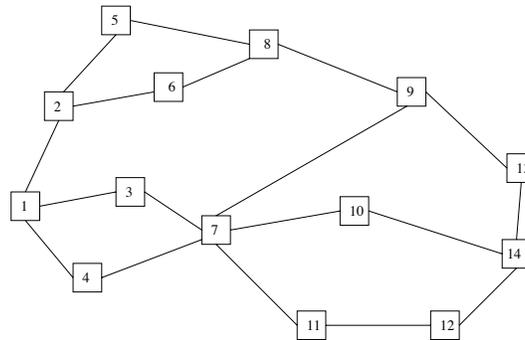


Fig. 1.4 We call this example a “reducible” topology, because there are many series-parallel simplifications possible here, when $s = 1$ and $t = 14$. After those reductions, the result is a bridge (see [4] for instance). In the homogeneous case, we can easily see, after some algebra, that when every link has the same unreliability $u_i = \epsilon$, the system unreliability is $U = 24\epsilon^3 + o(\epsilon^3)$. The model is the “reducible” architecture, Figure 1.4.

In Table 1.2 we show the relative efficiency of the proposed improvement for this “reducible” architecture. As we can see, the efficiency improvement is still significant, while less than in the previously presented small bridge example.

Finally, we consider in Table 1.3 the more challenging dodecahedron structure given in Figure 1.1. We performed the same experiments as with previous examples, in order to show that in this case there is no improvement over the original method (relative efficiency close to 1). The reason is that given the density of the graph, the probability $p = \mathbb{P}(Y = 1)$ is small, leading to a relative efficiency of $(1 - qp)/(1 - p) \approx 1$.

u_i , for all link i	estimation	variance	rel. efficiency
0.1	1.9 e-2	8.9 e-11	1.4
0.01	2.0 e-5	3.7 e-16	1.3
0.001	2.0 e-8	2.0 e-22	1.2

Table 1.2 We evaluate the graph given in Figure 1.4 when the elementary unreliability of all links is equal to 0.1, 0.01, 0.001.

u_i , for all link i	estimation	variance	rel. efficiency
0.1	2.9 e-3	2.9 e-12	1.02
0.01	2.0 e-6	4.1 e-18	1.01
0.001	2.0 e-9	4.3 e-25	1.01

Table 1.3 We evaluate the graph given in Figure 1.1 when the elementary unreliability of each link is equal to 0.1, 0.01, 0.001.

In the next section, we show that the efficiency can be improved further by using RQMC on top of the method proposed earlier.

1.6 Acceleration by Randomized Quasi-Monte Carlo

The previous sections make use of Monte Carlo methods. Very roughly, the basic idea is to choose sample points randomly and independently according to a given distribution. This random choice of points ensures that asymptotically, the empirical distribution of the estimator converges to the theoretical one at a speed of $O(n^{-1/2})$ for a sample size n . This rate can be improved thanks to better spreading of points (which are then no longer independent). This is the basic principle of *quasi-Monte Carlo* (QMC) methods [13]. In practice, randomized versions called RQMC are often used in order to obtain an unbiased estimator and allow error estimation. We will now explain briefly the QMC and RQMC methods before applying them to our static reliability problem.

Note that RQMC is not an appropriate method to handle the problem of rare events, but once that problem is handled (in our case via a path-based conditional Monte Carlo approach), RQMC can improve the efficiency by an additional order of magnitude.

1.6.1 Quasi-Monte Carlo methods

In most simulation studies by computer (including ours), a single (random) realization of the model is defined as a function of a uniform random variable over $(0, 1)^M$, or equivalently from M independent unidimensional uniform random variables over

$(0, 1)$, where M is possibly unbounded; those uniform random variates are actually replaced in practice by the output of a pseudorandom generator in Monte Carlo methods. To describe QMC and RQMC techniques, we will therefore use (without loss of generality) the framework of an estimation over the hypercube $(0, 1)^M$.

Suppose we want to estimate

$$\mathbb{E}[f(U)] = \int_{[0,1]^M} f(u) du,$$

where U is uniformly distributed over $[0, 1]^M$. While Monte Carlo methods use a sample $\{U_i, 1 \leq i \leq n\}$ of n independent random variables with the same distribution than U to get $(1/n) \sum_{i=1}^n f(U_i)$ as the estimator, QMC methods [10, 13] replace the independent U_i 's by a sequence of *deterministic* points $\mathcal{E} = \{\xi_n, n \geq 1\}$ in $[0, 1]^M$. A basic requirement is that the sequence is asymptotically uniformly distributed, in the sense that the proportion of points among the first n in the sequence \mathcal{E} falling in any (multivariate) interval B , namely $A_n(B, \mathcal{E}) = \#\{\xi_i, 1 \leq i \leq n : \xi_i \in B\}/n$, converges to $\lambda(B)$ as $n \rightarrow \infty$, where $\lambda(B)$ is the Lebesgue measure of B . There exist several different measures of the *discrepancy* between the empirical distribution of the n first points of the sequence and the uniform distribution. One of them is the *star discrepancy*, defined as

$$D_n^*(\mathcal{E}) = \sup_{[0,x] \subset [0,1]^M} \left| \frac{A_n([0,x], \mathcal{E})}{n} - \lambda([0,x]) \right|,$$

which takes the sup over all intervals with one corner at the origin. A sequence \mathcal{E} is actually asymptotically uniformly distributed if and only if $D_n^*(\mathcal{E}) \rightarrow 0$ as $n \rightarrow \infty$.

Discrepancy measures are helpful to bound the error in the estimation of the integral $\int_{[0,1]^M} f(u) du$. Using the star discrepancy, the Koksma-Hlawka bound [13] is

$$\left| \frac{1}{n} \sum_{k=1}^n f(\xi^{(k)}) - \int_{[0,1]^M} f(u) du \right| \leq V(f) D_n^*(\mathcal{E})$$

where $V(f)$ is the variation of the function f in the sense of Hardy and Krause [13]. For the best known sequences \mathcal{E} , we have $D_n^*(\mathcal{E}) = O(n^{-1}(\log n)^M)$ [13]; these are named *low discrepancy sequences*. In this paper we use one class of low-discrepancy sequences called the *Sobol' sequences* [17]. Those sequences are instances of (t, M) -sequences in base 2, which means that for a certain integer $t \geq 0$, $\forall m \geq t$, if we consider a set of 2^m successive points of the form $\{\xi^{(j)} : k2^m \leq k < (j+1)2^m\}$, for any $k \geq 0$ and $m > t$, and any dyadic interval

$$E = \prod_{i=1}^M [a_i 2^{-d_i}, (a_i + 1) 2^{-d_i}], \text{ where } a_i, b_i \in \mathbb{N}, d_i \geq 0, a_i \in \{0, 1\},$$

of size $\lambda(E) = 2^{t-m}$ with $m \geq t$, then the number of points in E is exactly 2^t . This means that for any function f which is constant in each dyadic interval of size 2^{t-m} ,

the integration error by a set of 2^m successive points of the above form is always zero.

QMC methods therefore asymptotically outperform MC, but from the practical side, evaluating the error is a very difficult task in general. The worst-case error bounds such as the Koksma-Hlawka bound are too hard to compute in practice and are often much too large to be useful anyway. Even if the bound converges asymptotically at rate $O(n^{-1}(\log n)^M)$, it often takes an astronomically large value of n before this bound becomes meaningful, as soon as the dimension M exceeds 10 or so [18]. Nevertheless, QMC methods are typically more effective than what the bounds tell us. RQMC methods permit one to estimate the error without relying on these bounds.

1.6.2 Randomized quasi-Monte Carlo methods

RQMC methods randomly perturb a low-discrepancy sequence *without losing its good distribution over* $[0, 1]^M$. A simple illustration of this is when all the points are shifted by the same uniform vector U . That is, \mathcal{E} is replaced by its randomly-shifted version $\{V_k := (\xi_k + U) \bmod 1, k \geq 1\}$, where “mod 1” means that we retain only the fractional part of each coordinate. Thus, the whole sequence is somehow just translated over the interval. Other types of randomization exist [10]; some of them are adapted to the structure of the low-discrepancy sequence. For the Sobol’ sequence, a *random digital shift* generates a uniform point in $[0, 1]^M$, expands each of its coordinates in base 2, and adds the digits modulo 2 to the corresponding digits of each point of the sequence. This randomization preserves the (t, M) -sequence property. With this particular sequence and randomization, if we assume that f has bounded variation, the variance of $(1/n)\sum_{k=1}^n f(V_k)$ is $O(n^{-2}(\log n)^{2M})$, which converges faster than the Monte Carlo rate of $O(1/n)$. The convergence speed can be even faster for specific classes of smooth functions (with square-integrable high-order partial derivatives, for example) and adapted randomized sequences [10, 14].

To estimate the error, one can make m of independent replicates of the randomization, and estimate the variance in a classic way by the sample variance of these m replicates. The central limit theorem applies when $m \rightarrow \infty$. In practice, confidence intervals are often computed by assuming (heuristically) that the average is approximately normally distributed even when m is small.

QMC/RQMC error bounds degrade rapidly when the dimension M increases, because the $(\log n)^M$ term becomes more important and a much larger value of n is required before this term is dominated by the $1/n$ term. As a general rule of thumb, QMC/RQMC is more effective when the dimension M is small, but sometimes it also works well in practice even when M is large [10]. This happens when the integrand f depends mostly on just a few coordinates, or can be decomposed (approximately) as a sum of terms where each term depends only on a small number of coordinates [15]. We then say that the integrand has *low effective dimension*.

1.6.3 Application to our static reliability problem

We now examine how to apply RQMC to our static reliability problem, starting with a crude implementation. We need to sample the status of M links. The state of the j th link in the i th replicate is sampled from the j th coordinate of the i th point of the low-discrepancy sequence: if this coordinate is less than r_j , then the state is 1, otherwise it is 0. Let ψ be the indicator function mapping each point $y = (y_1, \dots, y_M) \in [0, 1]^M$ to a vector state $x = (x_1, \dots, x_M)$ in $\{0, 1\}^M$, defined by $x_j = 1$ if $y_j < r_j$, and $x_j = 0$ otherwise. This mapping partitions the unit hypercube $[0, 1]^M$ into 2^M rectangular boxes, each one sharing one corner with the hypercube. The indicator function $\Phi \circ \psi$, where ‘ \circ ’ denotes the composition operator, takes a constant value over each of those boxes: It is equal to 0 for states in which the system is failed, and 1 for the other states. The reliability is therefore $R = \int_{[0,1]^M} \Phi \circ \psi(y) dy$ and the unreliability $U = \int_{[0,1]^M} (1 - \Phi) \circ \psi(y) dy$. We call *minimal state vector* any vector $z \in \{0, 1\}^M$ such that $\Phi(z) = 1$, and $\Phi(x) = 0$ for all $x < z$. Let N_p be the number of minimal state vectors (they correspond to elementary paths in the graph). We similarly define a *maximal state vector* as any vector $z \in \{0, 1\}^M$ such that $(1 - \Phi)(z) = 1$, and $(1 - \Phi)(x) = 0$ for all $x > z$. Let N_c be the number of maximal state vectors (corresponding to minimal cuts in the graph). Observe that the estimation error is the same when estimating the reliability or the unreliability, i.e.,

$$\left| \frac{1}{n} \sum_{i=1}^n \Phi \circ \psi(y_i) - \int_{[0,1]^M} \Phi \circ \psi(y) dy \right| = \left| \frac{1}{n} \sum_{i=1}^n (1 - \Phi) \circ \psi(y_i) - \int_{[0,1]^M} (1 - \Phi) \circ \psi(y) dy \right|. \quad (1.2)$$

Theorem 2. *We have the worst-case error bound*

$$\left| \frac{1}{n} \sum_{i=1}^n \Phi \circ \psi(y_i) - \int_{[0,1]^M} \Phi \circ \psi(y) dy \right| \leq (2^{\min(N_p, N_c)} - 1) D_n^*(\Xi).$$

Proof. Let $\{\pi_1, \dots, \pi_{N_p}\}$ be the set of minimal state vectors. For each π_ℓ , we define the corresponding sub-interval P_ℓ of $[0, 1]^M$ by

$$P_\ell = \prod_{i=1}^M [0, \alpha_i] \quad \text{where} \quad \begin{cases} \alpha_i = r_i & \text{if } \ell\text{-th coordinate of } \pi_\ell \text{ is } 1, \\ \alpha_i = 1 & \text{otherwise.} \end{cases}$$

Note that these P_ℓ 's are not disjoint. The subset of $[0, 1]^M$ on which $\Phi \circ \psi(y) = 1$ is $B = \cup_{\ell=1}^{N_p} P_\ell$. Furthermore,

$$\left| \frac{1}{n} \sum_{k=1}^n \Phi \circ \psi(\xi^{(k)}) - \int_{[0,1]^M} \Phi \circ \psi(y) dy \right| = \left| \frac{1}{n} \sum_{k=1}^n 1_B(\xi^{(k)}) - \lambda(B) \right|.$$

Applying the Poincaré formula and the triangular inequality,

$$\begin{aligned}
& \left| \frac{1}{n} \sum_{k=1}^N \Phi \circ \psi(\xi^{(k)}) - \int_{[0,1]^M} \Phi \circ \psi(y) dy \right| \\
&= \left| \sum_{\ell=1}^{N_p} (-1)^{\ell-1} \sum_{1 \leq h_1 < \dots < h_\ell \leq N_p} \left(\frac{1}{n} \sum_{\xi^{(k)} \in \cap_{j=1}^{\ell} P_{h_j}} 1_{\cap_{j=1}^{\ell} P_{h_j}}(\xi^{(k)}) - \lambda(\cap_{j=1}^{\ell} P_{h_j}) \right) \right| \\
&\leq \sum_{\ell=1}^{N_p} \sum_{1 \leq h_1 < \dots < h_\ell \leq N_p} \left| \frac{1}{n} \sum_{\xi^{(k)} \in \cap_{j=1}^{\ell} P_{h_j}} 1_{\cap_{j=1}^{\ell} P_{h_j}}(\xi^{(k)}) - \lambda(\cap_{j=1}^{\ell} P_{h_j}) \right| \\
&\leq \sum_{\ell=1}^{N_p} \sum_{1 \leq h_1 < \dots < h_\ell \leq N_p} D_n^*(\mathcal{E}) \\
&= (2^{N_p} - 1) D_n^*(\mathcal{E}).
\end{aligned}$$

Proceeding exactly in the same way for computing the error when estimating the unreliability from the set of maximal states instead of minimal ones, we get

$$\left| \frac{1}{n} \sum_{k=1}^N (1 - \Phi) \circ \psi(\xi^{(k)}) - \int_{[0,1]^M} (1 - \Phi) \circ \psi(y) dy \right| \leq (2^{N_c} - 1) D_n^*(\mathcal{E}).$$

From (1.2) and combining the two above inequalities, we obtain the theorem. \square

This result provides a worst-case error bound that converges asymptotically as $O(n^{-1}(\log n)^{N_p})$. The corresponding RQMC variance is $O(n^{-2}(\log n)^{2N_p})$. We may nevertheless need a very large n before this RQMC approach beats MC when N_p is large.

To apply RQMC with our path-based technique based on conditional Monte Carlo, the random variable Y for the i th replicate is sampled by first generating the first non-working link on each path from the initial coordinates of the point ξ_i , and then sampling all the other links (whose state is not yet known) from the remaining coordinates of ξ_i . The overall dimension of the integrand is again M , because in the worst-case we may need to sample all links, if the first link on each path is failed. Nevertheless, the number of required coordinates (or uniform random numbers) is often smaller than M , and the first few coordinates are more important. As a result, the RQMC method tends to be more effective. A worst-case error bound in terms of the discrepancy $D_n^*(\mathcal{E})$ can also be obtained, as for the crude implementation of RQMC discussed earlier.

1.6.4 Numerical results

We made an experiment to compare MC and RQMC for our three typical examples, the bridge, the dodecahedron, and the reducible topology, in each case with three

values of the links reliability ε : 0.9, 0.99 and 0.999. For RQMC, we use the first n points of a Sobol' sequence with a random digital shift and we perform $m = 500$ independent randomizations. For MC, we make nm independent replications (same total sample size). In both cases, we compute the half-width of a 95% confidence interval on the unreliability, using the path-based technique with conditional Monte Carlo. We then compute the ratio of the confidence interval half-width of MC over that of RQMC. The results are in Table 1.4, where "Half-width MC" is the half-width for MC, "Half-width RQMC" is that for RQMC, and "Ratio" is the ratio between the two.

We see that RQMC brings a significant variance reduction in all cases, even on reasonable-size topologies such as the dodecahedron. Also, the larger the cardinality n of the RQMC point set, the more the variance is reduced.

Topology	ε	n	Half-width MC	Half-width RQMC	Ratio
Bridge	0.9	2^{10}	9.70E-5	1.61E-5	0.166
Bridge	0.9	2^{14}	2.43E-5	1.55E-6	6.41E-2
Bridge	0.9	2^{20}	3.03E-6	4.21E-8	1.39E-2
Bridge	0.99	2^{10}	1.08E-6	1.05E-7	9.68E-2
Bridge	0.99	2^{14}	2.71E-7	8.15E-9	3.01E-2
Bridge	0.99	2^{20}	3.39E-8	2.20E-10	6.48E-3
Bridge	0.999	2^{10}	1.09E-8	7.25E-10	6.62E-2
Bridge	0.999	2^{14}	2.74E-9	3.17E-11	1.16E-2
Bridge	0.999	2^{20}	3.42E-10	1.19E-12	3.47E-3
Dodecahedron	0.9	2^{10}	9.30E-5	6.89E-5	0.741
Dodecahedron	0.9	2^{14}	2.33E-5	1.29E-5	0.556
Dodecahedron	0.9	2^{18}	5.81E-6	2.58E-6	0.444
Dodecahedron	0.99	2^{10}	1.10E-7	5.28E-8	0.479
Dodecahedron	0.99	2^{14}	2.77E-8	7.62E-9	0.275
Dodecahedron	0.99	2^{18}	6.93E-9	1.37E-9	0.197
Dodecahedron	0.999	2^{10}	1.13E-10	4.84E-11	0.430
Dodecahedron	0.999	2^{14}	2.83E-11	5.45E-12	0.193
Dodecahedron	0.999	2^{18}	7.07E-12	7.92E-13	0.112
Reducible	0.9	2^{10}	1.64E-4	8.18E-5	0.499
Reducible	0.9	2^{14}	4.09E-5	1.58E-5	0.386
Reducible	0.9	2^{18}	1.02E-5	2.49E-6	0.244
Reducible	0.99	2^{10}	2.36E-7	5.57E-8	0.236
Reducible	0.99	2^{14}	5.91E-8	9.96E-9	0.168
Reducible	0.99	2^{18}	1.48E-8	1.63E-9	0.111
Reducible	0.999	2^{10}	2.44E-10	3.70E-11	0.152
Reducible	0.999	2^{14}	6.10E-11	5.07E-12	8.31E-2
Reducible	0.999	2^{18}	1.53E-11	7.38E-13	4.83E-2

Table 1.4 Confidence interval half-widths for MC and for RQMC using the same total computing budget, and their ratio. The RQMC estimates are based on 500 independent replicates with n points. All edges in the network have reliability ε .

The fact that the improvements are smaller as the model size increases is due to the sensitivity of QMC methods with respect to the dimension of the problem. Basically, when the dimension is higher, the low discrepancy sequence needs more time to “distribute” its points well [10].

1.7 Conclusion

We have proposed and examined simulation techniques for static rare event models. Our discussion emphasizes the importance of an efficiency measure that account for both the accuracy of Monte Carlo methods and the cost (in CPU time) of the estimation procedures. A key concept that captures these ideas in the context of rare-event simulation is the notion of bounded work-normalized relative variance (BWNRV). The application that we considered is the analysis of a reliability metric in a static model. Our analysis was completed by proposals designed to improve efficiency in the considered estimation algorithms.

A last technical remark now on the BWNRV property: the computing time used in the definition may have unbounded relative variance itself, which may lead to a noisy work-normalized variance [9, 11]. In that case, we cannot assert that the probability that the estimator is within a value δ of its mean for a given computational budget c , goes to 0 uniformly in ε when c increases. Our definition only looks at the first moment of the computational time, which is less stringent. Considering also the second moment is a subject of further research.

References

1. C.J. Colbourn The Combinatorics of Network Reliability. Oxford University Press, 1987.
2. I. B. Gertbakh Statistical Reliability Theory. Marcel Dekker, Inc., New York and Bassel, 1989.
3. M.O. Ball, C.J. Colbourn and J.S. Provan. Network reliability. in Handbook of Operations Research: Network Models, pp. 673–762, Elsevier North-Holland, 1995.
4. G. Rubino. Network reliability evaluation. Chapter 11 in *State-of-the art in performance modeling and simulation*, K. Bagchi and J. Walrand editors, Gordon and Breach Books, 1998.
5. M. El Khadiri and G. Rubino. A Time Reduction Technique for Network Reliability Analysis. In *MCQMC'00: 4th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, 2000.
6. H. Cancela, G. Rubino, and B. Tuffin. New measures of robustness in rare event simulation. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, editors, *Proceedings of the 2005 Winter Simulation Conference*, pages 519–527, 2005.
7. G. Rubino and B. Tuffin (editors). Rare Event Simulation, edited by G. Rubino and B. Tuffin, John Wiley & Sons, 2009.
8. H. Cancela, M. El Khadiri, and G. Rubino. Rare Event Analysis by Monte Carlo Techniques in Static Models. in Rare Event Simulation, edited by G. Rubino and B. Tuffin, John Wiley & Sons, 2009.

9. P. W. Glynn, G. Rubino and B. Tuffin. Robustness Properties and Confidence Interval Reliability Issues. in *Rare Event Simulation*, edited by G. Rubino and B. Tuffin, John Wiley & Sons, 2009.
10. P. L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics*, 2009. To appear.
11. P. L'Ecuyer, J. H. Blanchet, B. Tuffin, and P. W. Glynn. Asymptotic robustness of estimators in rare-event simulation. *ACM Transactions on Modeling and Computer Simulation*, 2009. To appear.
12. P. Bratley, B. L. Fox and L. E. Schrage. *A Guide to Simulation, 2nd Edition*. Springer-Verlag, New York, 1987.
13. H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. CBMS-NSF, SIAM, Philadelphia, 1992.
14. A. B. Owen. Scrambled net variance for integrals of smooth functions. *Annals of Statistics*, 25(4):1541–1562, 1997.
15. A. B. Owen. Latin supercube sampling for very high-dimensional simulations. *ACM Transactions on Modeling and Computer Simulation*, 8(1):71–102, 1998.
16. R. Sedgewick. *Algorithms in C, Part 5: Graph Algorithms, 3rd Edition*. Addison-Wesley Professional, 2001.
17. I. M. Sobol'. The distribution of points in a cube and the approximate evaluation of integrals. *U.S.S.R. Comput. Math. and Math. Phys.*, 7:86–112, 1967.
18. B. Tuffin. Variance Reductions applied to Product-Form Multi-Class Queuing Network. *ACM Transactions on Modeling and Computer Simulation*, 7(4):478–500, 1997.

