Chapter 7

Rare events analysis by Monte Carlo techniques in static models

Héctor Cancela, Mohammed El Khadiri and Gerardo Rubino

This chapter discusses Monte Carlo techniques for rare event simulation in the case of *static models*, that is, models where time is not an explicit variable. The main example and the one that will be used in the chapter is the network reliability analysis problem, where the models are graphs with probabilities associated with their components (arcs or edges, and/or nodes). Other typical names in this domain are fault trees, block diagrams, etc. All these models are in general solved using combinatorial techniques, but only for quite small sizes, because their analysis is extremely costly in computational resources. The only methods able to deal with models having arbitrary size are Monte Carlo techniques, but there, the main difficulty to face with is the rare event case, the focus of this chapter. In many areas (in telecommunications, transportation systems, energy productions plants...), either the components are very reliable or redundancy schemes are adopted, resulting in extremely reliable systems. This means that a system's failure is (or should be) a rare event.

7.1 Introduction

The most referenced example in the area of static models in dependability analysis is the network reliability problem. It concerns the evaluation of reliability metrics of large classes of multicomponent systems. We will denote by $\mathcal{E}$ the set of components in the system (which will be, in a few lines, identified to the set of edges of the undirected graph modeling the system). In general, the structure of such a system is represented by a binary function Φ of $|\mathcal{E}|$ binary variables. The usual convention for the state of a component or for the whole system is that 1 represents the operational state (the device, component or system, is operational or *up*) and 0 represents the failed or *down* state. A *state vector* or *system configuration* is a vector $\vec{x} = (x_1, \ldots, x_{|\mathcal{E}|})$ where x_i is a possible state, 0 or 1, of the ith component (that is, $\vec{x}$ is an element of $[0,1]^{|\mathcal{E}|}$). With this notation, $\Phi(\vec{x}) = 1$ if the system is *up* when the configuration is $\vec{x}$, and 0 otherwise.

We may have different structure functions associated with the same system, each addressing a specific aspect of interest that must be evaluated (see below). Frequently (but not always) structure functions are *coherent*, corresponding to systems verifying the following properties: (i) when all the components are down (resp. up), the system is down (resp. up); (ii) if the system is up (resp. down) and we change the state of a component from 0 to 1 (resp. from 1 to 0), the system remains up (resp. down); (iii) all the components are relevant (a component i is irrelevant if the state of the system does not depend on the state of i). Formally, let us denote by $\vec{0}$ (resp. by $\vec{1}$) a state vector having all its entries equal to 0 (resp. equal to 1). We also denote by $\vec{x} \leq \vec{y}$ the relation $x_i \leq y_i$ for all i, by $\vec{x} < \vec{y}$ the fact that $\vec{x} \leq \vec{y}$ with, for some j, $x_j < y_j$, and by $(\vec{x}, 0_i)$ (resp. by $(\vec{x}, 1_i)$) the state vector constructed from $\vec{x}$ by setting x_i to 0 (resp. to 1). Then, Φ is coherent iff (i) $\Phi(\vec{0}) = 0$, $\Phi(\vec{1}) = 1$; (ii) if $\vec{x} < \vec{y}$ then $\Phi(\vec{x}) \leq \Phi(\vec{y})$ and (iii) for each component i there exists some state vector $\vec{x}$ such that $\Phi(\vec{x}, 0_i) \neq \Phi(\vec{x}, 1_i)$ (and thus, due to (ii), $\Phi(\vec{x}, 0_i) = 0$ and $\Phi(\vec{x}, 1_i) = 1$).

After specifying the function Φ, which defines what means that the system provides the service for which it was designed, a probabilistic structure must be added to take into account the failure processes. The usual framework is to assume that the state of the ith component is a random binary (Bernoulli) variable (r.v.) X_i with expectation $\mathbb{E}(X_i) = r_i$, and that the $|\mathcal{E}|$ r.v. $X_1, \ldots, X_{|\mathcal{E}|}$ are independent. The numbers $r_i = \mathbb{P}(X_i = 1)$ (called the *elementary reliabilities*) are input data. Sometimes we will also use the notation q_i for the *unreliability* of link i, that is, $q_i = 1 - r_i$. The output parameter is the reliability R of the system, defined by

$$R = \mathbb{P}(\Phi(\vec{X}) = 1) = \mathbb{E}(\Phi(\vec{X})) \tag{7.1}$$

where $\vec{X} = (X_1, \ldots, X_{|\mathcal{E}|})$, or its *unreliability* $Q = 1 - R$. Observe that this is a static problem, that is, time is not explicitly used in the analysis. When time relations are considered, the context changes and the general framework in which the analysis is usually done is the theory of stochastic processes and, in particular, of Markov processes (see Chapter 6 in this monograph). For an exposition concerning the general theory (including dynamic models) the reader can see [6], [7] or [31].

The structure function can be specified by providing a table describing the mapping from $[0,1]^{|\mathcal{E}|}$ into $[0,1]$, a sort of exhaustive description, or, at the other side of the spectrum, by a program (or an algorithm), which usually is a compact way of giving the function. An intermediate option is to define it by giving a *stochastic graph*, sometimes called a *network* in this context. These models are very useful, in particular, for communication network analysis. We will adopt them here as the reference systems. The lines of the communication network are modeled by the edges (or by the arcs in the directed case) of the graph, and the vertices represent the nodes. The basic model in this class (and in this chapter) is an undirected graph (lines are assumed to be bidirectional) with perfect nodes (corresponding to the situation where the reliability of a node is much higher than the reliability of a line), assumed to be connected and without loops. The state of line i at some instant of interest is a binary r.v. X_i. The structure function Φ is then specified by means of some property of the graph. To be more specific, let us denote by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the graph where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. The set $\mathcal{E}'$ of operational lines at the fixed considered instant defines a random subgraph $G = (\mathcal{V}, \mathcal{E}')$ of the previous one. The reliability of the system is then the probability that G has some graph property. For instance, if we are interested in the fact that all the nodes can communicate with each other and we want to quantify the ability of the network to support this, the corresponding metric, called *all–terminal* reliability, is the probability that G is connected. Another important case is when the user is interested only in the communications between two particular nodes, usually called *source* and *terminal*. Denoting these nodes by s and t, the associated metric is the so-called *2–terminal* or *source–to–terminal* reliability, defined as the probability that there exists in G at least one path between s and t having all its lines operational. This last case of graphs having an "entry" point s and an "exit" point t (the terminology is used even if the graph is undirected), has a broad field of applications since it is a general tool describing the structure of a system, its *block diagram*, not only in the communications area. For instance, it is widely used in circuit analysis or more generally in the description of electrical systems. The previous considered metrics are particular cases of the *$\mathcal{K}$–terminal* reliability in which a subset $\mathcal{K}$ of nodes is defined and the associated measure is the probability that all the nodes in $\mathcal{K}$ can communicate, that is, the probability that the nodes of $\mathcal{K}$ belong to the same connected component of G. By far, most of the research effort in the network reliability area has been done on the evaluation (exact or approximate) of this measure and the two particular cases described before (the all-terminal and 2-terminal ones).

These problems (and several other related reliability problems) have received considerable attention from the research community (see, for instance, [45], [18], [4] and [55] for references) mainly because of the general applicability of these models, in particular in the communication network area, and because of the fact that in the general case the computation of these metrics is in the #*P*–complete class [59], [3], a family of *NP*–hard problems not known to be in *NP*. A #*P*–complete problem is equivalent to counting the number of solutions to a *NP*–complete one (about connections between counting problems and rare event simulation, see Chapter 8 in this book). This implies that a #*P*–complete problem is at least as hard as a *NP*–complete one. This last fact justifies the continued effort to find faster solution methods. Concerning network reliability, even if we limit the models to very particular classes, the problems remain #*P*–complete. For instance, this is the case if we consider the 2–terminal reliability evaluation on planar graph with vertex degree at most equal to three.

It must be stated that all known *exact* techniques available to evaluate R are unable to deal with a network having, say, one hundred elements (except, of course, in case of particular types of topologies). For instance, in [51], the effective threshold is placed around 50 components. Our own experience confirms this figure. In [55] the different approaches that can be followed to evaluate these metrics numerically (exact combinatorial methods and bounding procedures, together with reduction techniques that allow to reduce the size of the models) are discussed, and some examples illustrate the limits of the different possible techniques (including simulation). Let us observe that in the communication networks area, usual model sizes are often very large. For instance, in [32] the authors report on computational results analyzing (in a deterministic context) the topology of real fiber optic telephone networks. They give the sizes of seven networks provided by Bell Communications Research, ranged from 36 nodes and 65 edges to 116 nodes and 173 edges. They also say that in this type of communication system, the number of nodes in practical implementations is not larger than, say, 200 nodes. In [33], the same authors report on a realistic model of the link connections in the global communication system of a ship, having 494 nodes and 1096 edges. Monte Carlo algorithms appear then to be the only way to obtain (probabilistic) answers to reliability questions for networks having, for instance, more than one hundred components. But, of course, specific techniques allowing to deal with the rare event case must be applied. This is the topic of this chapter.

The Crude Monte Carlo technique in this context consists of sampling N times the system configuration, that is, generating independent samples $\vec{X}^{(1)}, \ldots, \vec{X}^{(N)}$ of $\vec{X}$ and estimating the unknown parameter R by the unbiased estimator

$$\widehat{R} = \frac{1}{N} \sum_{n=1}^{N} \Phi(\vec{X}^{(n)}).$$

The evaluation of $\Phi(\vec{x})$ for a given configuration $\vec{x}$ takes the form of a graph exploration. For instance, in the source-to-terminal case, a DFS (Depth First Search) procedure is typically implemented to check if source and terminal are connected in the graph resulting from the initial model when all lines corresponding to the zeros in $\vec{x}$ have been deleted.

The case of interest here is the case of $R \approx 1$, and so, $Q = 1 - R \approx 0$. The rare event is "$\Phi(\vec{X}) = 0$", and the methods used to deal with it are the object of the rest of the chapter. After discussing about the many applications in this area through a literature revision in next section, Section 7.3 describes the main ideas used so far in order to analyze network reliability, focusing in the rare event situation. In Section 7.4 a specific approach is presented in more detail. Section 7.5 presents some numerical examples of the behavior of these techniques. Finally, Section 7.6 concludes the chapter.

7.2 Network reliability applications

There is a wide field of applications of network reliability techniques. We find these problems in evaluations of electrical power networks, transportation systems (specially urban transportation systems, see for instance [56]), interconnection networks (that is, networks connecting processors, memories and other devices inside a multiprocessor computer), fault tolerant computer architectures, etc. As stated before, a central application area is in the evaluation of communication systems. The usefulness of "connectivity" measures such as the ones presented before is clear for instance in packet switching communication networks using dynamic routing which allows rerouting of data in case of the failure of a link. It must be said that many modern packet switching networks are rather dense and that the considered reliability measures tend to be close to one. The computation of the unreliability of the system systematically corresponds then to the evaluation of the probability of a rare event. In this section, we give some applications examples in different contexts, such as in the design of telecommunication networks and other systems [5, 20, 21, 22, 23, 24, 25, 42, 43, 46, 50, 57], in the design and evaluation of mobile ad hoc networks and of tactical radio networks (specially in military contexts) [19], in the evaluation of transport networks and the assessment of the reliability of road networks with respect to seismic hazards and other disasters [34, 44, 49, 61]. The goal of this section is to underline the wide application range of these problems, and thus, of the methods proposed to solve them. Once again, let us recall that in most of the cases, the events of interest are rare.

The design of the topology of telecommunication and computer networks is one of the settings where the application of reliability models is more direct. As such, there are a number of paper which tackle different variants of this problem, which in general consists in deciding which components (links, and sometimes nodes) to include in the network so that the communication among terminals is reliable and the cost is as low as possible. Such network design problems are in general NP-hard, so that most literature includes the use of combinatorial optimization heuristics (most often Genetic Algorithms) to find approximate solutions.

One of the first papers on applying Genetic Algorithms (GA) to solve reliable network design problem was published by Kumar, Pathak and Gupta [42]. These authors tackled three different network design problems: to maximize reliability under diameter constraint, to maximize diameter under degree constraint, and to minimize average distance under degree constraint. The solution method applied was based on a GA, which solved very small (graphs up to 9 nodes) instances of these problems, attaining optimal solutions. Even if the network size tackled was very small, this work showed that GA could be designed to tackle reliable networks design problems.

The papers by Dengiz, Altiparmak and Smith [22, 23] study two variants of reliable network design: maximizing the all-terminal network reliability metric given a cost constraint, and minimizing the cost, given a reliability constraint. The node set is fixed, and the problem consists in choosing which links to install. The problem is solved using an evolutionary approach, based on GA plus a local search heuristic. Reliability is estimated using a specific heuristic, upper bounds, and Monte Carlo simulation. The authors evaluated their algorithm and an exact, branch-and-bound based alternative, using 79 randomly generated small test problems (with 6 to 20 nodes), and the results showed that both algorithms found the optimal solutions, and that the GA was the most computationally efficient. Deeter and Smith [20, 21] also faced the design of networks considering all-terminal reliability. These authors consider minimizing the network cost given a reliability constraint. In their setting, the nodes are given, and it is possible to choose which links to employ, and different "link options", each having different reliability and cost values. A Genetic Algorithm is used to select the links and the level of link connection; Monte Carlo simulation is used to compute estimates of the network reliability. Experiments with different topologies showed the effectiveness of the approach in identifying low cost solutions meeting the reliability requirements. Other more recent work by the same authors includes papers [1] and the GA by Altiparmak et al. [2]. Other authors, like Lin and Gen [43], have also studied the same all-terminal reliability network design problem and proposed alternative optimization methods with improved performances.

Barán and Laufer [5] proposed a parallel Asynchronous Team Algorithm (A-Team) applied to the reliable network design problem, where the nodes and links are fixed, but it is possible to choose (at a cost) a given reliability value for each link. A-Team is a hybrid technique that combines different algorithms interacting to solve the same global problem. Two approaches were used to estimate network reliability in this paper: an upper bound of all the candidates included in the population is efficiently calculated, and after that, a Monte Carlo simulation is used to get good approximations of the all-terminal reliability. The empirical results show good values for medium-size networks. Duarte and Barán [24]) addressed a multiobjective version of the previous problem, using a parallel asynchronous version of a Genetic Algorithm to search for optimal topologies for a network. The parallel version results outperform the sequential ones, considering standard metrics in the multiobjective domain (where the solution is not just a topology, but a set of Pareto efficient ones). Later, Duarte, Barán, and Benítez [25] published a comparison of several parallel multiobjective EAs for solving the same reliable network design problem.

Taboada, Baheranwala and Coit [57] and Taboada, Espiritu and Coit [58] also look at multiple-objective system reliability design problems, where it is necessary to decide the level of redundancy to allocate at each stage, and reliability, cost and weight are objective functions. In these papers different methodologies are explored; on one side, to help the decision maker make a selection, a pseudo-ranking scheme and clustering techniques to reduce the size of the Pareto optimal set are presented. The second paper presents a multiple objective genetic algorithm for solving the problem.

Marseguerra, Zio, Podofillini and Coit [46] used a stochastic model for network reliability, considering a function of imperfectly known reliability parameters of network components. The problem to solve is again a multiobjective one, the objective being to find the network topologies that maximize the network reliability and minimize the variance of this estimation (taking into account the imperfectly known reliability parameters). The decision variable is the type and the redundancy level of components to be allocated within a fixed network topology, where each component has an associated reliability probability distribution. The optimization method is based on GA, and a Monte Carlo evaluation algorithm is used to incorporate the uncertainty in the reliability values; the repeated evaluations of the good individuals are accumulated, to enhance the significance of the estimations. The numerical examples consider only very small networks (with 7 and 8 links), and allow to examine the Pareto optimal solutions obtained and to easily identify the differences in the configurations.

Premprayoon and Wardkein [50] tackle another variant of the reliable network design problem, where it is possible to define for each pair of nodes whether they will be connected by a link, whose characteristics (cost and reliability) can also be chosen from a given set. The objective is to minimize network cost subject to a requirement of attaining at least a given reliability level. The authors compare an Ant Colony Optimization method, a Tabu Search method and a Local Search method; the network reliability evaluation is done by backtracking (as only very small network topologies are studied). The computational results show better results for the Ant Colony Optimization.

Cook and Ramirez-Marquez [19] study mobile ad hoc wireless networks (MAWN), in particular in a military context. MAWN have their own characteristics, which this work describes, and a proposal is presented on how to adapt the classical analysis of network reliability to this new context. The methods proposed rely on considering the effect of node mobility and the continuous changes in the network's connectivity. Wakabayashi [61] also studies highway network reliability, takint into account normal and abnormal periods. The motivation of this work is to detect the critical link on the network, whose improvement will give the larger network reliability improvement. This paper presents a comparative study between using a probability importance index (Birnbaum's structural importance) and a criticality importance based on network reliability measures, which address some problems in the previous mentioned index.

Yan Li [44] proposes to employ an Accessible Node Rate (ANR) index based on two-terminal reliability to evaluate the anti-disaster level of a city road network. In particular, this author evaluates the connectedness to the start points of emergency vehicles, and provides a method of accessing such accessibility to individual residences in a real city by using GIS. Nojima [49] employs network reliability models to represent the risks on road networks caused by seismic activity. The performance measure of interest is defined as the system flow capacity of road networks subject to failures. In this paper, a variance reduction technique for Monte Carlo simulation method is presented to perform efficient reliability analysis in terms of the system flow capacity. This method is used to define performance-based prioritization order; this results in a road prioritization strategy according to various levels of vulnerability and system requirement. Günnec and Salman [34] propose to assess the post-disaster performance of a road network under most likely disaster scenarios for the purpose of both strengthening the components of the network and for planning the post-disaster logistics activities. In this paper, the authors seek to measure the reliability and the expected post-disaster performance of a network under disaster risk. In particular, they evaluate the reliability of connection between different pairs of origin-destination (O-D) nodes in the network, in terms of expected weighted sum of shortest travel time/distance between the O-D pairs. The estimation of this measure is done by Monte Carlo sampling.

7.3 Variance reduction techniques

There are many generic variance reduction techniques that have been proposed in order to improve the performance of Monte Carlo simulations, especially for the rare event case. Importance sampling is probably the most used one, but we also find techniques based on antithetic variates, on control variables, stratified sampling, etc. These techniques can be also applied to network reliability evaluation, with varying degrees of performance. Nevertheless, the special characteristics of this problem opens the opportunity to develop more specialized variance reduction methods, sometimes inspired by the classical ones and sometimes completely original, which attain much better performances. As many methods have been proposed, it is not possible to describe each of them in detail. In this section, we will briefly present the main ideas which have appeared in the literature, and we will reference the publications which fall in the same broad categories. We also give an assessment about the most promising approaches.

7.3.1 Sampling techniques based on bounds

This family of methods can be interpreted as an hybrid of classical Importance Sampling with control variates. Its first application to network reliability problems was presented by Van Slyke and Frank [60], and afterwards by Kumamoto [40] and Fishman [30]. It can be applied to any reliability evaluation problem where there are available two functions Φ^L and Φ^U which bound below and above the system structure function Φ. These functions must fulfill the following properties:

- $\Phi^L(\vec{x}) \leq \Phi(\vec{x}) \leq \Phi^U(\vec{x})$ for any state vector $\vec{x}$.
- For $k = 1, \dots, |\mathcal{E}|$ and for any value assignment $\tilde{x}^{(k)} = (\tilde{x}_1, \dots, \tilde{x}_k)$ of the first k components of the state vector $\vec{x}$, the values

  $$R_k^L(\tilde{x}^{(k)}) \equiv \mathbb{P}(\Phi^L(\vec{X}) = 1 \,|\, X_1 = \tilde{x}_1, \dots, X_k = \tilde{x}_k)$$

  and

  $$R_k^U(\tilde{x}^{(k)}) \equiv \mathbb{P}(\Phi^U(\vec{X}) = 1 \,|\, X_1 = \tilde{x}_1, \dots, X_k = \tilde{x}_k)$$

  can be computed in polynomial time.

The numbers R_0^L and R_0^U are defined by $R_0^L = \mathbb{P}(\Phi^L(\vec{X}) = 1)$ and $R_0^U = \mathbb{P}(\Phi^U(\vec{X}) = 1)$. For the bound based sampling, we define the remaining state space

$$W = \{\vec{x} : \ \Phi^L(\vec{x}) = 0, \ \Phi^U(\vec{x}) = 1\}$$

from where the samples will be chosen proportionally to their probability in the original state space. From the estimator obtained there and the previous information, we construct an estimator of the system reliability. The variance reduction attained is directly proportional to the fraction of the total probability that is included in the subspace W. We give now a more detailed description of the sampling routine $\mathcal{M}$ for the bound based sampling:

```
Input: network $\mathcal{G}$, terminal set $\mathcal{K}$, $\Phi^L$ and $\Phi^U$
Output: an estimator of $R$ ($\mathcal{K}$-terminal reliability)

Procedure $\mathcal{M}$ :
        Sample $\tilde{X}$; result: $\tilde{x} = (\tilde{x}_1, \dots, \tilde{x}_{|\mathcal{E}|})$
```

```
Compute $R = R_0^L + \Phi(\tilde{x})(R_0^U - R_0^L)$
Return $R$
```

The sample $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_{|\mathcal{E}|})$ is chosen by sampling succesively, for $l = 1, \ldots, |\mathcal{E}|$, the state $\tilde{X}_l$ of link l following the Bernoulli distribution with parameter

$$\begin{aligned}\tilde{r}_l &= \mathbb{P}(X_l = 1 \mid X_1 = \tilde{x}_1, \ldots, X_{l-1} = \tilde{x}_{l-1} \text{ and } \Phi^U(\vec{X}) = 1, \Phi^L(\vec{X}) = 0) \\ &= \left[\frac{R_l^U(\tilde{x}^{(l-1)}) - R_l^L(\tilde{x}^{(l-1)})}{R_{l-1}^U(\tilde{x}^{(l-1)}) - R_{l-1}^L(\tilde{x}^{(l-1)})} \right] r_l.\end{aligned}$$

The variance of each sample of the methods of this family is:

$$\begin{aligned}\text{Var} &= R(R_0^U - R) - R_0^L(R_0^U - R) \\ &= R(1-R) - (1 - R_0^U)R - R_0^L(R_0^U - R)\end{aligned}$$

which is lower than the crude Monte Carlo one ($R(1-R)$). The difference will depend on the tightness of bounds Φ^L and Φ^U. The execution time performance depends on the computational complexity of the evaluation of these bounds.

In [27] a variant of this family is proposed, for the case where all links have the same elementary reliability. The method is based on the efficient computation of a lower bound of the network reliability, based on the evaluation of a subset of states with number of failed links less than the cardinal of the smallest minimal cutset of the network, and it does not employ any upper bound. The variance reduction obtained with this method is of order $1/\lfloor 1 - R_0^L \rfloor$, for a computational cost per sample similar to the crude Monte Carlo one. The authors classify their method within the antithetic variates based family.

7.3.2 Dagger sampling and other related techniques

Dagger sampling was proposed by Kumamoto et al. [41] and can be seen as an extension of the antithetic variates technique. The main idea behind this sampling method is the generation of sample blocks of size L such that within each block the random variables are chosen in order to induce negative correlations between the individual samples. The size of the blocks, L, is fixed in such a way that for each edge l the sequence of L replications can be partitioned in exactly N_l sub-blocks of size L/N_l, where $N_l = \lfloor 1/q_l \rfloor$. For each of these sub-blocks, a single position is randomly chosen; this position corresponds to a sample where link l will fail. In this way, the failure pattern is such that the sampled failure frequency for each link of the network is proportional to the link unreliability value. After all random variables have been sampled, the method checks every replication within the block, checking in each case whether the resulting network is connected or not, in order to obtain an estimation of R.

In this method, a single invocation of $\mathcal{M}$ corresponds to L crude Monte Carlo samples; this must be taken into account when comparing the computational complexity of the algorithms.

```
Input: network $\mathcal{G}$, terminal set $\mathcal{K}$
Output: an estimator of $R$ ($\mathcal{K}$-terminal reliability)

Initialization $\mathcal{I}$:
            Compute the integer vector $(N_l : l \in \mathcal{E})$: $N_l = \lfloor 1/q_l \rfloor$.
            Choose the sample number: $L = \mathrm{lcm}\{N_l : l \in \mathcal{E}\}$.

Procedure $\mathcal{M}$:
      For each link $l$
         For each sub-block of replications of size $L/N_l$,
            sample random state vectors $\vec{X}^{(j)}$ in this sub-block:
            Choose randomly a replication from the sub-block :
                 Sample $U$ uniformly on [0,1]; result: $u$; set $k = \lceil u/q_l \rceil$
            Link $l$ is failed in that replication:
                 $X_l^{(j)} = 1$, $\forall j \neq k$; $X_l^{(k)} = 0$, if $k \leq L/N_l$.
         endFor
      endFor
      Initialize $T = 0$.
      For $i$ from 1 to $L$
         Count  replications corresponding to an operational state of the network:
              $T = T + \Phi(\vec{X}^{(i)})$.
      endFor
      Return $R = T/L$.
```

The complexity analysis of this algorithm, given in [29], shows that the execution time per sample has worst case complexity $O(|\mathcal{E}|)$, as in the case of crude Monte Carlo. Nevertheless, as the number of random variables that are needed in the Dagger method is much smaller than in the crude Monte Carlo, there can be a large gain in execution time, which is an important advantage of the method in relation to the crude one.

The variance reduction obtained by the Dagger method is based on the induced negative correlation among the samples which belong to the same sub-block for some link of the network, and is relatively small, approaching 0 when the link reliability is near 1. El Khadiri and Rubino in their work [26] discuss some problems of the Dagger method, which happen in particular when value L is too large. As it is necessary to generate and save into memory L states of the network, memory requirements grow linearly with L. The authors show that the block size L may be chosen arbitrarily, and they propose an alternative method, inspired on Dagger and applying a generalization of the standard antithetic method. This new method obtains better results, both because L can be chosen arbitrarily, and because the sampling algorithm used incorporates some features to improve its efficiency. The algorithm employs mechanisms similar to those used in discrete event simulation, employing a list with the incumbent failure instants for each link, and finding which is the first replication including at least one failure. This implementation has important gains in execution time and also results in a minimal memory complexity (which depends linearly on the network size and is independent of L).

When all links have the same reliability p, the generalized Antithetic algorithm has sample complexity $O(\alpha|\mathcal{E}|)$, where

$$\alpha = \max(1/L, 1-p).$$

This shows an improvement over the crude method which grows with the block size L, up to a bound given by the inverse of the link reliabilities (in the worst case, $p = 0.5$, we have that the maximum gain that can be attained is a factor of 2, corresponding to setting $L = 2$).

7.3.3 Graph evolution models

These methods, instead of using the static network reliability model, employ alternative timed models where the states of the links are assumed to change over time. These models correspond to Markov processes, whose properties can then be exploited to obtain efficient estimators of the classical reliability.

Easton and Wong [47] proposed the Sequential Construction and Sequential Destruction methods, which complement the previous idea with the use of an ordering of the network links. In the Sequential Construction method, all links are considered as being failed at an initial instant, and then they are successively repaired, one by one following the ordering chosen, until the network reaches an operative state. The reliability estimator can be interpreted as a function of the expectation of how much time is needed to reach an operative state of the network. The Sequential Destruction is similar, but all links are considered to be operating at the initial time, and they are progressively put into a failure state. These techniques can be classified as hybrids between stratified sampling and Importance Sampling procedures.

The sample space for the Sequential Construction method consists in pairs $(\tilde{x}, \tilde{\pi})$ where $\tilde{x}$ is a state vector or configuration and $\tilde{\pi} = (\tilde{\pi}_1, \dots, \tilde{\pi}_{|\mathcal{E}|})$ is a permutation of the link indexes in $\mathcal{E}$. There exists an index k such that

$$\tilde{x}_{\tilde{\pi}_1} = \ldots = \tilde{x}_{\tilde{\pi}_k} = 1, \;\; \tilde{x}_{\tilde{\pi}_{k+1}} = \ldots = \tilde{x}_{\tilde{\pi}_{|\mathcal{E}|}} = 0.$$

If we choose a vector $\tilde{x}$ following the system state probabilities (i.e, with the same distribution as $\vec{X}$) and we choose a permutation $\tilde{\pi}$ independently and uniformly over all the compatible permutations, then the probability of observing a given pair $(\tilde{x}, \tilde{\pi})$ is

$$\rho(\tilde{x}, \tilde{\pi}) = \frac{\mathbb{P}(\vec{X} = \tilde{x})}{k!(|\mathcal{E}| - k)!} = \frac{1}{|\mathcal{E}|!} C_k^{|\mathcal{E}|} \mathbb{P}(\vec{X} = \tilde{x})$$

where k is the number of links working in $\tilde{x}$. The Sequential Construction method samples $\tilde{\pi}$, and considers simultaneously the set $\mathcal{P}_{\tilde{\pi}}$ of all possible pairs $(\tilde{x}, \tilde{\pi})$, such that $\tilde{x}$ is consistent with $\tilde{\pi}$ following the previous criterion. The reliability estimator R is then the conditional probability of operation of the system given $\mathcal{P}_{\tilde{\pi}}$, corresponding to the quotient of the sum of the probabilities of the pairs $(\tilde{x}, \tilde{\pi}) \in \mathcal{P}_{\tilde{\pi}}$ such that $\Phi(\tilde{x}) = 1$ divided by the probability of $\mathcal{P}_{\tilde{\pi}}$. We give below a more detailed description of the algorithm $\mathcal{M}$ for generating a Sequential Construction sample (no initialization is needed):

Input: network $\mathcal{G}$, terminal set $\mathcal{K}$
Output: an estimator of R ($\mathcal{K}$-terminal reliability)

```
Procedure $\mathcal{M}$ :
        Sample $\tilde{\pi} = (\tilde{\pi}_1, \dots, \tilde{\pi}_{|\mathcal{E}|})$
        For $k = 1, \dots, |\mathcal{E}|$ (Define $\tilde{x}^{(k)}$)
            $\tilde{x}^{(k)}_{\tilde{\pi}_1} = \dots = \tilde{x}^{(k)}_{\tilde{\pi}_k} = 1,\ \tilde{x}^{(k)}_{\tilde{\pi}_{k+1}} = \dots = \tilde{x}^{(k)}_{\tilde{\pi}_{|\mathcal{E}|}} = 0$
        endFor
        Determine first $r \in 0, \dots, |\mathcal{E}|$
                such that $\Phi(\tilde{x}^{(r)}) = 1$
        Compute
        $R = \frac{\sum_{k=0}^{|\mathcal{E}|} \Phi(\tilde{x}^{(k)}) \rho(\tilde{x}^{(k)}, \tilde{\pi})}{\sum_{k=0}^{|\mathcal{E}|} \rho(\tilde{x}^{(k)}, \tilde{\pi})} = \frac{\sum_{k=r}^{|\mathcal{E}|} C_k^{|\mathcal{E}|} \mathbb{P}(\vec{X} = \tilde{x}^{(k)})}{\sum_{k=0}^{|\mathcal{E}|} C_k^{|\mathcal{E}|} \mathbb{P}(\vec{X} = \tilde{x}^{(k)})}$
        Return $R$.
```

It is possible to show that the estimator obtained by this method has smaller variance than the one corresponding to a crude Monte Carlo sample. The larger computational effort is needed to determine the first index r, which depends on the effort needed to compute $\Phi(\tilde{x}^{(k)})$, as it is necessary to determine when the network arrives to an operational state when the links are repaired one by one. In the worst case, and using a DFS for computing Φ, the complexity is $O(|\mathcal{E}| \max(|\mathcal{V}|, |\mathcal{E}|))$. Nevertheless, for the case of the 2–terminal metric, it is possible to determine the value of r with a computational cost similar to a single computation of Φ, so that this method obtains a single sample with a cost similar to the one of the crude method.

As mentioned previously, the Sequential Destruction method is very similar; in the case of very low reliability systems, it may have better computational performances, as the value of r would be determined in less iterations than those needed by the Sequential Construction method.

Other methods based in graph evolution models have been published in [28]. In particular, three methods are discussed in that work: *destruction processes*, *construction processes*, and *merge processes*. All the methods rely on constructing a Markov chain $(\vec{Y}(t))$ such that at time $t = 1$ we have that $\mathbb{P}(\Phi(\vec{Y}(1)) = 1) = \mathbb{P}(\Phi(\vec{X}) = 1)$, so that computing the expectation of $\Phi(\vec{Y}(1))$ gives also the reliability R. Then, the algorithm samples a permutation of the order at which links go up (in a creation or merge process) or go down (in destruction process), it identifies the critical link (the one which causes the system to change from down to up state, or vice-versa), and it exactly computes (using a convolution of exponential r.v.) the exact conditional probability that at time 1 the critical link will have changed its state. In the case of merge processes, they improve the previous ideas identifying irrelevant links and partitions of the subjacent network. In these methods, the sample complexity is higher, of order $O(|\mathcal{E}|^2)$, but the variance is much smaller than the one obtained by the Sequential Construction method. An important result is that, for fixed $\mathcal{E}$, the merge processes method has coefficient of variation uniformly bounded for all values of links' reliabilities. Anoher related work is [35], which gives an hybrid variant of crude Monte Carlo and graph evolution models, completed with the use of Importance Sampling to further speed-up the simulations.

More recently, Hui et al. [36] have applied Cross-Entropy techniques to improve the performances of crude Monte Carlo and of graph evolution methods, in particular creation processes and merge processes. The main idea is to apply an Importance Sampling scheme, changing the underlying network reliability parameters, and to use Cross-Entropy to search for an optimal change of measure. Their results show that Cross-Entropy does indeed give better accuracy; the improvement is quite large over crude Monte Carlo. In the case of construction and merge processes based Monte Carlo, the application of Cross-Entropy results in much more modest improvements. Similar results have been obtained by Murray and Cancela [48], who compared the behavior of these methods (and of a generalized antithetic method) when evaluating the diameter constrained reliability of a network, a variant of the classical model taking into account a bound on path lengths).

Finally, a quite different approach to exploit the Markov process modeling a creation process of the network has been employed by Murray et al. [15]. This work applies the well-know splitting technique (see Chapter 3), much employed for rare-event simulation in the context of stochastic processes, to the stochastic process consisting in starting from an empty network, and creating (or putting into operational states) the links one by one, taking independent exponential distributions for these times. As it was mentioned before, the state of this system at time $t = 1$ has the same distribution as the state of the static network model; in a highly reliable network, almost always the network becomes operational before time 1, and the rare event is to observe $\Phi(\vec{Y}(1)) = 0$. The splitting strategy developed in the mentioned work consists in taking a number of intermediate time thresholds, and of splitting such trajectories of process $(\vec{Y}(t))$ that at these thresholds still verify $\Phi(\vec{Y}(t)) = 0$. The results given by the mentioned authors show that this method is very robust and can reach better performances than the one by Hui et al. [36].

Last, let us also mention [37], where the authors propose to directly estimate the reliability ranking of some edge relocated networks without estimating their reliabilities and compare the proposed approach to the traditional approach using the Merge Process estimation algorithm. Another recent related paper is [39], which is concerned with network planning. Here, the objective is to maximize network's reliability, subject to a fixed budget. The authors show how the Cross Entropy method can be easily modified to tackle the noise brought by the use of network reliability estimators in the objective function instead of exact evaluations.

7.3.4 Coverage method

The Coverage method was proposed by Karp and Luby [38]. It can be seen as an hybrid variant of Importance Sampling and Stratified Sampling, and employs the list of the minimal cuts of the system to improve the crude sampling procedure. The main idea is to embed the set F of network failure events within an *universal weighted space* $(\mathcal{U}, w)$, where w is a non-negative weighting function in $\mathcal{U}$, satisfying the following criteria:

- $w(F) = \mathbb{P}(F) = Q$.
- $w(\mathcal{U})$ can be efficiently computed (in polynomial time). Moreover, it is possible to efficiently sample values in $\mathcal{U}$ with probability proportional to the weights of the elements of the set.
- It can be efficiently decided whether an element of $\mathcal{U}$ belongs to F.
- $w(\mathcal{U})/w(F)$ is bounded above by a value M for all the instances in the considered problem class.
- $w(C)$ is the total weight of the elements of $\mathcal{U}$ with second component equal to C.

If we take a sample from $\mathcal{U}$, and we obtain the estimator $\hat{Q}$ multiplying the proportion of elements of this sample that are included in F by $w(\mathcal{U})$, then $\hat{Q}$ is an unbiased estimator of Q.

Let $\mathcal{C}$ be the set of the $\mathcal{K}$-mincuts of network $\mathcal{G}$. We define the universal weighted sample space $\mathcal{U}$ composed of the pairs $(\vec{x}, C)$ where $\vec{x}$ is a state vector of the network, $C \in \mathcal{C}$ is a cut, and $x_l = 0$ for all links l belonging to C. This way every system failure state $\vec{x}$ will appear in $\mathcal{U}$ as many times as the number of failed mincuts in $\vec{x}$; in order to embed F in $\mathcal{U}$ it is necessary to assign to each $\vec{x}$ a single cut $C \in \mathcal{C}$. To do this we choose a node $s \in K$, then we find the set N of all nodes reachable from s following paths formed by operational links, and we select $C \equiv C(\vec{x})$ the set of links from N to $\mathcal{V} - N$. The elements of F appear then in $\mathcal{U}$ as pairs $(\vec{x}, C)$ such that $C = C(\vec{x})$, and it can be decided in linear time if an element from $\mathcal{U}$ belongs to F just by verifying the condition $C = C(\vec{x})$. The weighting function w is given by $w(\vec{x}, C) = \mathbb{P}(\vec{X} = \vec{x})$.

We give now a pseudocode of the initialization and sampling routines for the Coverage method:

Input: network $\mathcal{G}$, terminal set $\mathcal{K}$, list of mincuts $\mathcal{C}$
Output: an estimator of R ($\mathcal{K}$-terminal reliability)

```
Initialization $\mathcal{I}$:
    For each $C \in \mathcal{C}$
        Compute $w(C) = \prod_{l \in C} q_l$
    endFor
    Compute $w(\mathcal{U}) = \sum_{C \in \mathcal{C}} w(C)$

Procedure $\mathcal{M}$:
        Sample $(\vec{X}, C)$ from $\mathcal{U}$ with distribution $w$:
        Sample a cut $C$ with probability $w(C)/w(\mathcal{U})$
        Build $\vec{X}$:
            $\forall l \in C, X_l = 0$
            $\forall l \notin C$, $X_l = 1$ with probability $p_l$,
                    $X_l = 0$ otherwise.
        If $C = C(\vec{X})$
          $R = 1 - w(\mathcal{U})$
        Else
            $R = 1$
        endIf
        Return $R$.
```

This method can obtain good variance reduction levels, but has the drawback of depending on the previous calculation and storage of the list of all mincuts of the considered network. As the size of this list grows exponentially with the size of the network, the requirements of time and space can quickly make unfeasible its application. Also, there are some exact methods that compute the reliability in time polynomial in the number of mincuts of the network [52], further reducing the interest of the approach.

7.3.5 State space partitioning and conditioning methods

A number of methods are based on sampling within the space of the state vectors of the network, using techniques related to partitioning this space and/or to conditionally sampling within it.

One of these methods is the Total Hazard one. Random hazard variables, and in particular the total hazard ones, have been employed in different contexts to simulate stochastic models [53]. Ross and Jun [54] developed a total hazard estimation to compute the reliability R.

Let C_1 be a $\mathcal{K}$-mincut. The first hazard, h_1, is the probability that all the components in C_1 are failed (implying that the network is not $\mathcal{K}$-connected); so,

$$h_1 = \prod_{i \in C_1} q_i.$$

The Total Hazard method consists of simulating the state of all the links belonging to C_1. If all the links are failed, the procedure ends. If at least one link is operational, we fix the states of the simulated links, and we look for a new mincut C_2 in the modified network. From this new cut we compute the second hazard, h_2, given by

$$h_2 = \prod_{i \in C_2} q_i.$$

Afterwards we simulate the state of the links belonging to C_2 and the previous process is repeated, generating new networks until all the components of a mincut are failed or until a trivial network is reached, with no mincuts (all the links' states have been fixed). If we considered r hazards, the total hazard is given by

$$H(\mathcal{G}) = \sum_{i=1}^{r} h_i,$$

and it is an unbiased estimator of Q. The implementation suggested in [53] employs the list of all mincuts of the system under consideration, and it updates it as the states of the links are fixed; the mincut is chosen at each step in order to result in the maximum risk.

```
Inputs: network $\mathcal{G}$, set $\mathcal{K}$, list of $\mathcal{K}$-mincuts of $\mathcal{G}$
Output: an estimation for $R$

Inicialization$\mathcal{I}$:
            $H = 0$.

Procedure $\mathcal{M}$:
        Select a $\mathcal{K}$-mincut $C$.
        Simulate the state of the links in $C$
        Repeat until all the links in the selected mincut are failed
            Update the list of mincuts of network $\mathcal{G}$.
            Compute the hazard: $h = \prod_{i \in C} q_i$.
            Accumulate in $H$: $H = H + h$.
            Select a $\mathcal{K}$-mincut $C$.
            Simulate the state of the links in $C$.
        endRepeat
        Return $R = 1 - H$.
```

The variance reduction that can be obtained with this method depends strongly on how the $\mathcal{K}$-mincut C is chosen. A heuristic with good behavior is to select in each step the mincut with the highest associated hazard, but this implies an important computational overhead, as it must be implemented as a search in the list of mincuts (whose size is exponential in the size of the graph), or employing a maximal flow algorithm in each iteration of the method. The analysis of the computational complexity of generating a sample strongly depends on this step. With the computationally less costly possible choice for the mincut selection, the computational complexity per sample is of order $O(|\mathcal{E}|)$.

In [9], Cancela and El Khadiri highlighted that there are some cases where the Total Hazard estimator is less efficient than the CMC one. They proposed a modification leading to a more precise estimator and having always variance lower than that of the CMC method.

7.4 The RVR sampling principle

Another family of related methods are the Recursive Variance Reduction (RVR) ones. These methods, first proposed in [8], have been extensively discussed and adapted to different contexts [10, 11, 12, 13, 14, 16, 17]. RVR methods combine different ideas to obtain good performance estimators. On one hand, they employ either one or more cutsets, pathsets, or both, of the network being evaluated, in order to partition the state vector space in subsets depending on the operational/failed status of the links belonging to the chosen sets. Some of the elements of this partition correspond to network configurations known in advance (corresponding either to a failed or an operational network). Then, the rest of the state vector space is explored by recursively sampling one of the subsets in the partition, which corresponds to a subnetwork of the original one, including some particular links failed and other ones operational; once this subnetwork has been randomly chosen, an RVR method recursively searchs for new cutsets and/or pathsets, and restarts the whole process.

In this section we provide some details about the RVR approach. For the presentation of the RVR principle, we consider the 2-terminal problem where we look at the unreliability between two given nodes s and t in $\mathcal{G}$, the version using series–parallel simplification for reducing the size of the network and a selected cut for transforming a network reliability problem into a smaller one and then recursively until the considered network has unreliability equal to 0 or to 1. This estimator has been proposed in [11].

If s and t are not connected in $\mathcal{G}$, we define $Z(\mathcal{G}) = 1$. Otherwise, let us denote by sp-red($\mathcal{G}$) the result of making all possible series-parallel reductions in $\mathcal{G}$. As these reductions preserve the unreliability, we set $Z(\mathcal{G}) = Z(\text{sp-red}(\mathcal{G}))$. Let γ be a st-cut in sp-red($\mathcal{G}$), $\gamma = \{l_1, l_2, \cdots, l_H\}$ where l_1, l_2... are the links in the cut. Let L_h be the event "link l_h is down". If Ω is the set of all possible configurations in the model, consider the partition $\Omega = (E_0, E_1, \ldots, E_H)$ where

$E_0 = L_1 L_2 \cdots L_H =$ all links in γ are down,

$E_1 = L_1^c =$ at least one link in γ is up, and the first such link is l_1,

$E_2 = L_1 L_2^c$ = at least one link in γ is up, and the first such link is l_2,

$E_3 = L_1 L_2 L_3^c$ = at least one link in γ is up, and the first such link is l_3,

...

$E_H = L_1 L_2 \cdots L_H^c =$ at least one link in γ is up, and the first such link is l_H.

We have $\mathbb{P}(E_0) = q_1 \cdots q_H$ where here we denote $q_h = \mathbb{P}(L_h) = 1 - r_h$, and $\mathbb{P}(E_h) = q_1 q_2 \cdots q_{h-1} r_h$ for $h = 1, 2, \cdots, H$. To simplify the notation, call π_h the product $\pi_h = q_1 q_2 \cdots q_h$ for $h = 1, 2, \cdots, H$, $\pi_0 = 1$. We have $\mathbb{P}(E_0) = \pi_H$ and $\mathbb{P}(E_h) = \pi_{h-1} r_h$ for $h = 1, 2, \cdots, H$.

Let I be the r.v. "index in γ of the first link up", with $I = 0$ if all links in π are down. We have $\mathbb{P}(I = h) = \mathbb{P}(E_h)$. Define now the r.v. V on $\{1, 2, \cdots, H\}$ by

$$\mathbb{P}(V = h) = \mathbb{P}(I = h \mid I \neq 0) = \frac{\pi_{h-1} r_h}{1 - \pi_H}.$$

Last, for $h = 1, 2, \cdots, H$, denote by $\mathcal{G}_h = \text{sp-red}(\mathcal{G}) \mid E_h$ the network obtained from sp-red$(\mathcal{G})$ by deleting links $l_1, l_2, \cdots, l_{h-1}$ and contracting link l_h. We are now ready to give the estimator proposed in [11]:

$$Z(\mathcal{G}) = \pi_H + (1 - \pi_H) \sum_{h=1}^{H} \mathbb{1}(V = h) Z(\mathcal{G}_h).$$

Let us denote by v a sample from the distribution of V. Then a sample $Z^{(k)}(\mathcal{G})$ of $Z(\mathcal{G})$ can be deduced from a sample $Z^{(k)}(\mathcal{G}_v)$ of $Z(\mathcal{G}_v)$ by

$$Z^{(k)}(\mathcal{G}) = \pi_H + (1 - p_H) Z^{(k)}(\mathcal{G}_v).$$

If s and t are merged into final single node, the unreliability of $\mathcal{G}_v$ is equal to 0 and then $Z^{(k)}(\mathcal{G}) = \pi_H + (1 - p_H) \times 0 = \pi_H$. If s and t are not connected, unreliability of $\mathcal{G}_v$ is equal to 1 and then $Z^{(k)}(\mathcal{G}) = \pi_H + (1 - p_H) \times 1 = 1$. Otherwise, we found an st-cut in $\mathcal{G}_v$ and we proceed again as before. The main interest of this procedure is that $\mathcal{G}_v$ is smaller than $\mathcal{G}$, and sometimes much smaller, because of the series-parallel simplifications, deletions and contractions operations.

A function which returns a trial of $Z(\mathcal{G})$ can be summarized as follows:

```
TRIAL-RVR($\mathcal{G}, K$)
    1. Check end recursion condition:
        If $|K| = 1$ return(0)
        If $\mathcal{G}$ is not $\mathcal{K}$-connected return(1)
    2. Construct sp-red$(\mathcal{G})$ by applying series-parallel reductions to $\mathcal{G}$
    3. Find a $\mathcal{K}$-cut $\pi$ in sp-red$(\mathcal{G})$: $\pi = \{l_1, \ldots, l_H\}$
    4. Compute the probability $\pi_H$ that all links in $\pi$ are down
    5. Compute the p.m.f. distribution of the r.v. $V$
    6. Generate a trial $v$ of $V$
    7. Construct the network $\mathcal{G}_v$ = sp-red$((\mathcal{G} - l_1 - l_2 - \cdots - l_{v-1}) * l_v)$
    8. Recursive step: return($\pi_H + (1 - \pi_H) \times$ Trial-RVR$(\mathcal{G}_v, K_v)$).
```

The memory space complexity of the function TRIAL-RVR$(\mathcal{G}, K)$ is of order $O(|\mathcal{E}|(|\mathcal{E}| + |\mathcal{V}|))$ and time complexity is, in the worst case, of order $O(|\mathcal{E}|(|\mathcal{K}|^2 |\mathcal{E}||\mathcal{V}|^2))$. The worst case corresponds to a version using a maximal flow procedure in order to select a $\mathcal{K}$-cut π at step 3 of the above algorithm.

By calling N times the function TRIAL-RVR($\mathcal{G}, K$), we obtain N independent trials $Z^{(k)}(\mathcal{G})$ of $Z(\mathcal{G})$, $1 \leq k \leq N$. The sample mean $\widehat{Z}(\mathcal{G})$ of these trials leads to an estimate of $Q = 1 - R$ and the variance is estimated by

$$\widehat{V}_{RVR} = \frac{1}{N(N-1)} \sum_{k=1}^{N} \left(\widehat{Z}(\mathcal{G}) - Z^{(k)}(\mathcal{G}) \right)^2 .$$

In [12] it is shown how the computational complexity of the RVR method can be improved by generating simultaneously the N samples, and in that way avoiding replicating a large part of the computations, and [13] discusses the sensitivity of the RVR's accuracy to the strategy of choosing cuts. Instead of using cuts for recursively changes the original problem into smaller one, the method in [10] exploits paths and the one in [14] exploits both paths and cuts leading to a more interesting behavior than the versions based on only paths or cuts.

7.5 Numerical Results and conclusions

To the best of our knowledge, the CE-MP method [36] which uses the Cross-Entropy technique to further improve the performances of the Merge Process method [28] and the RVR technique which exploits series-parallel reductions and a minimum cost s, t-cut strategy [13], where each link l is valued by $-\ln(q_l)$, are the more adapted procedures published in the literature to compute network reliability in a rare event context. In this section we present some numerical illustrations of these methods.

Network	common link unreliability q	Q
G_3	10^{-3}	4.01199×10^{-06}
G_3	10^{-6}	4.00001×10^{-12}
G_6	10^{-3}	4.00800×10^{-06}
G_6	10^{-6}	4.00001×10^{-12}

Table 7.1: Exact unreliabilities of grid networks (see Figure 7.1) used for numerical illustrations [36].

For the examples, we consider highly reliable grid topologies G_3 and G_6 (see Figure 7.1), where links are assigned equal unreliability $q = 10^{-3}$ or $q = 10^{-6}$ as in [36] and $\mathcal{K}$ is the set of the four corner nodes. For those networks exact values of $Q = 1 - R$ are tabulated at column 3 of Table 7.1. Each exact unreliability Q serves in the computation of the relative error parameter which helps to appreciate the quality of the estimates produced by the two considered estimators. Tables 7.2 and 7.3 show that both methods lead to small relative errors and the RVR method offers more accurate estimates. In the general case, we do not know the exact values. Then, the best estimator in terms of accuracy is the one having smaller variance for a fixed sample size N, leading to smallest lengths of confidence intervals. Column 6 of Table 7.3 shows that the RVR

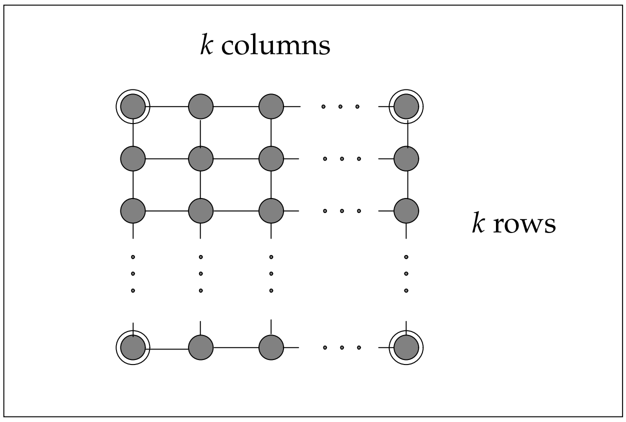

Figure 7.1: G_k: the grid network topology. The four corner nodes are the terminals.

Network	q	Estimate of Q [36]	RE (%)	Variance[36]
G_3	10^{-3}	4.01172×10^{-06}	6.73×10^{-03}	1.84515×10^{-17}
G_3	10^{-6}	3.99876×10^{-12}	3.12×10^{-02}	1.85116×10^{-29}
G_6	10^{-3}	4.00239×10^{-06}	1.40×10^{-01}	3.74067×10^{-17}
G_6	10^{-6}	3.99869×10^{-12}	3.30×10^{-01}	3.75850×10^{-29}

Table 7.2: Performances of the MP-CE method for the evaluation of G_3 and G_6. Terminals are the four corner nodes and $N = 10^6$.

method reduces significantly the variance with respect to the CE-MP one and the best gains are obtained for highly reliable cases.

For illustrating the behavior of the RVR method on dense networks, let us now consider the evaluation of complete topologies for which we calculate the exact values of Q by a Maple program and we tabulate them at the column 2 of Table 7.4. We consider a sample size $N = 10$ for each network. The estimates obtained by the RVR method are given in column 3, associated relative errors and variances are in column 4 and 5 respectively. At column 6, we give the variance-gains when RVR method is compared to the Crude Monte Carlo one. The CMC's variance is equal to $Q(1 - Q)/N$. We can

Network	q	Estimate of Q	RE (%)	Variance	V_{CE-MP}/V_{RVR}
G_3	10^{-03}	4.01208×10^{-06}	2.16×10^{-03}	3.01610×10^{-18}	$6.12 \times 10^{+00}$
G_3	10^{-06}	3.99992×10^{-12}	2.31×10^{-03}	1.00018×10^{-35}	$1.85 \times 10^{+06}$
G_6	10^{-03}	4.00803×10^{-06}	7.49×10^{-04}	4.02467×10^{-21}	$9.29 \times 10^{+03}$
G_6	10^{-06}	4.00001×10^{-12}	5.00×10^{-05}	3.99998×10^{-36}	$9.40 \times 10^{+06}$

Table 7.3: Performances of the RVR method for the evaluation of G_3 and G_6. Terminals are the four corner nodes and $N = 10^6$.

Network	Q	Estimate	RE (%)	Variance	V_{CMC}/V_{RVR}
C_{10}	$4.58481e-02$	$4.67262e-02$	$1.92e+00$	$6.57617e-06$	$6.65e+02$
C_{20}	$2.33295e-04$	$2.32376e-04$	$3.94e-01$	$2.07036e-11$	$1.13e+06$
C_{30}	$8.86419e-07$	$8.74893e-07$	$1.30e+00$	$4.59076e-16$	$1.93e+08$
C_{40}	$2.99368e-09$	$3.00302e-09$	$3.12e-01$	$1.07423e-21$	$2.79e+11$
C_{50}	$9.47855e-12$	$9.58856e-12$	$1.16e+00$	$1.84357e-26$	$5.14e+13$

Table 7.4: Performances of the RVR method for the evaluation of complete networks. $\mathcal{K} = \mathcal{V}$, the common link unreliability q is equal to 0.55 and the sample size N is equal to 10.

see that relative errors are acceptable for all considered cases even if the sample size is small ($N = 10$) and substantial gains in variance are obtained for all cases. In particular, the improvement of RVR over Crude Monte Carlo increases with the rarity of the considered event.

7.6 Conclusions

As the reader can see, the literature on this topic is considerable, and the number of ideas that have been explored so far to deal with rare event in the network reliability family of metrics is high. One of the reasons of this is probably the fact that the computational cost of the exact computation of these measures of the quality of a system face to the failures of the components is extremely high.

In the chapter, we underlined the qualities of some of the methods that have been presented, and we can say that the algorithmic development in the area has reached pretty good performances. Most of the methods combine, on one hand, the application of some general probabilistic properties, and on the other the exploitation of the particular structure of the network reliability evaluation problem, in order to reach an efficient solution. The ideas based on putting the problem in terms of a dynamic auxiliary model, and the methods that operate recursively on the network while using polynomial reduction techniques appear as the most promising ones. In both cases, even if there are some results about the complexity of the procedures as a function of some graph properties, and about their theoretical efficiencies, a considerable research effort is still needed to better understand their behavior.

Bibliography

[1] F. Altiparmak, B. Dengiz, and A. E. Smith. Optimal design of reliable computer networks: A comparison of metaheuristics. *Journal of Heuristics*, 9(6):471–487, 2003.

[2] F. Altiparmak, M. Gen, B. Dengiz, and A. Smith. A network-based genetic algorithm for design of communication networks. *Journal of Society of Plant Engineers Japan*, 15(4):184–190, 2004.

[3] M. O. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Transactions on Reliability*, R-35(3), 1986.

[4] M. O. Ball, C. J. Colbourn, and J. S. Provan. Network reliability. In *Handbook of Operations Research: Network Models*, pages 673–762. Elsevier North-Holland, 1995.

[5] B. Barán and F. Laufer. Topological optimization of reliable networks using a-teams. In *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, 1999.

[6] E. E. Barlow and F. Proschan. *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart & Winston, New York, 1975.

[7] Y. K. Belyayev, B. V. Gnedenko, and A. D. Solovyev. *Mathematical Methods of Reliability Theory*. Academic Press, New York, 1969.

[8] H. Cancela and M. El Khadiri. A recursive variance-reduction algorithm for estimating communication-network reliability. *IEEE Transactions on Reliability*, 44(4):595–602, December 1995.

[9] H. Cancela and M. El Khadiri. An improvement to the total hazard method for system reliability simulation. *Probability in the Engineering and Informational Sciences*, 10(2):187–196, 1996.

[10] H. Cancela and M. El Khadiri. A simulation algorithm for source-terminal communication network reliability. In *Proceedings of the 29th Annual Simulation Symposium*, pages 155–161, New Orleans, Louisiana, April 1996. IEEE Computer Society Press.

[11] H. Cancela and M. El Khadiri. Series-parallel reductions in Monte Carlo network reliability evaluation. *IEEE Transactions Rel.*, 47(2):159–164, June 1998.

[12] H. Cancela and M. El Khadiri. On the RVR simulation algorithm for network reliability evaluation. *IEEE Transactions Rel.*, 52(2):207–212, June 2003.

[13] H. Cancela and M. El Khadiri. On the accuracy of the RVR estimator of K-terminal unreliability parameter. In *7th International Workshop on Rare Event Simulation (RESIM 2008)*, Rennes, France, September 2008.

[14] H. Cancela, M. El Khadiri, and G. Rubino. An efficient simulation method for K-network reliability problem. In *6th International Workshop on Rare Event Simulation (RESIM 2006)*, Bamberg, Germany, October 2006.

[15] H. Cancela, L. Murray, and G. Rubino. Splitting in source-terminal network reliability estimation. In *7th International Workshop on Rare Event Simulation (RESIM 2008)*, Rennes, France, September 2008.

[16] H. Cancela, G. Rubino, and M. E. Urquhart. Path set based conditioning for transient simulation of highly dependable networks. In *7th IFAC Symposium on Cost Oriented Automation*, Gatineau/Ottawa, Canada, June 2004.

[17] H. Cancela and M. E. Urquhart. RVR simulation techniques for residual connectedness network reliability evaluation. *IEEE Transactions on Computers*, 51(4):439–443, April 2002.

[18] C. J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, New York, 1987.

[19] J. L. Cook and J. E. Ramirez-Marquez. Two-terminal reliability analyses for a mobile ad hoc wireless network. *Reliability engineering and systems safety*, 92(6):821–829, 2007.

[20] D. Deeter and A. Smith. Heuristic optimization of network design considering all-terminal reliability. In *Proceedings Annual Reliability and Maintainability Symposium*, pages 194–199, 1997.

[21] D. Deeter and A Smith. Economic design of reliable networks. *IIE Transactions*, 30(12):1161–1174, 1998.

[22] B. Dengiz, F. Altiparmak, and A.E. Smith. Efficient optimization of all-terminal reliable networks, using an evolutionary approach. *IEEE Transactions on Reliability*, 46(1):18–26, 1997.

[23] B. Dengiz, F. Altiparmak, and A.E. Smith. Local search genetic algorithm for optimal design of reliable networks. *IEEE Transactions on Evolutionary Computation*, 1(3):179–188, 1997.

[24] S. Duarte and B. Barán. Multiobjective network design optimisation using parallel evolutionary algorithms (text in spanish). In *XXVII Conferencia Latinoamericana de InformŠatica*, 2001.

[25] S. Duarte, B. Barán, and D. Benítez. Telecommunication network design with parallel multi-objective evolutionary algorithms. In *Proceedings of the 2003 IFIP/ACM Latin America conference*, pages 1 – 11. ACM, 2003.

[26] M. El Khadiri and G. Rubino. A Monte-Carlo method based on antithetic variates for network reliability computations. Technical Report 626, IRISA, Campus de Beaulieu, 35042 Rennes, France, 1992.

[27] M. El Khadiri and G. Rubino. Reliability evaluation of communication networks. In *SAFECOMP'92, International Conference on Safety, Security and Reliability of Computers*, Zurich, Suisse, October 1992.

[28] T. Elperin, I. Gertsbakh, and M. Lomonosov. Estimation of network reliability using graph evolution models. *IEEE Transactions on Reliability*, 40(5), 1991.

[29] G. S. Fishman. A comparison of four Monte Carlo methods for estimating the probability of s-t conectedness. *IEEE Transactions on Reliability*, R-35(2), June 1986.

[30] G.S. Fishman. A Monte Carlo sampling plan for estimating network reliability. *Operational Research*, 34, 1986.

[31] I. B. Gertsbakh. *Statistical Reliability Theory*. Marcel Dekker, Inc., New York and Bassel, 1989.

[32] M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. Technical Report 188, Institüt fur Mathematik, Universität Augsburg, Allemagne, 1989.

[33] M. Grötschel, C. L. Monma, and M. Stoer. Polyhedral approaches for network survivability. Technical Report 189, Institüt fur Mathematik, Universität Augsburg, Allemagne, 1990.

[34] D. Günnec and F. S. Salman. Assessing the reliability and the expected performance of a network under disaster risk. In *Proceedings of the International Network Optimization Conference, INOC 2007*, Spa, Belgium, 2007.

[35] K.-P. Hui, N. Bean, M. Kraetzl, and D. Kroese. The tree cut and merge algorithm for estimation of network reliability. *Probability in the Engineering and Informational Sciences*, 17:23–45, 2003.

[36] K-P. Hui, N. Bean, M. Kraetzl, and Dirk Kroese. The cross-entropy method for network reliability estimation. *Annals of Operations Research*, 134:101–118(18), February 2005.

[37] Kin-Ping Hui. Monte Carlo Network Reliability Ranking Estimation. IEEE *Transactions on Reliability*, 56(1):50–57, March 2007.

[38] R. Karp and M.G. Luby. A new Monte Carlo method for estimating the failure probability of an n-component system. *Computer Science Division*, University of California(Berkeley), 1983.

[39] Hui Kroese, D. P., K.-P., and S. Nariai. Network Reliability Optimization via the Cross-Entropy Method. IEEE *Transactions on Reliability*, 56(2):275–287, June 2007.

[40] H. Kumamoto, K. Tanaka, and K. Inoue. Efficient evaluation of system reliability by monte carlo method. *IEEE Transactions on Reliability*, R-26(5), Dec. 1977.

[41] H. Kumamoto, K. Tanaka, K. Inoue, and E. J. Henley. Dagger-sampling Monte Carlo for system unavailability evaluation. *IEEE Transactions on Reliability*, R-29(2), June 1980.

[42] A. Kumar, R Pathak, and M Gupta. Genetic algorithm based approach for designing computer network topology. In *Proceedings of the 1993 ACM Conference on Computer Science*, pages 358–365, 1993.

[43] Lin L. and M. Gen. A self-controlled genetic algorithm for reliable communication network design. In *IEEE Congress on Evolutionary Computation*, pages 640–647, 2006.

[44] Yan Li. Measuring individual residence's accessible probability by using geographical information systems. In *Proceedings of the 2nd Symposium on Transportation Network Reliability (INSTR 2004)*, pages vol.3, pp. 239 –244, Christchurch, New Zealand, 2004.

[45] M. O. Locks and A. Satyanarayana, editors. *Network Reliability – The State of the Art*, volume R-35, No. 3. IEEE Transactions on Reliability, 1986.

[46] M. Marseguerra, E. Zio, L. Podofillini, and D. Coit. Optimal design of reliable network systems in presence of uncertainty. *IEEE Transactions on Reliability*, 54(2):243–253, 2005.

[47] C.K. Wong M.C. Easton. Sequential destruction method for Monte Carlo evaluation of system reliability. *IEEE Transactions on Reliability*, R-29(1), April 1980.

[48] L. Murray and H. Cancela. Comparison of five monte carlo methods to estimate the network diameter constrained reliability. In *Proceedings of CLEI 2007 (Latin American Conference on Informatics)*, San José, Costa Rica, October 2007.

[49] Nobuoto Nojima. Prioritization in upgrading seismic performance of road network based on system reliability analysis. In *Third China-Japan-US Trilateral Symposium on Lifeline Earthquake Engineering*, Kunming, China, 1998.

[50] P. Premprayoon and P. Wardkein. Topological communication network design using ant colony optimization. In *The 7th International Conference on Advanced Communication Technology*, pages 1147–1151, 2005.

[51] J. S. Provan. Bounds on the reliability of networks. *IEEE Transactions on Reliability*, R-35(3), 1986.

[52] J.S. Provan and M.O. Ball. Computing network reliability in time polynomial in the number of cuts. *Operational Research*, 32, 1984.

[53] S. Ross. System reliability evaluation by simulation: random hazards versus importance sampling. *Probability in the Engineering and Informational Sciences*, 6:119–126, 1992.

[54] S.M. Ross. Variance reduction in simulation via random hazard. *Probability in the Engineering and Informational Sciences*, 4, 1990.

[55] G. Rubino. Network reliability evaluation. In K. Bagchi and J. Walrand, editors, *State-of-the art in performance modeling and simulation*. Gordon and Breach Books, 1998.

[56] B. Sanso and F. Soumis. Communication and transportation networks reliability using routing models. *IEEE Transactions on Reliability*, R-30(5), 1981.

[57] H. A. Taboada, F. Baheranwala, and D. W. Coit. Practical solutions for multi-objective optimization: An application to system reliability design problems. *Reliability Engineering and System Safety*, 92:314–322, 2007.

[58] H. A. Taboada, J. Espiritu, and D. W. Coit. Moms-ga: A multi-objective multi-state genetic algorithm for system reliability optimization design problems. *IEEE Transactions on Reliability*, 57(1), 2007.

[59] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal of Computing*, 8:410–421, 1979.

[60] R.M. Van Slyke and H. Frank. Network reliability analysis: part i. *Networks*, 1, 1972.

[61] H. Wakabayashi. Network reliability improvement: Probability importance and criticality importance. In *Proceedings of the 2nd Symposium on Transportation Network Reliability (INSTR 2004)*, pages vol.3, pp. 204–210, Christchurch, New Zealand, 2004.