Analysis of Two Methods for Reliability Estimation on the Static Network Model

Héctor Cancela
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
cancela@fing.edu.uy

and

Leslie Murray
FCEIA, Universidad Nacional de Rosario
Rosario. Argentina
lmurray@fceia.unr.edu.ar

and

Gerardo Rubino
INRIA-Rennes, Bretagne Atlantique
Rennes, France
Gerardo.Rubino@inria.fr

Abstract

This work discusses the F-Monte Carlo and the Destruction Spectrum Monte Carlo methods for unreliability estimation of the Static Network Model. The objective is to develop a fine analysis using analytical expressions for both methods, designed to work on the homogeneous case, in which the components share the same distribution. For the first time in literature, the variances of these methods are studied using explicit formulas, which are useful, for instance, to study the precision of the procedures. Also, an analytical study of the computational complexity of the algorithms is performed, based on the observation that the most time-consuming operation in their implementations is the evaluation of the structure function of the system under study. Both techniques are also compared against the Standard Monte Carlo method. A number of computational experiments are performed to validate the analytical formulas developed in the paper, and thus our results, showing good agreement and cross-validating our starting observation and the simulators used.

1 Introduction

In recent years, network models have been an important support for the study of a wide variety of problems. The dynamic or static nature, the size, topology and type of connectivity, as well as the different definitions of reliability, make network models appropriate in different contexts. Below, we show a selection of published articles in which network models serve as a basis for the analysis of a variety of problems.

The design of power transmission networks was addressed in [1]. Also in this context, [2] discusses reliability and vulnerability of electric power networks based on the IEEE RTS96 test system.

In [3], the authors focus on the reliability of critical infrastructure networks and social systems —like, for example, potable water networks— in order to predict important changes and critical events. In [4], the authors analyse water distribution networks, exploring the trade-off between cost and reliability.

Vehicle networks are the subject of study in [5], where the authors analyse survivability and rewards by means of reliability block diagrams. Critical events on railway networks over cases of study taken from the Italian public network are addressed in [6]. In [7], the authors study connectivity and reliability problems over transportation networks, considering, in particular, the analysis of remote destinations and the influence of natural disasters. In [8], we can find the unification into a single model of systems usually modeled separately, like public transport and electricity provision.

Reliability analysis of large and complex infrastructure networks using Bayesian Networks is addressed in [9]. The location of relief stations on a road network in cases of disaster is the subject of study in [10].

The present work is based on the so called Static Network Model, a model designed for the analysis of connectivity problems. In this model, the network must guarantee the existence of link paths to allow the connection between a set of nodes called terminal nodes. Failures only affect links, not nodes. The failure of a link consists of its total interruption and occurs with known probability. Then, there is a probability —called reliability— that the non-failed links form paths that guarantee the connection between the terminal nodes.

The computation of the reliability thus defined belongs to the class of NP-hard problem (see [11] and [12]), so, for medium to large sized networks, the exponential complexity of the algorithms makes its computation impossible [13, 14]. One proposal to avoid this problem is to resign the exact computation and to estimate the reliability using Standard Monte Carlo. However, when the network is highly reliable (i.e., when links' failure probabilities are extremely low), this becomes very inefficient because the estimator's variance grows boundlessly.

Over the years, a large amount of research has aimed to obtain reliability estimators for the Static Network Model with lower variance than that of the Standard Monte Carlo. Over the years, a large amount of research has aimed to obtain reliability estimators for the Static Network Model with lower variance than that of the Standard Monte Carlo.

The pioneering works of applying Monte Carlo simulation to system and network reliability evaluation range from the 1970s to the 1990s. These works, among which we have [15], [16], [17], [18], [19], [20], [21] and [22], introduce techniques such as Dagger-Sampling, Sequential Destruction and specific sampling plans along with graph evolution models. They laid the conceptual and algorithmic foundations for the following research.

At a later stage, works like [23], [24], [25], [26] and [27] aim to improve the efficiency of Monte Carlo sampling, turning the problem into rare event analysis. The contributions of these works include Recursive Variance Reduction algorithms, Importance Sampling and Splitting methods. They made Monte Carlo simulation practical in scenarios where network failures are extremely rare yet critical to evaluate.

Another series of articles like [28], [29], [30] and [31] propose more sophisticated algorithms and probabilistic models. They include the Tree Cut and Merge algorithm, the Cross-Entropy method, the use of Marshall-Olkin copulas for dependent failures and Diameter-Constrained Reliability. They represent techniques beyond pure Monte Carlo, incorporating advanced statistics and probabilistic modeling.

Recent publications such as [32], [33], [34], [35], [36] and [37] have focused their analysis of network reliability on new variance reduction strategies, exploring adaptive surrogate models, sampling-based comparisons, and LP-embedded approaches for stochastic flow networks. They also integrate active learning, Quasi-Monte Carlo, and hybrid resampling Cross-Entropy techniques to improve computational efficiency while maintaining accuracy in both static and dynamic network settings.

We can find a historical survey of many years of network reliability research in [38], and a book that synthesizes combinatorial, analytical, and simulation-based approaches in [39]. Both offer broad overviews of achievements and challenges in the field, serving as reference points for researchers.

While these methods have been shown to be useful to improve the efficiency of reliability estimation (in terms of precision and computational effort), in general their performance has been studied empirically over a set of example graph topologies, and there is a lack of analytical studies that support these conclusions in general.

In this work, we analyse in detail two methods for estimating the reliability measures defined in the Static Network Model. These are: F-Monte Carlo [31] and Destruction Spectrum [39]. Both methods are based on an exact reliability formula and mechanisms for estimating some of the parameters of this formula. We compute the variances and the computational cost of both methods, giving explicit, analytical formulations. Some of these results are verified experimentally.

The remainder of the paper is organized as follows. Section 2 presents the details of the Static Network Model. Section 3 presents the F-Monte Carlo simulation method and develops an analytical formulation for its variance. Section 4 presents the Destruction Spectrum based Monte Carlo method, and Section 5 develops an analytical formulation for its variance. Section 6 is a brief review of Standard Monte Carlo in order to compare its variance with that of previous methods. Section 7 is devoted to the validation of the analytical formulations previously developed, by means of a numerical example over a small graph topology. Section 8 develops analytical computational cost formulations under the hypothesis that the most costly operation is the evaluation of the structure function of the network for each simulated sample. Section 9 validates the computational cost formulations by

comparing them against empirical data from experiments over several example topologies. Finally, Section 10 presents the general conclusions of the work.

2 Static Network Model

The Static Network Model is a graph $G = G(V,E)$, here assumed to be undirected, in which a set of nodes, V, is connected by a set of links, E, with $|V| = n$ and $|E| = m$. Nodes are perfect, only links fail, and they do it independently from each other. Links are modeled by the random vector $\mathbf{X} = (X_1, \ldots, X_m) \in \Omega = \{0,1\}^m$, where $X_i = 0$ means that link i is failed (we say *down*), which is equivalent to being removed from the graph, and $X_i = 1$ means that link i is operational (we say *up*). Vector $\mathbf{X}$ is the *configuration* of the network. We denote by $\mathbf{x} = (x_1, \ldots, x_m)$ any possible value of $\mathbf{X}$.

The network itself also has a state in $\{0,1\}$: state 1 means that the network is *up*, or working, and state 0 means that the network is *down*, or failed. With each configuration, one of these two values is associated. If for configuration $\mathbf{x}$ we remove all the links i where $x_i = 0$, then we obtain a *partial graph* $G' = G'(V,\mathbf{x})$ of G (same nodes, part of the edges). Then we use a connectivity-based criterion on G' to decide which network state corresponds to configuration $\mathbf{x}$. A pretty general one, used in this paper, is the following: given a subset $K \subseteq V$ of nodes, the network is *up* when $\mathbf{X} = \mathbf{x}$ if K is a subset of one of $G'(V,\mathbf{x})$'s connected components. Observe that Ω is thus partitioned into two sets, the set of configurations where the network is *up*, and that of the configurations where it is *down*. These two network conditions are reflected by the so called structure function ϕ from Ω into $\{0,1\}$ as follows: $\phi(\mathbf{x}) = 0$ indicates that the network is *down* when $\mathbf{X} = \mathbf{x}$, whereas $\phi(\mathbf{x}) = 1$ means that the network is *up* in the same case.

It is also useful to work with the complement of $\phi(\mathbf{X})$, that is, with $\psi(\mathbf{X}) = 1 - \phi(\mathbf{X})$, with the obvious meaning that $\psi(\mathbf{X}) = 1$ indicates that the network is *down* and $\psi(\mathbf{X}) = 0$ indicates that the network is *up*. When we deal with reliability, it is usually preferable to use $\phi(\mathbf{X})$, whereas for working with unreliability, it is more convenient to use $\psi(\mathbf{X})$.

A homogeneous network model is one for which $\mathbb{P}\{X_i = 1\} = p$ and $\mathbb{P}\{X_i = 0\} = q = 1 - p$, $i = 1, \ldots, m$. As said in the Abstract, we want to base our paper on analytical expressions to analyze the performance of two Monte Carlo procedures designed to operate on homogeneous network models.

Now we define the unreliability Q of the network and its reliability R as follows:

$$Q = \mathbb{P}\{\psi(\mathbf{X}) = 1\} \quad \text{and} \quad R = \mathbb{P}\{\phi(\mathbf{X}) = 1\} = 1 - Q.$$

From now on, we will focus on the analysis of the network unreliability based on $\psi(\mathbf{X})$.

Let us make a partition on the space $\Omega = \{0,1\}^m$ in terms of the following subsets:

$$\Omega_j = \left\{ \mathbf{x} \in \Omega \ : \ \sum_{i=1}^{m} x_i = m - j \right\} \quad j = 0, \ldots, m,$$

In words, Ω_j is the subset of all configurations having exactly j zeros. As the links are mutually independent, the probability of any of the configurations in Ω_j is $q^j\, p^{m-j}$. Considering that there are $\binom{m}{j}$ configurations in Ω_j, we have:

$$\sum_{j=0}^{m} \binom{m}{j} q^j\, p^{m-j} = 1.$$

If $\alpha_j \leq 1,\ j = 0, \ldots, m$, is the fraction of configurations $\mathbf{x}$ in Ω_j for which $\psi(\mathbf{x}) = 1$, then:

$$\sum_{j=0}^{m} \alpha_j \binom{m}{j} q^j\, p^{m-j} = Q. \tag{1}$$

Observe that there is only one configuration $\mathbf{x} \in \Omega_0$ (none of the links *down*). For this configuration, we have $\psi(\mathbf{x}) = 0$ and, therefore, $\alpha_0 = 0$. Besides, there is only one configuration $\mathbf{x} \in \Omega_m$ (all the links *down*). In this case $\psi(\mathbf{x}) = 1$ and then $\alpha_m = 1$. Taking this into account:

$$Q = \sum_{j=1}^{m} \alpha_j \binom{m}{j} q^j\, p^{m-j}. \tag{2}$$

As $p = 1 - q$, the unreliability Q, seen as a function of q, is a polynomial, $Q = \sum_{j=1}^{m} \alpha_j \binom{m}{j} q^j\, (1-q)^{m-j}$, out of which only the coefficients $\alpha_j,\ j = 1, \ldots, m-1$, are unknown.

3 F-Monte Carlo

F-Monte Carlo is a simulation method that has not been widely used, nor cited, by the scientific community, and that we take from [31]. It is based on replacing the coefficients α_j in (2) by unbiased estimators, $\widehat{\alpha}_j,\ j=1,\ldots,m$, in order to build an unbiased network unreliability estimator, $\widehat{Q}_{\mathrm{F}}$, as follows:

$$\widehat{Q}_{\mathrm{F}} = \sum_{j=1}^{m} \widehat{\alpha}_j \binom{m}{j} q^j\, p^{m-j}. \tag{3}$$

Note that, as $\alpha_m = 1$, this coefficient does not need any estimation, and we simply set $\widehat{\alpha}_m = 1$.

Every addend in (3) is composed of three parts: $q^j\, p^{m-j}$ that indicates the probability mass of the term, $\binom{m}{j}$ that only depends on the index j, and $\widehat{\alpha}_j$ that, for every j, is only a function of the network topology but does not depend on the probability distribution of $\mathbf{X}$. This is probably the most important feature of F-Monte Carlo because, while for most of the variance reduction methods, sampling becomes more inefficient as the unreliability of the components becomes higher, in F-Monte Carlo the values of q and p are not part of the sampling process at all.

3.1 Sampling F-Monte Carlo

The parameters $\widehat{\alpha}_j$ are Crude Monte Carlo estimators in the subspaces $\Omega_j,\ j=1,\ldots,m-1$. The estimator $\widehat{\alpha}_j$ is built using N_j samples, $\mathbf{X}_j^{(i)},\ i=1,\ldots,N_j$, where $\psi(\mathbf{X}_j^{(i)})$ may be equal to 1 or to 0, regardless of the probability distribution of the links. If the network is highly reliable, sampling $\mathbf{X}^{(i)}$ under the original links' probability distributions (i.e., using q and p) will generate a vast majority of random configurations for which $\psi(\mathbf{X}_j^{(i)}) = 0$ and a few or none for which $\psi(\mathbf{X}_j^{(i)}) = 1$, resulting in high variance estimators per subspace.

As the values of $\widehat{\alpha}_j$ do not depend on the probability distribution of $\mathbf{X}$, it is convenient to use different links' probability distributions in order to make more efficient estimates. To cope with this problem we introduce the random variable $\mathbf{Y} = (Y_1,\ldots,Y_m) \in \Omega = \{0,1\}^m$, similar to $\mathbf{X}$ in the sense that $Y_i = 0$ means that link i is *down* and $Y_i = 1$ means that link i is *up*, and we propose that $\mathbb{P}\{Y_i=0\} = \mathbb{P}\{Y_i=1\} = 0.5\ \forall\ i$. Based on $\mathbf{Y}$, we may partition the state space $\Omega = \{0,1\}^m$ similarly to how we did with$\mathbf{X}$:

$$\Omega_j = \left\{ \mathbf{y} \in \Omega : \sum_{i=1}^{m} y_i = m - j \right\}, \quad j = 0,\ldots,m.$$

Here we define $\mathbf{Y}_j \in \Omega_j$, as a random vector of size m, composed of $m-j$ ones and j zeros. Under the probability distribution of $\mathbf{Y}$, the network is still homogeneous and the values $\mathbf{Y}_j$ are uniformly distributed in Ω_j: each value of $\mathbf{Y}_j$ appears with probability $1/\binom{m}{j}$. As α_j was defined as the fraction of configurations $\mathbf{x} \in \Omega_j$ for which $\psi(\mathbf{x}) = 1$, no matter the probability distribution of $\mathbf{X}$, α_j is also the fraction of values of $\mathbf{y}_j$ for which $\psi(\mathbf{y}_j) = 1$.

A possible implementation for sampling $\mathbf{Y}_j$ consists of (i) set the value of j, i.e. select the corresponding subspace, (ii) fill with 1s all the components of a vector $\mathbf{y}_j$, and (iii) choose uniformly j out of the m components of $\mathbf{y}_j$ and set them to 0. This can be accomplished by the algorithm RPERM proposed by Fishman in [40]. Finally, a standard estimator of α_j is:

$$\widehat{\alpha}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \psi\big(\mathbf{Y}_j^{(i)}\big), \quad j = 1,\ldots,m, \tag{4}$$

where $\mathbf{Y}_j^{(i)},\ i=1,\ldots,N_j$, is a set of independent samples of $\mathbf{Y}_j$. As the estimators $\widehat{\alpha}_j$ are obtained separately, they are all mutually independent.

Now we transform expression (3), replacing each $\widehat{\alpha}_j$ with the estimators proposed in (4), and we obtain $\widehat{Q}_{\mathrm{F}}$, the F-Monte Carlo network unreliability estimator:

$$\widehat{Q}_{\mathrm{F}} = \sum_{j=1}^{m} \left(\frac{1}{N_j} \sum_{i=1}^{N_j} \psi\left(\mathbf{Y}_j^{(i)}\right) \right) \binom{m}{j} q^j\, p^{m-j}.$$

Recalling that $\alpha_m = \widehat{\alpha}_m = 1$, $\widehat{Q}_{\mathrm{F}}$ can be written as:

$$\widehat{Q}_{\mathrm{F}} = q^m + \sum_{j=1}^{m-1} \left(\frac{1}{N_j} \sum_{i=1}^{N_j} \psi\left(\mathbf{Y}_j^{(i)}\right) \right) \binom{m}{j} q^j\, p^{m-j}. \tag{5}$$

3.2 Variance of the F-Monte Carlo Estimator

The variance of $\widehat{Q}_{\mathrm{F}}$ can be obtained directly from the expression (5):

$$\begin{aligned}
\mathbb{V}\{\widehat{Q}_{\mathrm{F}}\} &= \mathbb{V}\left\{q^m + \sum_{j=1}^{m-1}\left(\frac{1}{N_j}\sum_{i=1}^{N_j}\psi\left(\mathbf{Y}_j^{(i)}\right)\right)\binom{m}{j}q^j\,p^{m-j}\right\} \\
&= \mathbb{V}\left\{\sum_{j=1}^{m-1}\left(\frac{1}{N_j}\sum_{i=1}^{N_j}\psi\left(\mathbf{Y}_j^{(i)}\right)\right)\binom{m}{j}q^j\,p^{m-j}\right\} \\
&= \sum_{j=1}^{m-1}\binom{m}{j}^2 p^{2j}\,q^{2(m-j)}\,\mathbb{V}\left\{\left(\frac{1}{N_j}\sum_{i=1}^{N_j}\psi\left(\mathbf{Y}_j^{(i)}\right)\right)\right\} \\
&= \sum_{j=1}^{m-1}\binom{m}{j}^2 \frac{p^{2j}\,q^{2(m-j)}}{N_j^2}\,\mathbb{V}\left\{\sum_{i=1}^{N_j}\psi\left(\mathbf{Y}_j^{(i)}\right)\right\} \\
&= \sum_{j=1}^{m-1}\binom{m}{j}^2 \frac{p^{2j}\,q^{2(m-j)}}{N_j^2}\,N_j\,\mathbb{V}\left\{\psi\left(\mathbf{Y}_j^{(1)}\right)\right\} \\
&= \sum_{j=1}^{m-1}\binom{m}{j}^2 \frac{p^{2j}\,q^{2(m-j)}}{N_j}\;\alpha_j(1-\alpha_j).
\end{aligned}$$

If we distribute the N Monte Carlo replications uniformly, and let $N_j = N/(m-1),\ j = 1,\ldots,m-1$, we have:

$$\mathbb{V}\{\widehat{Q}_{\mathrm{F}}\} = \frac{m-1}{N}\sum_{j=1}^{m-1}\binom{m}{j}^2 p^{2j}\,q^{2(m-j)}\;\alpha_j(1-\alpha_j).$$

4 Destruction Spectrum and Network Unreliability

Here, we introduce the fundamentals of Destruction Spectrum [39], a method conceived for estimating the unreliability of homogeneous network models. Referring to Destruction Spectrum, in [41] it is stated that "numerically it is identical to the so-called *signature* introduced in [42]". It is also worth noting that the author of [42] addressed the issue again in [43].

Based on the state of the links modeled by the variables $X_i,\ i = 1,\ldots,m$, we call *failure* any transition from 1 to 0, and *repair* any transition from 0 to 1. Similarly, based on the state of the network given by the function $\psi(\mathbf{X})$, we call failure any transition from 0 to 1 and repair any transition from 1 to 0.

We say that a link i whose state X_i equals 0 is failed or *down*, whereas if $X_i = 1$, it is repaired or *up*. On the other hand, $\psi(\mathbf{X}) = 1$ means that the network is failed or *down*, and $\psi(\mathbf{X}) = 0$ indicates that the network is repaired or *up*.

Now we introduce a dynamic model in which, at the beginning, all the links are *up*, and they fail sequentially, no matter the time of the first failure or the times between failures. There are $m!$ different ways of ordering links' failures, each of which can be referred to as a *permutation* of the set of links $E = \{e_1, e_2, \ldots, e_m\}$. If Π is the set of all the *permutations* of E, the general form of any $\pi \in \Pi$ is:

$$\pi = \{e_{(1)}, e_{(2)}, \ldots, e_{(m)}\},$$

where the terms $e_{(i)},\ i = 1,\ldots,m$, are the order statistics of the set E.

At the beginning, when all the links are *up*, the network is also *up*, and it fails simultaneously with the failure of some $e_{(j)},\ 1 \le j \le m$. Thus, for any given *permutation* π, the network is *up* while the first $j-1$ links fail, and it goes *down* exactly when the j–th link fails. We call *anchor* of π, denoted $a(\pi)$, the number j.

As π is a random variable in the state space Π, $a(\pi)$ is a random variable in $\{1,2,\ldots,m\}$. The PDF of $a(\pi)$, called Destruction Spectrum, is:

$$\mathrm{D}_S = (f_1, f_2, \ldots, f_m),$$

where $f_j = \mathbb{P}\{a(\pi) = j\},\ j = 1,\ldots,m$. For small networks, every f_j can be obtained quite simply from a counting process. To do this, we have to generate all the *permutations*, $(\pi_1,\ldots,\pi_{m!})$, and to find the *anchor* of each one of them. Calling A_j the number of *permutations* π for which $a(\pi) = j$, we have $f_j = A_j/m!$. It is clear that

$\sum_{j=1}^{m} f_j = 1$, which shows, as said before, that D_S is a PDF. We call Cumulative Destruction Spectrum, CD_S, the corresponding CDF, that is:

$$\mathrm{CD}_S = (F_1, F_2, \ldots, F_m)$$

where $F_j = \sum_{i=1}^{j} f_i,\ j = 1, \ldots, m$.

In words, $f_j \in \mathrm{D}_S$ is the probability that, for a given *permutation*, the network fails exactly at the time that the j–th link fails, but not earlier, whereas $F_j \in \mathrm{CD}_S$ is the probability that, for a given *permutation*, the network fails at the j–th failure or earlier.

The Destruction Spectrum is closely related to the unreliability of the network model that we have been working with so far. This issue is going to be introduced next. Recall that $\Omega = \bigcup_{j=0}^{m} \Omega_j$, where Ω contains all the possible network configurations, and Ω_j contains all the configurations composed of j zeros and $m-j$ ones. Besides, there are $\binom{m}{j}$ configurations in Ω_j, out of which $\alpha_j \binom{m}{j}$ of them make $\psi(\mathbf{X}) = 1$.

Consider a configuration $\mathbf{x}' \in \Omega_j$ with $\psi(\mathbf{x}') = 1$. Without loss of generality (since all repair sequences are considered), assume that the j links equal to zero in $\mathbf{x}'$ are the first to fail. These j links can be arranged in $j!$ permutations, occupying the first j positions in a sequence of length m. The remaining $m-j$ links (with state one in $\mathbf{x}'$) can be arranged in $(m-j)!$ permutations. Therefore, each such configuration in Ω_j with $\psi(\mathbf{X}) = 1$ gives rise to $j!(m-j)!$ permutations with anchor j or less.

On the other hand, the set E produces $m!$ permutations, of which $f_i\, m!$ have anchor i, for $i = 1, \ldots, m$. Thus,

$$\begin{aligned}
\sum_{i=1}^{j} f_i\, m! &= \alpha_j \binom{m}{j}\, j!\,(m-j)! \\
\sum_{i=1}^{j} f_i &= \alpha_j \binom{m}{j}\, \frac{j!\,(m-j)!}{m!} \\
\sum_{i=1}^{j} f_i &= \alpha_j \\
F_j &= \alpha_j
\end{aligned}$$

Finally, expression (2) can be rewritten in terms of the Destruction Spectrum as:

$$Q_{\mathrm{D}_S} = \sum_{j=1}^{m} F_j \binom{m}{j} q^j\, p^{m-j}. \tag{6}$$

5 Monte Carlo and Destruction Spectrum

As said above, the exact computation of $F_j,\ j = 1, \ldots, m$, requires the generation of the $m!$ possible permutations in Π. Suppose that, instead, we generate randomly $N < m!$ *permutations*, $(\pi_1, \ldots, \pi_N)$, and for each one we find its *anchor*, $a(\pi_j)$. If A_j is the number of *permutations* π for which $a(\pi) = j$, then $\widehat{f}_j = A_j/N$ is a standard estimator of $f_j,\ j = 1, \ldots, m$. Let us denote

$$\widehat{\mathrm{D}}_S = (\widehat{f}_1, \widehat{f}_2, \ldots, \widehat{f}_m).$$

Clearly, $\sum_{j=1}^{m} \widehat{f}_j = 1$. We can also estimate the Cumulative Destruction Spectrum:

$$\widehat{\mathrm{CD}}_S = (\widehat{F}_1, \widehat{F}_2, \ldots, \widehat{F}_m),$$

where $\widehat{F}_j = \sum_{i=1}^{j} \widehat{f}_i,\ j = 1, \ldots, m$. Based on these estimators it is possible to build the following unbiased network unreliability estimator based on the Destruction Spectrum:

$$\widehat{Q}_{\mathrm{D}_S} = \sum_{j=1}^{m} \widehat{F}_j \binom{m}{j} q^j\, p^{m-j}. \tag{7}$$

5.1 Sampling the Destruction Spectrum

Let us transform the expression of $\widehat{Q}_{\mathrm{D}_S}$ in (7), in order to make it more suitable for the computation of its variance.

$$\widehat{Q}_{\mathrm{D}_S} = \widehat{F}_1 \binom{m}{1} q^1\, p^{m-1} + \widehat{F}_2 \binom{m}{2} q^2\, p^{m-2} + \cdots + \widehat{F}_{m-1} \binom{m}{m-1} q^{m-1}\, p + \widehat{F}_m \binom{m}{m} q^m$$

$$
\begin{aligned}
&= \widehat{f}_1 \binom{m}{1} q^1 p^{m-1} + \left(\widehat{f}_1 + \widehat{f}_2\right) \binom{m}{2} q^2 p^{m-2} + \cdots + \left(\widehat{f}_1 + \widehat{f}_2 + \cdots + \widehat{f}_{m-1}\right) \binom{m}{m-1} \\
&\quad q^{m-1} p + \left(\widehat{f}_1 + \widehat{f}_2 + \cdots + \widehat{f}_{m-1} + \widehat{f}_m\right) \binom{m}{m} q^m \\
&= \widehat{f}_1 \left[\binom{m}{1} q^1 p^{m-1} + \binom{m}{2} q^2 p^{m-2} + \cdots + \binom{m}{m-1} q^{m-1} p + \binom{m}{m} q^m \right] + \\
&\quad \widehat{f}_2 \left[\binom{m}{2} q^2 p^{m-2} + \cdots + \binom{m}{m-1} q^{m-1} p + \binom{m}{m} q^m \right] + \cdots + \\
&\quad \widehat{f}_{m-1} \left[\binom{m}{m-1} q^{m-1} p + \binom{m}{m} q^m \right] + \widehat{f}_m \left[\binom{m}{m} q^m \right] \\
&= \widehat{f}_1 U_1 + \widehat{f}_2 U_2 + \cdots + \widehat{f}_{m-1} U_{m-1} + \widehat{f}_m U_m = \sum_{j=1}^{m} \widehat{f}_j U_j,
\end{aligned}
$$

where

$$
U_j = \sum_{i=j}^{m} \binom{m}{i} q^i p^{m-i}
$$

is the probability mass of the set of configurations containing j or more zeros. In summary, we have the following two different expressions to compute the network unreliability estimator based on the Destruction Spectrum:

$$
\widehat{Q}_{\mathrm{D}_S} = \sum_{j=1}^{m} \widehat{F}_j \binom{m}{j} q^j p^{m-j} = \sum_{j=1}^{m} \widehat{f}_j U_j. \tag{8}
$$

Let us now define the indicator random vector $\mathbf{B} = (B_1, \ldots, B_m)$ as follows: for $j = 1, \ldots, m$,

$$
B_j = \begin{cases} 1 & \text{if } a(\pi) = j, \\ 0 & \text{otherwise.} \end{cases} \tag{9}
$$

See that $f_j = \mathbb{P}\{B_j = 1\}$. Then,

$$
\widehat{f}_j = \frac{1}{N} \sum_{i=1}^{N} B_j^{(i)}.
$$

Now, the estimator on the right side of (8) can be written

$$
\widehat{Q}_{\mathrm{D}_S} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{m} B_j^{(i)} U_j. \tag{10}
$$

The components $B_j^{(i)}$ are obtained from N independent samples $\mathbf{B}^{(i)}$, $i = 1, \ldots, N$. All in all, this process consists of sampling N independent and uniformly distributed *permutations* $\pi^{(i)}$, computing $a(\pi^{(i)})$ for each one, setting to 1 the corresponding components of $\mathbf{B}^{(i)}$ and to 0 the rest.

5.2 Variance of the Destruction Spectrum Estimator

The variance of $\widehat{Q}_{\mathrm{D}_S}$ can be obtained directly from (10):

$$
\begin{aligned}
\mathbb{V}\{\widehat{Q}_{\mathrm{D}_S}\} &= \mathbb{V}\left\{ \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{m} B_j^{(i)} U_j \right\} \\
&= \frac{1}{N^2} \mathbb{V}\left\{ \sum_{i=1}^{N} \sum_{j=1}^{m} B_j^{(i)} U_j \right\} \\
&= \frac{1}{N^2} \sum_{i=1}^{N} \mathbb{V} \underbrace{\left\{ \sum_{j=1}^{m} B_j^{(i)} U_j \right\}}_{Z^{(i)}}.
\end{aligned}
$$

Here, $Z^{(i)},\ i = 1,\ldots,N$, is a set of independent copies of a discrete random variable Z, which produces the values U_j with probabilities f_j, $j = 1,\ldots,m$.

$$\begin{aligned}\mathbb{V}\{Z\} &= \mathbb{E}\{Z^2\} - \mathbb{E}\{Z\}^2 \\ &= \sum_{j=1}^{m} f_j\, U_j^2 \;-\; Q^2 \\ &= \sum_{j=1}^{m} f_j\, U_j^2 \;-\; \left(\sum_{j=1}^{m} f_j\, U_j\right)^2.\end{aligned}$$

Finally,

$$\begin{aligned}\mathbb{V}\{\widehat{Q}_{\mathrm{D}_S}\} &= \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{V}\left\{Z^{(i)}\right\} \\ &= \frac{1}{N}\,\mathbb{V}\{Z\} \\ &= \frac{1}{N}\left[\sum_{j=1}^{m} f_j\, U_j^2 \;-\; \left(\sum_{j=1}^{m} f_j\, U_j\right)^2\right],\end{aligned}$$

where $f_1 = \alpha_1$, $f_j = \alpha_j - \alpha_{j-1},\ j = 2,\ldots,m$, and, as stated before, $U_j = \sum_{i=j}^{m} \binom{m}{i} q^i p^{m-i}$.

6 Standard Monte Carlo

The simplest and most straightforward simulation method for network unreliability estimation is Standard —also known as Crude— Monte Carlo. We will briefly review it now, because it will be included in the experimental analysis, along with F-Monte Carlo and Destruction Spectrum.

For the Static Network Model, the unreliability standard Monte Carlo estimator, $\widehat{Q}_{\mathrm{S}}$, is based on N independent copies of the random configuration $\mathbf{X}$, sampled from the original distribution: $\mathbb{P}\{X_i = 0\} = q$ and $\mathbb{P}\{X_i = 1\} = p = 1-q$, $i = 1,\ldots,m$. According to this, the estimator is:

$$\widehat{Q}_{\mathrm{S}} = \frac{1}{N}\sum_{i=1}^{N}\psi\left(\mathbf{X}^{(i)}\right)$$

and its variance is

$$\begin{aligned}\mathbb{V}\{\widehat{Q}_{\mathrm{S}}\} &= \frac{1}{N^2}\,\mathbb{V}\left\{\sum_{i=1}^{N}\psi\left(\mathbf{X}^{(i)}\right)\right\} \\ &= \frac{1}{N}\,\mathbb{V}\left\{\psi\left(\mathbf{X}^{(i)}\right)\right\} \\ &= \frac{1}{N}\,Q(1-Q) \\ &= \frac{1}{N}\left[\sum_{j=1}^{m}\alpha_j\binom{m}{j}q^j\,p^{m-j} \;-\; \left(\sum_{j=1}^{m}\alpha_j\binom{m}{j}q^j\,p^{m-j}\right)^2\right].\end{aligned}$$

7 Variance Formulas Validation

Now we carry out an experimental validation of the variance formulas obtained in previous sections. This validation is based on the Small Grid network shown in Figure 1 and it consists of comparing the exact values, obtained from the formulas, with the estimations of these values obtained from simulations. In compact notation, the variance formulas to validate (which were developed in the previous sections) are the following ones:

$$\mathbb{V}\{\widehat{Q}_{\mathrm{S}}\} = \frac{1}{N}\left[\sum_{j=1}^{m}\alpha_j\,M_j \;-\; \left(\sum_{j=1}^{m}\alpha_j\,M_j\right)^2\right],$$

$$\mathbb{V}\{\widehat{Q}_{\mathrm{F}}\} = \frac{m}{N}\left[\sum_{j=1}^{m}\alpha_j M_j^2 - \sum_{j=1}^{m}\alpha_j^2 M_j^2\right],$$
$$\mathbb{V}\{\widehat{Q}_{\mathrm{D}_S}\} = \frac{1}{N}\left[\sum_{j=1}^{m} f_j U_j - \left(\sum_{j=1}^{m} f_j U_j\right)^2\right],$$

where

$$\begin{cases} M_j = \binom{m}{j} q^j p^{m-j},\ j = 1,\ldots,m \\ f_1 = \alpha_1 \\ f_j = \alpha_j - \alpha_{j-1},\ j = 2,\ldots,m \\ U_j = \sum_{i=j}^{m} M_i = \sum_{i=j}^{m} \binom{m}{i} q^i p^{m-i}. \end{cases}$$

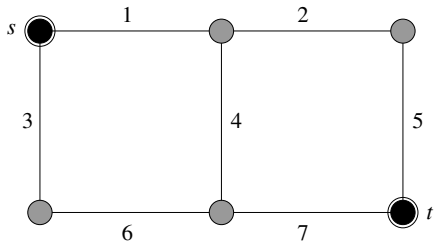

Figure 1: Small Grid

The parameters of the Small Grid network —which can be computed very simply— are the following:

$$\begin{cases} \alpha_1 = 0 \\ \alpha_2 = 4/21 \\ \alpha_3 = 23/35 \\ \alpha_4 = 32/35 \\ \alpha_5 = 1 \\ \alpha_6 = 1 \\ \alpha_7 = 1 \end{cases}$$

Based on these parameters, the unreliability of this network is:

$$\begin{aligned} Q &= \sum_{j=1}^{m} \alpha_j \binom{m}{j} q^j p^{m-j} \\ &= \alpha_1 \binom{7}{1} q\, p^{7-1} + \alpha_2 \binom{7}{2} q^2 p^{7-2} + \alpha_3 \binom{7}{3} q^3 p^{7-3} + \alpha_4 \binom{7}{4} q^4 p^{7-4} + \alpha_5 \binom{7}{5} q^5 p^{7-5} \\ &\quad + \alpha_6 \binom{7}{6} q^6 p^{7-6} + \alpha_7 \binom{7}{7} q^7 p^{7-7} \\ &= \frac{4}{21}\binom{7}{2} q^2 p^5 + \frac{23}{35}\binom{7}{3} q^3 p^4 + \frac{32}{35}\binom{7}{4} q^4 p^3 + \binom{7}{5} q^5 p^2 + 7, q^6 p + \binom{7}{7} q^7 \\ &= 4\,q^2 p^5 + 23\,q^3 p^4 + 32\,q^4 p^3 + 21\,q^5 p^2 + 7\,q^6 p + q^7. \end{aligned}$$

Tables 1, 2 and 3 report the results for the standard Monte Carlo, F-Monte Carlo and Destruction Spectrum Monte Carlo methods respectively. The columns of the tables employ the following notation:

$$\begin{cases} Q : \text{exact value of the network unreliability,} \\ \widehat{Q}_{\mathrm{M}} : \text{estimation of } Q \text{ obtained by means of method M,} \\ \mathbb{V}\{\widehat{Q}_{\mathrm{M}}\} : \text{exact variance of } \widehat{Q}_{\mathrm{M}}, \\ \widehat{\mathbb{V}}\{\widehat{Q}_{\mathrm{M}}\} : \text{variance of } \widehat{Q}_{\mathrm{M}} \text{ estimated by simulation.} \end{cases}$$

Table 1: Standard Monte Carlo $N = 10^8$

	Q	$\widehat{Q}_{\mathrm{S}}$	$\mathbb{V}\{\widehat{Q}_{\mathrm{S}}\}$	$\widehat{\mathbb{V}}\{\widehat{Q}_{\mathrm{S}}\}$
$p = 0.9$	4.12E−02	4.12E−02	3.95E−10	3.95E−10
$p = 0.99$	4.03E−04	4.02E−04	4.03E−12	4.02E−12
$p = 0.999$	4.00E−06	3.83E−06	4.00E−14	3.83E−14

From the first table, we can see how the Standard Monte Carlo method does a reasonable job when approximating the network unreliability values for different levels of link reliability p. The only case where the estimated network unreliability does not quite agree with the exact value is for the most difficult case, where $p = 0.999$ (which is also the case where the estimated variance does not agree with the exact one).

The second and the third tables correspond to the F-Monte Carlo and the Destruction Spectrum procedures. Those techniques can achieve good estimates of the unreliability value for all values of p. The tables also show an extremely good fit between the estimated variances (coming from the experimental results) and the exact variances computed with the analytical formulas developed in this work. This is interesting in two ways: it works as validation of the development of the formulas and, at the same time, it shows that the methods have robust estimates of the precision of their results.

Finally, the comparison between the three tables shows that Standard Monte Carlo has variances much larger than the other two methods (thus, lower precision). For this particular topology, F-Monte Carlo has variances one order of magnitude larger than Destruction Spectrum, which is the most precise algorithm for a given sample size.

Table 2: F-Monte Carlo $N = 10^8$

	Q	$\widehat{Q}_{\mathrm{F}}$	$\mathbb{V}\{\widehat{Q}_{\mathrm{F}}\}$	$\widehat{\mathbb{V}}\{\widehat{Q}_{\mathrm{F}}\}$
$p = 0.9$	4.12E−02	4.12E−02	1.74E−10	1.75E−10
$p = 0.99$	4.03E−04	4.03E−04	4.31E−14	4.36E−14
$p = 0.999$	4.00E−06	4.00E−06	4.71E−18	4.77E−18

Table 3: Destruction Spectrum $N = 10^8$

	Q	$\widehat{Q}_{\mathrm{D}_S}$	$\mathbb{V}\{\widehat{Q}_{\mathrm{D}_S}\}$	$\widehat{\mathbb{V}}\{\widehat{Q}_{\mathrm{D}_S}\}$
$p = 0.9$	4.12E−02	4.12E−02	2.88E−11	2.89E−11
$p = 0.99$	4.03E−04	4.03E−04	6.24E−15	6.27E−15
$p = 0.999$	4.00E−06	4.00E−06	6.74E−19	6.77E−19

8 Computational Cost

In all the methods introduced so far, the most time-consuming operation is the evaluation of the function $\psi(\mathbf{X})$. Given a configuration $\mathbf{x} \in \{0,1\}^m$, i.e. given a state *up* or *down* for every link in the network, $\psi(\mathbf{x}) = 0$ indicates that the subset $K \subseteq V$ of nodes is connected by means of *up* links, whereas $\psi(\mathbf{x}) = 1$ indicates that the subset K of nodes is not connected, despite the *up* links.

To find the value of $\psi(\mathbf{x})$, the usual practice is to launch a graph traversal algorithm like Depth First Search (DFS) from one of the nodes in the subset K and

- set $\psi(\mathbf{x}) = 0$ if at the end of a single traversal all the nodes in K are visited,
- set $\psi(\mathbf{x}) = 1$ otherwise.

The overall computational cost of either Standard Monte Carlo, F-Monte Carlo or Destruction Spectrum is directly related to the number of times that $\psi(\mathbf{X})$ is evaluated. Moreover, the time consumed by the DFS algorithm

(used in this work) strongly depends on the size of the network to which it is applied, measured by the number of *up* links.

Let $\Delta_\psi(\mathbf{X})$ be a random variable that returns the computational time of a single replication of the DFS algorithm used to evaluate the function $\psi(\mathbf{X})$. For networks composed of m links, the complexity of this algorithm is known to be $O(m)$. Note, however, that in each of the successive replications of the Monte Carlo methods, the size of the resulting network is not always m, but $m' \leq m$, where m' is the number of *up* links at the time of the Monte Carlo replication. The value of $\Delta_\psi(\mathbf{X})$ is, therefore, a linear function of the number of *up* links at the time of the execution of the DFS algorithm.

In the following sections, we translate this analysis to the methods presented in this article.

8.1 Standard Monte Carlo Computational Cost

In Standard Monte Carlo, $\psi(\mathbf{X})$ is evaluated each time the configuration $\mathbf{X}^{(i)}$, $i = 1,\ldots,N$, is built. Then:

$$\widehat{Q}_{\mathrm{S}} = \frac{1}{N}\sum_{i=1}^{N}\psi\left(\mathbf{X}^{(i)}\right) \quad \rightarrow \quad \Delta_{\mathrm{S}} = \sum_{i=1}^{N}\Delta_\psi(\mathbf{X}^{(i)}),$$

where $\Delta_\psi(\mathbf{X})$ is the time consumed by a single evaluation of $\psi(\mathbf{X})$ and Δ_S is the overall cost of the Standard Monte Carlo unreliability estimation, measured as the execution time of the whole simulation. If we are interested in the average cost of this estimation, $\overline{\Delta}_{\mathrm{S}}$, we have to find the average of the random variables $\Delta_\psi(\mathbf{X}^{(i)})$, $i = 1,\ldots,N$. To do this, we need to consider the number of *up* links in every sampled state.

There are $\binom{m}{i}$ configurations in which i links are *up* and $m-i$ links are *down*. Besides, each configuration containing exactly i links *up* occurs with probability $p^i q^{m-i}$. Calling m_{up} to the average number of *up* links in a sampled configuration, we have:

$$m_{up} = \sum_{i=0}^{m} i\binom{m}{i}p^i q^{m-i} = m \times p.$$

The average value of $\Delta_\psi(\mathbf{X}^{(i)})$ is, therefore, $\delta \times m_{up}$, where δ is a proportionality constant. Finally:

$$\begin{aligned}\overline{\Delta}_{\mathrm{S}} &= \sum_{i=1}^{N}\delta \times m_{up} \\ &= N \times \delta \times m \times p.\end{aligned} \tag{11}$$

8.2 F-Monte Carlo Computational Cost

$$\widehat{Q}_{\mathrm{F}} = q^m + \sum_{j=1}^{m-1}\left(\frac{1}{N_j}\sum_{i=1}^{N_j}\psi\left(\mathbf{Y}_j^{(i)}\right)\right)\binom{m}{j}q^j p^{m-j} \quad \rightarrow \quad \Delta_{\mathrm{F}} = \sum_{j=1}^{m-1}\sum_{i=1}^{N_j}\Delta_\psi(\mathbf{Y}_j^{(i)}).$$

Here, $\Delta_\psi(\mathbf{Y}_j)$ is the time consumed by the evaluation of $\psi(\mathbf{Y}_j)$ and Δ_F is the total cost of the F-Monte Carlo unreliability estimation. See that $\mathbf{Y}_j^{(i)}$, $j = 1,\ldots,m$, is a configuration sampled randomly with exactly j links in the *down* state and $m-j$ links in the *up* state. Since the number of *up* links in any of the DFS applications is not random, we must compute Δ_{F} as a deterministic value:

$$\Delta_{\mathrm{F}} = \sum_{j=1}^{m-1} N_j \times \delta \times (m-j)$$

Assuming that the total number of Monte Carlo replications is N, if we let $N_j = N/(m-1)$, $j = 1,\ldots,m-1$, we have:

$$\begin{aligned}\Delta_{\mathrm{F}} &= \sum_{j=1}^{m-1}\frac{N}{m-1} \times \delta \times (m-j) \\ &= \frac{N}{m-1} \times \delta \times \sum_{j=1}^{m-1}(m-j)\end{aligned}$$

$$= \frac{N}{m-1} \times \delta \times \frac{(m-1)\,m}{2}$$
$$= \frac{N \times \delta \times m}{2}.$$

8.3 Destruction Spectrum Computational Cost

$$\widehat{Q}_{\mathrm{D}_S} = \sum_{j=1}^{m} \widehat{F}_j \binom{m}{j} q^j\, p^{m-j} \quad \rightarrow \quad \Delta_{\mathrm{D}_S} = \sum_{i=1}^{N} H(G)^{(i)}. \tag{12}$$

At the beginning, all the links are *up* and the network is also *up*. The links start to fail randomly until the nodes in K become disconnected. Function $H(G)$, where $G = (V,E)$ is the network under analysis, returns the time elapsed since the start of the replication, with all the links *up*, and the failure that disconnects the nodes in K. Note that after every single failure, it is necessary to check whether the nodes in K are still connected, which means that in every replication there will be as many evaluations of $\psi(\mathbf{X})$ as link failures. It remains to consider the size of the networks on which $\psi(\mathbf{X})$ is evaluated every time.

The expected number of replications in which the network fails after the first link failure is $N \times f_1$. This indicates that $\psi(\mathbf{X})$ should be evaluated $N \times f_1$ times, on a network composed of $m-1$ links *up*.

The expected number of replications in which the network fails after the second link failure is $N \times f_2$. In every one of these replications $\psi(\mathbf{X})$ is evaluated two times, the first one on a network composed of $m-1$ links *up*, the second one on a network composed of $m-2$ links *up*.

The number of links failing in a single replication may grow up to m, because the nodes in K may disconnect only when the m–th failure occurs. However, when all the links fail it is certain that the nodes in K will be disconnected, and it would not be necessary to evaluate $\psi(\mathbf{X})$. According to this, it is only necessary to evaluate configurations in which the number of links failed is between 1 and $m-1$. Then, the computational cost for the estimation $\widehat{Q}_{\mathrm{D}_S}$ is:

$$\begin{aligned}
\Delta_{\mathrm{D}_S} &= N \times \delta\,(f_1(m-1) + f_2((m-1)+(m-2)) + \cdots + f_{m-1}((m-1)+(m-2)+\cdots+(1))) \\
&= N \times \delta\,((f_1+f_2+\cdots+f_{m-1})(m-1) + (f_2+f_3+\cdots+f_{m-1})(m-2) + \\
&\qquad\qquad (f_3+f_4+\cdots+f_{m-1})(m-3) + \cdots + f_{m-1}(1) \\
&= N \times \delta \sum_{i=1}^{m-1} (F_{m-1} - F_{i-1})\,(m-i).
\end{aligned}$$

As the terms F_i, $i = 0,\ldots,m-1$, are unknown, we can use their estimators $\widehat{F}_i$, $i = 0,\ldots,m-1$ instead, building the estimator $\widehat{\Delta}_{\mathrm{D}_S}$ as follows:

$$\widehat{\Delta}_{\mathrm{D}_S} = N \times \delta \sum_{i=1}^{m-1} \left(\widehat{F}_{m-1} - \widehat{F}_{i-1}\right)(m-i).$$

9 Computational Cost Formulas Validation

Here we provide an experimental validation of the formulas derived in previous sections for the computational cost of Standard Monte Carlo, F-Monte Carlo and Destruction Spectrum. In the first part of the experiments, we present two graphs for each method, varying the value of p, one of them showing the computational cost taken from the proposed formula and the other one showing the execution time of a simulation. In the second part of the experiments, we proceed in the same way as in the first part, but varying the number of links, m, of the network under analysis.

In compact notation, the formulas to validate are the following ones:

$$\begin{aligned}
\overline{\Delta}_{\mathrm{S}} &= N \times \delta \times m \times p \\
\Delta_{\mathrm{F}} &= \frac{N \times \delta \times m}{2} \\
\widehat{\Delta}_{\mathrm{D}_S} &= N \times \delta \sum_{i=1}^{m-1} \left(\widehat{F}_{m-1} - \widehat{F}_{i-1}\right)(m-i)
\end{aligned}$$

The first part of the experiments is based on the Dodecahedron Network shown in Figure 2. The Dodecahedron graph is a benchmark in network reliability analysis due to its properties, which include a balanced structure, a significant number of minimal cuts, and a rich cyclic structure. It is small enough for exact calculations in theoretical studies, yet complex enough to test algorithms, making it an ideal benchmark for testing enumeration methods and Monte Carlo simulations. Some examples of using the Dodecahedron as a test benchmark can be found in [44], [45] and [46].

In all the simulations, the programs are executed for a sufficiently long time so that initialization routines have no significant influence on the total execution time. All the results are based on experiments in which $N = 10^8$. In order to compute the formulas, we consider that $N \times \delta = 1$.

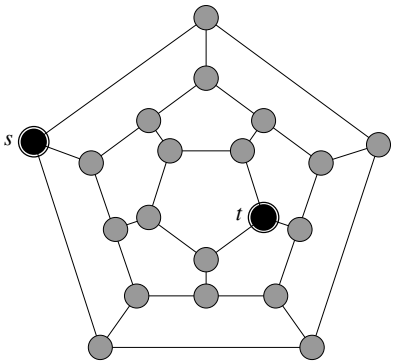

Figure 2: Dodecahedron Network

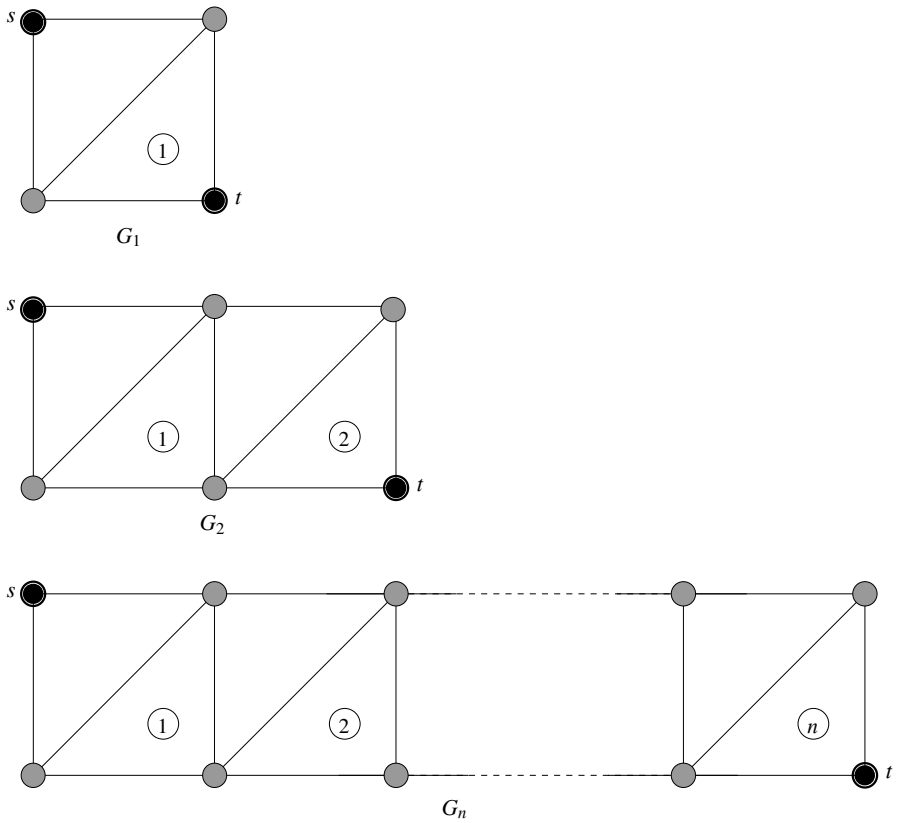

Figure 3: G-Networks

Figures 4, 5 and 6 show the evolution of the computation time over the Dodecahedron Network for different values of the link unreliability q, for the different methods under study. On the left, we see the computational cost from the analytical formulas, and on the right, the actual computing time for running the experiments. As

expected from the formulas, Standard Monte Carlo computing time grows approximately linearly with the value of q; while F-Monte Carlo and Destruction Spectrum have computing times that do not depend on the value of q.

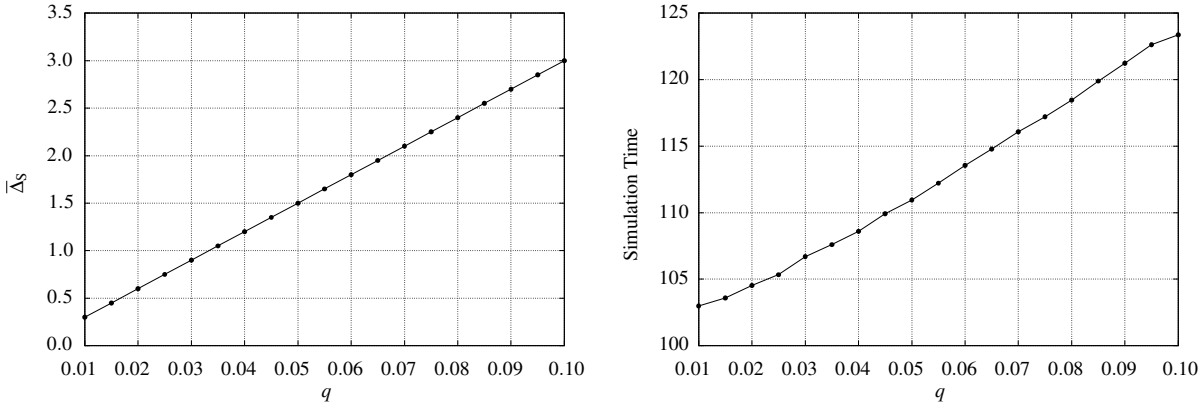

Figure 4: Standard Monte Carlo over Dodecahedron Network

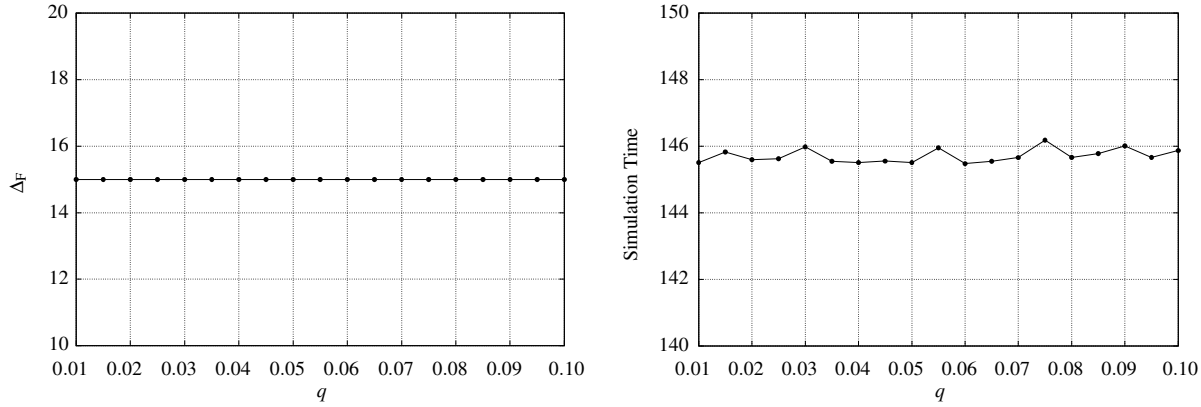

Figure 5: F-Monte Carlo over Dodecahedron Network

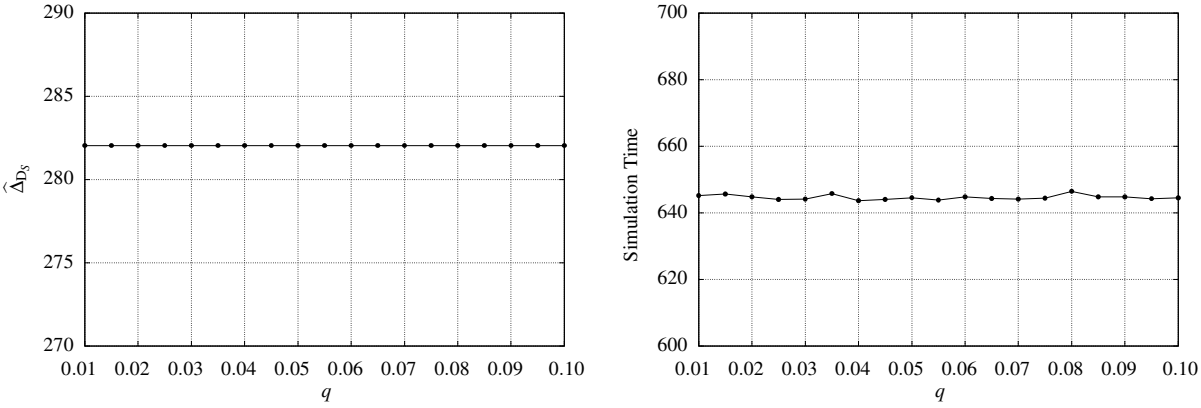

Figure 6: Destruction Spectrum over Dodecahedron Network

The second part of the experiments is based on the G-Networks shown in Figure 3. Again, we fixed $N = 10^8$ for all the simulations, and we computed the formulas considering that $N \times \delta = 1$.

We performed a set of experiments, varying the parameter n for the networks from $n = 1$ (which corresponds to a network with 5 links) up to $n = 20$ (which corresponds to a network with 81 links).

Figures 7, 8 and 9 correspond to the results for the different methods under study, as a function of the value of links, m, according to these graph topologies.

We see again a close agreement between the shape of the predicted computational cost, as given by the analytical formulas developed in this work, and the actual computing time measured when performing the experiments.

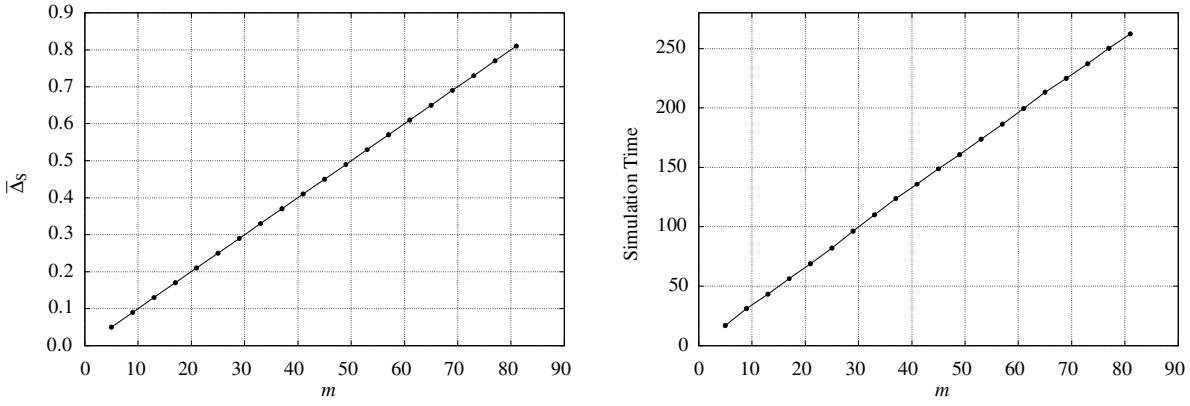

Figure 7: Standard Monte Carlo over *G*-Networks

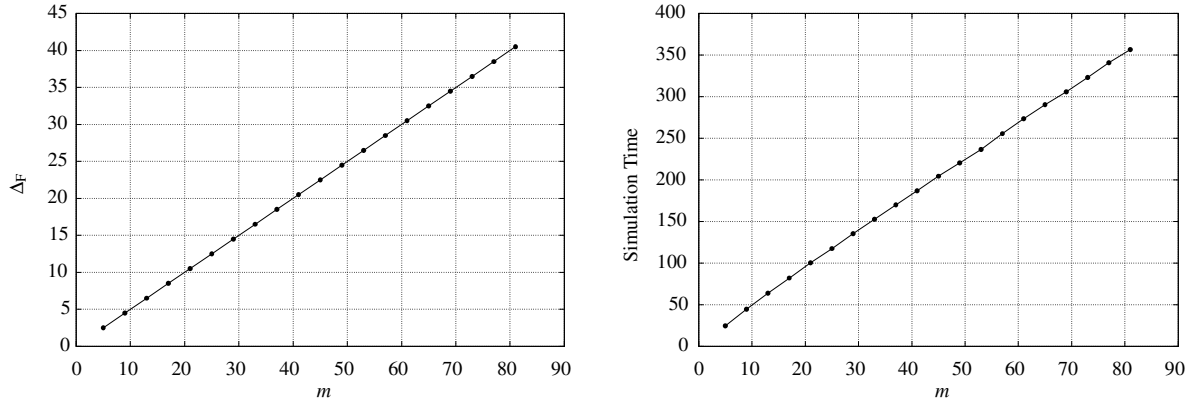

Figure 8: F-Monte Carlo over *G*-Networks

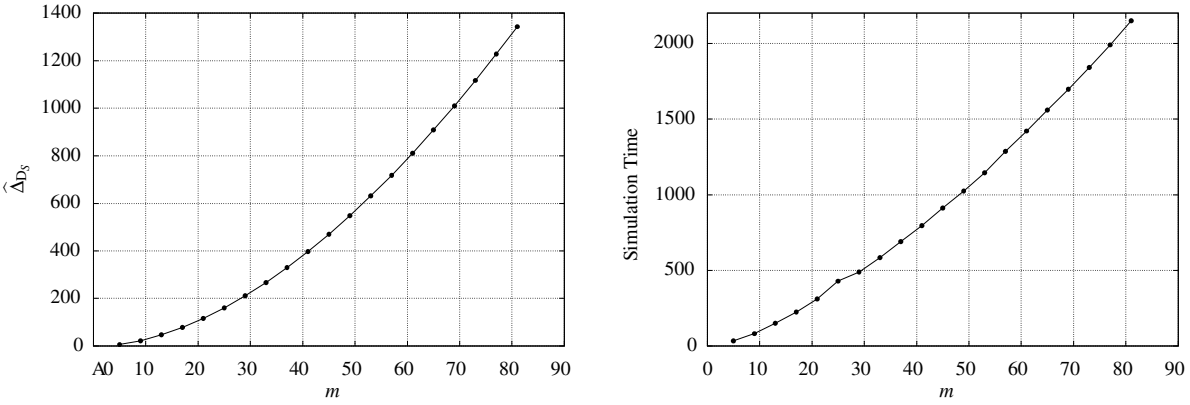

Figure 9: Destruction Spectrum over *G*-Networks

Regardless of the differences in scales and units of measurement, comparing each formula graph with the corresponding simulation time graph, we can see the same trends. This confirms the hypothesis according to which, for all the methods, the highest time-consuming operation is the evaluation of the function $\psi(\mathbf{X})$, and validates the analytical formulas as accurate estimates for the computational costs of the different procedures.

The comparison of the computational costs between methods is not as straightforward as just comparing the analytical formulas, as their implementation implies that the per-operation cost can be different. Nevertheless, as the constants that multiply the number of operations do not change with the network size, it is possible to estimate the growth/evolution of the computational cost for different topologies, and to have a prediction of which method will be more computationally demanding for different graphs even before performing the experimental evaluation.

10 Conclusions

In this work, we have studied in some detail two efficient Monte Carlo methods for evaluating the reliability of a Static Network Model, namely the F-Monte Carlo method and the Destruction Spectrum method.

In particular, we have studied the variance of the estimation provided by these methods, giving exact formulas for their computation. To our knowledge, this is the first time that analytical formulations have been developed for the exact variance values of these two methods.

We have also performed a study of the computational cost of these techniques, starting from the observation that the most costly operation corresponds to the evaluation of the network structure function. This allowed us to also develop analytical formulas for the computational cost.

Numerical experiments validated the formulations developed in this paper, and showed their usefulness for comparing two performing techniques (and also to compare them to the Standard Monte Carlo method, which is not as efficient but serves as a reference point for the efficiency of reliability estimation).

Table 4 summarizes the main contributions of the article.

Future work could include performing this analysis for other estimation methods from the literature, as well as using the insights provided by the analytical formulations for improving the techniques, attaining still better efficiency, both in terms of precision and computational cost.

Table 4: Contributions of the Article

Contribution	Details	Methods Involved
Main topics and methods under analysis	The paper discusses and analyzes the F-Monte Carlo and Destruction Spectrum methods for estimating unreliability in the homogeneous case of the Static Network Model.	F-Monte Carlo and Destruction Spectrum
Development of analytical variance formulas	For the first time in the literature, variance formulas for F-Monte Carlo and Destruction Spectrum are developed. These formulas are useful for studying the accuracy of these methods.	F-Monte Carlo and Destruction Spectrum
Development of analytical computational complexity formulas	Analytical formulas for the computational complexity of F-Monte Carlo and Destruction Spectrum are developed, based on the observation that the most time-consuming operation in their implementations is the evaluation of the network structure function.	F-Monte Carlo and Destruction Spectrum
Comprehensive comparison with Standard Monte Carlo	Both, F-Monte Carlo and Destruction Spectrum are compared with Standard Monte Carlo to evaluate their efficiency, showing that the variance of Standard Monte Carlo is much higher (its accuracy is much lower) than that of both methods.	F-Monte Carlo, Destruction Spectrum and Standard Monte Carlo
Experimental validation of Analytical Formulas	A series of computational experiments are performed to validate the analytical formulas developed in the article, showing good agreement between the formulas and the empirical results.	F-Monte Carlo, Destruction Spectrum and Standard Monte Carlo

References

[1] J. Li, L. Dueñas-Osorio, C. Chen, and C. Shi, "Connectivity reliability and topological controllability of infrastructure networks: A comparative assessment," *Reliability Engineering and System Safety*, vol. 156, pp. 24–33, 2016.

[2] J. Johansson, H. Hassel, and E. Zio, "Reliability and vulnerability analyses of critical infrastructures: Comparing two approaches in the context of power systems," *Reliability Engineering & System Safety*, vol. 120, pp. 27 – 38, 2013.

[3] R. Guidotti, P. Gardoni, and N. Rosenheim, "Integration of physical infrastructure and social systems in communities' reliability and resilience analysis," *Reliability Engineering & System Safety*, vol. 185, pp. 476 – 492, 2019.

[4] D. Raad, A. Sinske, and J. Van Vuuren, "Robust multi-objective optimization for water distribution system design using a meta-metaheuristic," *International Transactions in Operational Research*, vol. 16, no. 5, pp. 595–626, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-3995.2009.00705.x

[5] S. Dharmaraja, R. Vinayak, and K. S. Trivedi, "Reliability and survivability of vehicular ad hoc networks: An analytical approach," *Reliability Engineering & System Safety*, vol. 153, pp. 28 – 38, 2016.

[6] M. Macchi, M. Garetti, D. Centrone, L. Fumagalli, and G. P. Pavirani, "Maintenance management of railway infrastructures based on reliability analysis," *Reliability Engineering & System Safety*, vol. 104, pp. 71 – 83, 2012.

[7] J. E. Muriel-Villegas, K. C. Alvarez-Uribe, C. E. Patiño-Rodríguez, and J. G. Villegas, "Analysis of transportation networks subject to natural hazards - insights from a Colombian case," *Reliability Engineering & System Safety*, vol. 152, pp. 151 – 165, 2016.

[8] N. Goldbeck, P. Angeloudis, and W. Y. Ochieng, "Resilience assessment for interdependent urban infrastructure systems using dynamic network flow models," *Reliability Engineering & System Safety*, vol. 188, pp. 62 – 79, 2019.

[9] I. Tien and A. D. Kiureghian, "Algorithms for Bayesian network modeling and reliability assessment of infrastructure systems," *Reliability Engineering & System Safety*, vol. 156, pp. 134 – 147, 2016.

[10] S. Rath, M. Gendreau, and W. J. Gutjahr, "Bi-objective stochastic programming models for determining depot locations in disaster relief operations," *International Transactions in Operational Research*, vol. 23, no. 6, pp. 997–1023, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12163

[11] A. Rosenthal, "Computing the reliability of complex networks," *SIAM Journal on Applied Mathematics*, vol. 32, no. 2, pp. 384–393, 1977.

[12] S. J. Provan and M. O. Ball, "The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected," *SIAM Journal on Computing*, vol. 12, no. 4, pp. 777–788, 1983.

[13] L. Schäfer, S. García, and V. Srithammavanh, "Simplification of inclusion-exclusion on intersections of unions with application to network systems reliability," *Reliability Engineering and System Safety*, vol. 173, pp. 23–33, 2018.

[14] K. Kobayashi, H. Morohosi, and T. Oyama, "Applying path-counting methods for measuring the robustness of the network-structured system," *International Transactions in Operational Research*, vol. 16, no. 3, pp. 371–389, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-3995.2008.00688.x

[15] H. Kumamoto, K. Tanaka, and K. Inoue, "Efficient evaluation of system reliability by Monte Carlo method," *IEEE Transactions on Reliability*, vol. R–26, no. 5, pp. 311–315, Dec. 1977.

[16] H. Kumamoto, K. Tanaka, K. Inoue, and E. Henley, "Dagger–sampling Monte Carlo for system unavailability evaluation," *IEEE Transactions on Reliability*, vol. R–29, no. 2, pp. 122–125, June 1980.

[17] M. Easton and C. Wong, "Sequential destruction method for Monte Carlo evaluation of system reliability," *IEEE Transactions on Reliability*, vol. R–29, no. 1, April 1980.

[18] G. Fishman, "A comparison of four Monte–Carlo methods for estimating the probability of s–t connectedness," *IEEE Transactions on Reliability*, vol. R–35, no. 2, pp. 145–155, June 1986.

[19] ——, "A Monte Carlo sampling plan for estimating network reliability," *Operational Research*, vol. 34, 1986.

[20] T. Elperin, I. B. Gertsbakh, and M. Lomonosov, "Estimation of network reliability using graph evolution models," *IEEE Transactions on Reliability*, vol. 40, no. 5, pp. 572–581, Dec 1991.

[21] M. Lomonosov, "On Monte Carlo estimates in network reliability," *Probability in the Engineering and Informational Sciences*, vol. 8, pp. 245–264, 1994.

[22] S. M. Ross, "A new simulation estimator of system reliability," *Journal of Applied Mathematics and Stochastic Analysis*, vol. 7, no. 3, 1994.

[23] H. Cancela and M. El Khadiri, "A recursive variance–reduction algorithm for estimating communication–network reliability," *IEEE Transactions on Reliability*, vol. 44, no. 4, pp. 595–602, December 1995.

[24] ——, "The recursive variance–reduction simulation algorithm for network reliability evaluation," *IEEE Transactions on Reliability*, vol. 52, no. 2, pp. 207–212, June 2003.

[25] H. Cancela, M. El Khadiri, and G. Rubino, "Rare events analysis by Monte Carlo techniques in static models," in *Rare Event Simulation using Monte Carlo Methods*, G. Rubino and B. Tuffin, Eds. John Wiley & Sons, 2009, ch. 7, pp. 145–170.

[26] P. L'Ecuyer, G. Rubino, S. Saggadi, and B. Tuffin, "Approximate Zero-Variance Importance Sampling for Statistic Network Reliability Estimation," *IEEE Transactions on Reliability - TR*, vol. 60, no. 3, pp. 590–604, 2011.

[27] L. Murray, H. Cancela, and G. Rubino, "A splitting algorithm for network reliability," *IIE Transactions*, vol. 45, no. 2, pp. 177–189, 2013.

[28] 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*, vol. 17, no. 1, pp. 23–45, 2003.

[29] ——, "The cross–entropy method for network reliability estimation," *Annals of Operations Research*, vol. 134, pp. 101–118(18), February 2005.

[30] Z. I. Botev, P. L'Ecuyer, R. Simard, and B. Tuffin, "Static network reliability estimation under the Marshall-Olkin copula," *ACM Transactions on Modeling and Computer Simulation*, vol. 26, no. 2, p. Article 14, 2016.

[31] E. Canale, F. Robledo, P. Romero, and P. Sartor, "Monte carlo methods in diameter-constrained reliability," *Optical Switching and Networking*, vol. 14, pp. 134–148, 2014, special Issue on RNDM 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1573427714000691

[32] N. L. Dehghani, J. Song, and M. Pandey, "Adaptive surrogate-based network reliability analysis through integration of bayesian additive regression trees and monte carlo," *Reliability Engineering & System Safety*, vol. 213, p. 107701, 2021.

[33] J. Chan, "A comparative study on sampling-based algorithms for network reliability assessment," *Reliability Engineering & System Safety*, vol. 230, p. 108940, 2023.

[34] D. H. Huang, "Network reliability of a stochastic flow network by wrapping lp-based minimal-path models into monte-carlo," *European Journal of Operational Research*, 2024, in press.

[35] C. Ding, "Sampling and active-learning methods for efficient network reliability estimation," *IEEE Transactions on Reliability*, 2024, early Access.

[36] X. Zhou, Y. Li, and K. Zhang, "Dynamic reliability of infrastructure networks using augmented monte carlo simulation," *Applied Mathematical Modelling*, vol. 102, pp. 532–549, 2022.

[37] I. Bjerkebæk and T. Østerdal, "Reliability assessment combining importance resampling and the cross-entropy method," *Methodology and Computing in Applied Probability*, 2024, online first.

[38] H. Pérez-Rosés, "Sixty years of network reliability," *Mathematics in Computer Science*, vol. 12, no. 3, pp. 275–293, 2018.

[39] I. B. Gertsbakh and Y. Shpungin, *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*, 1st ed. USA: CRC Press, Inc., 2009.

[40] G. S. Fishman, *Monte Carlo : concepts, algorithms, and applications*, ser. Springer series in operations research. New York: Springer-Verlag, 1996.

[41] I. Gertsbakh, R. Rubinstein, Y. Shpungin, and R. Vaisman, "Permutational methods for performance analysis of stochastic flow networks," *Probability in the Engineering and Informational Sciences*, vol. 28, no. 1, pp. 21–38, 2014.

[42] F. Samaniego, "On closure of the IFR class under formation of coherent systems," *Reliability, IEEE Transactions on*, vol. R-34, pp. 69 – 72, 05 1985.

[43] F. J. Samaniego, *System Signatures*. John Wiley & Sons, Ltd, 2008. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470061572.eqr077

[44] Z. I. Botev, P. L'Ecuyer, and B. Tuffin, "Static network reliability estimation under the marshall–olkin dependence model," *Methodology and Computing in Applied Probability*, vol. 17, no. 2, pp. 421–447, 2015.

[45] A. Moshnikova, "Evaluation of network reliability and element importance metrics," in *CEUR Workshop Proceedings*, vol. 2702, 2020, pp. 122–131.

[46] J. Jenczyk, "Regular dodecahedron-based network structures," *Symmetry*, vol. 16, no. 2, p. 300, 2024.