On the Reliability Estimation of Stochastic Binary Systems

Héctor Cancela[1], Leslie Murray[2], Franco Robledo[1], Pablo Romero[1], and Pablo Sartor[3]

[1]Facultad de Ingeniería, Universidad de la República. Julio Herrera y Reissig 565, PC 11300. Montevideo, Uruguay
[2]FCEIA, Universidad Nacional de Rosario. Riobamba 245 Bis, 2000 Rosario, Argentina
[3]IEEM Business School, Universidad de Montevideo. Lord Ponsonby 2542, PC 11600. Montevideo, Uruguay

This is the submitted version of the following article: Cancela, H., Murray, L., Robledo, F., Romero, P. and Sartor, P. (2022), On the reliability estimation of stochastic binary systems. Intl. Trans. in Op. Res., 29: 1688-1722. , which has been published in final form at https://doi.org/10.1111/itor.13034.

Abstract

A stochastic binary system is a multi-component on-off system subject to random independent failures on its components. After potential failures, the state of the subsystem is ruled by a logical function (called structure function) that determines whether the system is operational or not.

Stochastic binary systems (SBS) serve as a natural generalization of network reliability analysis, where the goal is to find the probability of correct operation of the system (in terms of connectivity, network diameter or different measures of success). A particular subclass of interest is stochastic monotone binary systems (SMBS), which are characterized by non-decreasing structure.

We explore the combinatorics of SBS, which provide building blocks for system reliability estimation, looking at minimal non-operational subsystems, called mincuts. One key concept to understand the underlying combinatorics of SBS is duality. As methods for exact evaluation take exponential time, we discuss the use of Monte Carlo algorithms. In particular, we discuss the F-Monte Carlo method for estimating the reliability polynomial for homogeneous SBS, the Recursive Variance Reduction (RVR) for SMBS, which builds upon the efficient determination of mincuts, and three additional methods that combine in different ways the well–known techniques of Permutation Monte Carlo and Splitting. These last three methods are based on a stochastic process called Creation Process, a temporal evolution of the SBS which is static by definition. All the methods are compared using different topologies, showing large efficiency gains over the basic Monte Carlo scheme.

1 Introduction

The goal in reliability analysis is to determine the probability of correct operation of a system; see Colbourn (1999). Using foundational results on Computational Complexity, Arnie Rosenthal formally proved the hardness of the network reliability evaluation; see Cook (1971); Karp (1972) and Rosenthal (1977). Provan and Ball (1983) proved that even the reliability evaluation of a source-terminal communication system is a hard problem as well. The computational complexity is inherited in other more complex models, such as diameter-constrained reliability models; see Canale et al. (2015c).

While there has been a remarkable advance in the understanding of reliability analysis in the context of communication networks (see for instance Pérez-Rosés (2018) for a recent review), including the development of new evaluation algorithms, metrics and evaluation methods such us the works introduced by Schäfer et al. (2018) and Kobayashi et al. (2009), the complexity results show that (unless $\mathcal{P} = \mathcal{NP}$) exact evaluation will always take exponential time.

At the same time, the practical applications of network reliability analysis are steadily growing in frequency and diversity. For instance, in the area of civil infrastructures we can find many recent papers, we discuss some of them to give an idea of the issues at stake. The work by Li et al. Li et al. (2016) presents a comparative study of two important measures, the connectivity reliability and the topological controllability of infrastructure systems in terms of topology, robustness, and node importance, taking eight city-level power transmission networks and thousands of artificial networks as examples, and discussing how these measures can improve reliability-based design of infrastructure networks increasingly dependent on information systems. Tien and Kiureghian (2016) discuss how to represent large, complex infrastructure networks using Bayesian networks to represent efficiently the reliability behavior of these systems with a compact representation. Guidotti et al. (2019) discuss another reliability model variant, based on probabilistic flow-based networks for representing the capacity and demand of critical infrastructure, to evaluate the probability of reduction or loss of functionality of the infrastructure, integrating physical infrastructure and social systems to predict the change in demand after critical events. The proposal is exemplified by model of the potable water network of Seaside, Oregon considering a seismic event as the damaging event. Johansson et al. (2013) discuss both reliability and vulnerability analysis of critical infrastructures, discussing their complementary contributions towards understanding system behavior both theoretically and in a case study based on the IEEE RTS96 electric power test system. Goldbeck et al. (2019) study urban infrastructure systems and discuss the limitations of modeling separately different but interdependent systems, proposing instead an integrated, dynamic modeling and simulation framework combining network and asset representations of infrastructure systems, using tree–based scenarios to consider dependencies and proposing a case study based on London's metro and electric power networks and evaluating their resilience to a local flooding incident. Raad et al. (2009) discuss the design of an urban water distribution system involving a tradeoff between system cost and reliability objectives. Another interesting work is Dharmaraja et al. (2016), which studies vehicular ad hoc networks and their reliability and survivability as a function of reliable hardware and channel availability, using reliability block diagrams and exploring the survivability of the network by Markov chains and Markov reward models. Macchi et al. (2012) study railway track maintenance management, building a reliability model of the railway system to identify the most critical items which are classified in families having similar reliability targets. The methods are implemented and tested in practical case studies arising from the Italian public company Rete Ferroviaria Italiana. Muriel-Villegas et al. (2016) analyze transportation networks subject to natural hazards, and in particular study the connectivity reliability and vulnerability of inter-urban transportation in the case of Colombia, focusing in the connectivity of remote populations in the case of disasters such as floods, taking as case study the state of Antioquia, Colombia, based on historical records from the 2010 to 2011 rainy season. Rath et al. (2016) study how to determine depot location in disaster relief operations, using a bi-objective model where the accessibility of the road network is given by a probability distribution. Masri et al. (2019) discuss a network optimization problem where the nodes can communicate using a predetermined set of connections characterized by a capacity, a lead time, and a reliability, developing a multiobjective model to optimize both delays and average reliability. These and other practical applications motivate the need for flexible modeling mechanisms and for fast evaluation of these metrics. In consequence, Monte Carlo estimation methods have been developed as a practical alternative; see Botev et al. (2016); Canale et al. (2014); Cancela and Khadiri (2003); L'Ecuyer et al. (2011); Murray et al. (2013). Other efficient Monte Carlo methods in the context of performability and robust network design can be found in Cancela et al. (2014); Saling and White Jr. (2013). The theory of stochastic binary systems can be seen as a generalization of network reliability models; as the system structure function in an SBS can be any arbitrary boolean function, while in classical network reliability it is usually some variant of a connectivity function.

In this paper, we look at some theoretical results on SBS and we discuss Monte Carlo evaluation. Our goal is to extend the success of well–known reliability estimation methods previously applied to network models to arbitrary SMBS. Some of our previous works, such as s Canale et al. (2015a, 2014), represent our point of departure for this work; other papers, such asCancela et al. (2018) and Romero (2016, 2019), can be useful as SBMS methodological background.

The main contributions of this paper are summarized as follows:

- The combinatorics of Stochastic Monotone Binary Systems is revisited.
- We exploit duality and the optimal algorithm to find mincuts in order to apply the Recursive Variance Reduction (RVR) method for the reliability evaluation of SMBS.
- We also discuss the F-Monte Carlo (FMC) method for estimating the reliability

polynomial in the case of homogeneous SBS.

- The interplay between SMBS and some dynamical reliability models, based on the classical Creation Process by Elperin et al. (1991), is explored.
- Based on the prior item, some Permutational and Splitting–like methods, the last of which is a novel proposal, are analyzed.
- The performance of these Monte Carlo estimation methods is evaluated by computational experiments in different scenarios.

The document is organized in the following manner. Section 2 presents the concept of stochastic binary system, its reliability, and examples. The underlying combinatorics and complexity of the problem is studied in Section 3 where we also introduce an optimal algorithm to find mincuts. The generalization to SMBS of FMC, RVR, and of the three methods derived from the Creation Process,is considered as a pointwise reliability estimation in Section 4. The performance of all the methods introduced is illustrated via several tests in Section 5. Finally, Section 6 presents concluding remarks and trends for future work.

2 Concepts

The following terminology is adapted from Ball (1986).

Definition 1. A *stochastic binary system* (SBS) is a triad (S, p, ϕ):

- $S = \{1, \dots, m\}$ is a ground set of *components*,
- $p = (p_1, \dots, p_m) \in [0, 1]^m$ has their *elementary reliabilities*, and
- $\phi : \{0, 1\}^m \to \{0, 1\}$ is the logical rule or *structure* of the system.

Definition 2 (Reliability/Unreliability). Let $\mathcal{S} = (S, p, \phi)$ be an SBS, and consider a random vector $X = (X_1, \dots, X_m)$ with independent coordinates governed by Bernoulli random variables such that $P(X_i = 1) = p_i$.
The *reliability* of $\mathcal{S}$ is the probability of correct operation:

$$r_{\mathcal{S}} = P(\phi(X) = 1) = E(\phi(X)). \tag{1}$$

The *unreliability* of $\mathcal{S}$ is $q_{\mathcal{S}} = 1 - r_{\mathcal{S}}$.

Definition 3 (Pathsets/Cutsets). Let $\mathcal{S} = (S, p, \phi)$ be an SBS. A possible state $x \in \{0, 1\}^m$ is a *pathset* (resp. *cutset*) if $\phi(x) = 1$ (resp., if $\phi(x) = 0$).

Pathsets and cutsets are also known as *up* and *down* states respectively. The binary set $\{0, 1\}$ is equipped with the partial order, defined by $0 \leq 0$, $0 \leq 1$ and $1 \leq 1$. The set $\{0, 1\}^m$ inherits a natural order in the Cartesian product:

Definition 4 (Canonical Order). If $x = (x_1, \dots, x_m) \in \{0, 1\}^m$ and $y = (y_1, \dots, y_m) \in \{0, 1\}^m$, we denote $x \leq y$ if and only if $x_i \leq y_i$ for all $i = 1, \dots, m$.

Given two partially ordered sets A and B, a function $f : A \to B$ is monotone if $f(a_1) \leq f(a_2)$ whenever $a_1 \leq a_2$. As usual, we denote $y < x$ if $y \leq x$ and $y \neq x$. Let us denote $\vec{0}_m$ (resp. $\vec{1}_m$) to be the binary word with all bits set to 0 (resp. to 1), and e_i to be the binary word with all bits in 0 but the bit in position i set to 1.

Definition 5 (Stochastic Monotone Binary System). The triad $\mathcal{S} = (S, p, \phi)$ is a *stochastic monotone binary system* (SMBS) if $\phi(\vec{0}_m) = 0$, $\phi(\vec{1}_m) = 1$ and the structure function $\phi : \{0,1\}^m \to \{0,1\}$ is non-decreasing, where both the domain and codomain are equipped with the canonical order for binary words.

Definition 6 (Minpaths/Mincuts). Let $\mathcal{S} = (S, p, \phi)$ be an SMBS. A pathset x is a *minpath* if $\phi(y) = 0$ for all y such that $y < x$. A cutset y is a *mincut* if $\phi(x) = 1$ for all x such that $y < x$.

If we repair the component s from the state x, the result is denoted by $x(s = 1)$. Analogously, $x(s = 0)$ denotes the setting of the component s to non-operational state.

Definition 7 (Essential/Irrelevant Components). Let $\mathcal{S} = (S, p, \phi)$ be an SBS. A component $s \in S$ is *essential* if the system is always down when s fails. The component s is *irrelevant* if the state of this particular component s does not affect the global system: $\phi(x(s = 0)) = \phi(x(s = 1))$ for all possible states $x \in \{0,1\}^m$.

An SBS is *homogeneous* if the elementary reliabilities are identical (i.e., $p_i = p$ for all i). Given a structure ϕ, we denote $\overline{x}$ to be the complementary state in bits (i.e., 0's are set to 1's and vice-versa). In particular, $\overline{\phi}(x) = 1 - \phi(x)$.

Our later analysis of monotonicity and cutsets promotes the following definition of duality found in Romero (2016):

Definition 8 (Dual System). The dual of $\mathcal{S} = (S, p, \phi)$ is another SBS with identical ground set S, elementary reliabilities $p_i^d = 1 - p_i$, and structure $\phi^d(x) = 1 - \phi(\overline{x})$, for all possible states $x \in \{0,1\}^m$. The dual is denoted by $\mathcal{S}^d = (S, 1-p, \phi^d)$.

The following examples provide an insight of the different applications of stochastic binary systems. Classical examples include a reference in the field for the interested reader.

1. K-Terminal Reliability: let $G = (V, E)$ a simple graph, and consider a terminal-set $K \subseteq V$. The ground set is E, so the links fail, and the system is up if all the nodes from K belong to the same connected component in the resulting random graph; see Ball (1980).

2. All-Terminal Reliability: choose $K = V$ in the first model. The system is up if the resulting random graph is connected.

3. Source-Terminal Reliability: choose $K = \{s, t\}$ in the first model.

4. Diameter Constrained Reliability: a diameter constraint d is added to the K-Terminal Reliability; see Petingi and Rodríguez (2001); Canale et al. (2015b, 2013). The system is up if every pair of terminals are connected by paths whose length is not greater than d.

5. k-m-Survivability: the system is up if and only if there are at least k components in operational state out of m equally reliable components. This homogeneous system is known as k-out-of-m system; see Wang et al. (2018). We will denote $\phi_{(k,m)}$ to its structure according to Canale et al. (2015a).

6. k-m-Degraded: the system is down if and only if there are at least k identical components in failure state out of m. We will denote $\psi_{(k,m)}$ to its structure. Clearly: $\psi_{(k,m)} = \phi_{(m-k,m)}$.

7. Circular k-m-balanced Survivability: the system is up if and only if at least k components are operating and the system is spatially balanced (i.e., the operational components are uniformly spread); see Endharta et al. (2018).

8. Residual Reliability: the ground set is a node-subset of a simple graph. The system is up if, after eliminating failed nodes, the operational terminal nodes belong to the same connected component; see Cancela and Urquhart (2002).

9. Node-Edge Reliability: both links and nodes fail in a random graph. The system is up if and only if the resulting subgraph is connected; see Dash et al. (2012).

10. Feasibility: consider an arbitrary integer linear program P with binary decision variables $x_1, \ldots, x_m$. The elementary reliability p_i is the likelihood of the event $x_i = 1$, or $p_i = 1/2$ if there is no available experimental data, and the structure is $\phi(x) = 1$ if x is feasible for P.

11. Forbidden Pattern: given a binary string w, the system is non-operational, i.e, $\phi_w(x) = 0$, if and only if w appears in the binary word x.

All the models can be homogeneous if we fix the elementary reliabilities $p_i = p$ for all the components of the system (otherwise, the system is heterogeneous). The reader is invited to check that the examples 1-6 are SMBS, while the examples 7-11 are non monotone.

3 Computational Complexity

In this section, the combinatorics of SMBS is reviewed. The interested reader can consult further details in Cancela et al. (2018) and Romero (2016, 2019).

Let us explore the link between an SBS and propositional logic. In this study we assume that the structure accepts a polynomial time evaluation for any state $x \in \{0,1\}^m$. Recall that a theorem-proving procedure is the first $\mathcal{NP}$-Complete decision problem established by Cook (1971). In other words, the recognition of a tautology is a hard decision problem from propositional logic.

Theorem 1. *The reliability evaluation of an arbitrary SMBS belongs to the class of $\mathcal{NP}$-Hard problems.*

Proof. Rosenthal (1977) formally proved that the reliability evaluation for the K-terminal reliability model belongs to the class of $\mathcal{NP}$-Hard computational problems. Since K-Terminal is a particular SMBS, the result follows by inclusion. □

Corollary 1. *The reliability evaluation of an arbitrary SBS belongs to the class of $\mathcal{NP}$-Hard problems.*

Theorem 2. *The determination of a cutset in an arbitrary SBS is an $\mathcal{NP}$-Complete decision problem.*

Proof. Consider an arbitrary propositional logic φ with m literals. Build the corresponding SBS with m elements and φ as the structure. Then, φ is a tautology if and only if the corresponding SBS has no cutsets. □

Let us study the problem for arbitrary SMBS. First, three elementary results:

Lemma 1. *The dual of the dual is the original system.*

Proof. $\phi^{d^d}(x) = 1 - \phi^d(\overline{x}) = 1 - (1 - \phi(x)) = \phi(x)$. □

Lemma 2. *The dual of an SMBS is another SMBS.*

Proof. Consider arbitrary states $x \leq y$ and a monotone structure ϕ. Since $\overline{x} \geq \overline{y}$, we get that $\phi(\overline{y}) \leq \phi(\overline{x})$. Therefore: $\phi^d(x) = 1 - \phi(\overline{x}) \leq 1 - \phi(\overline{y}) = \phi^d(y)$. □

Since the elementary reliabilities are identical in the dual system, we get the following result:

Corollary 2. *Consider a homogeneous SMBS. Then, a state x is the pathset with maximum probability if and only if $\overline{x}$ is a cutset with maximum probability in the dual.*

Proof. First, assume that x is a pathset with maximum probability. Then $\phi(x) = 1$, and $\phi^d(\overline{x}) = 1 - \phi(x) = 0$, so $\overline{x}$ is a cutset in the dual system. Assume that x has precisely r elements in operational state. Then $P(x) = p^r(1-p)^{m-r}$. In the dual the elementary reliability equals $1 - p$. Then, in the dual the probability is $P(\overline{x}) = (1-p)^{m-r}p^r$, identical to the probability of state x in the original system. The converse holds by Lemma 1. □

From Corollary 2, we can study pathsets instead of cutsets, and the results under monotonicity hold. Recall that we want to find the cutset with maximum probability in SMBS. Instead, we study pathsets:

Proposition 1. The determination of a pathset with maximum probability in arbitrary SMBS belongs to the class of $\mathcal{NP}$-Hard problems.

Proof. Consider the K-Terminal Reliability model, in the homogeneous case. Since the model is homogeneous, a pathset with maximum probability is precisely a minimum cardinality minpath. But in the K-Terminal model, this is the Steiner Problem in Graphs, which belongs to the $\mathcal{NP}$-Hard class as proved by Karp (1972). □

Theorem 3. *The determination of a cutset with maximum probability in arbitrary SMBS belongs to the class of $\mathcal{NP}$-Hard problems.*

Proof. Combine Proposition 1 with Lemma 2 and Corollary 2. □

Theorem 3 is a negative result that has a deep impact on the understanding of SMBS. In the homogeneous Source-Terminal Reliability model, finding the mincut with maximum probability is precisely the minimum cardinality s-t cutset. Using the theory of flows in networks, it is known that the mincut can be determined from the maximum flow between s and t with unit capacities in the links. Therefore, Theorem 3 discards any possibility of the existence of a maximum flow theory in SBS, unless $\mathcal{P} = \mathcal{NP}$.

The following result, in contrast with Theorem 2, gives an efficient method for finding mincuts in SMBS. The algorithim improves from the $O(m^2)$ method discussed in a previous conference paper Canale et al. (2015a). The result will be useful to develop an efficient Monte Carlo based reliability estimation method (RVR) for SMBS:

Theorem 4. *A mincut can be found using m rule-evaluations in an arbitrary SMBS.*

Proof. The evidence is Algorithm 1. Clearly, it requires m evaluations, and it returns a cutset x. We will prove the statement in two steps:

(1) The state x is a mincut.

(2) Algorithm 1 is optimal in terms of rule-evaluations.

For (1), suppose that the output x is not a mincut. Therefore, there exists some $j \in \{1, \ldots, m\}$ such that $\phi(x + e_j) = 0$. Let us denote $x^{(j)}$ the state for the iteration j in the **for**-loop. Observe that x is (possibly) increased in each iteration. Therefore, $x^{(j)} \leq x \leq x + e_j$. Since $\phi(x + e_j) = 0$ and ϕ is monotone, we get that $\phi(x^{(j)}) = 0$. But in this case the j–th bit would have been set to 1, and this bit is set to 0 in the output. This is impossible, since the bits in x are only increased during the execution of Algorithm 1.
For (2), observe that in the worst case the null vector $x = 0$ is a mincut. The only way to determine that 0 is a mincut is to test $\phi(e_i) = 1$ for all possible canonical vectors $\{e_i\}_{i=1,\ldots,m}$, and this requires m rule-evaluations. □

Theorem 4 is constructive. It provides an interplay with propositional logic. In other words, the linear-time algorithm from Theorem 4 is optimal, and serves to find a *minimal false assignment* in the most efficient way.

Let us close this section with three elementary results for the determination of essential and irrelevant components. The determination of irrelevant components is a key aspect in network reliability models, so it is expected to play a central role in understanding the underlying combinatorics of SBS.

Proposition 2. Consider an arbitrary SBS, given by the triad $\mathcal{S} = (S, p, \phi)$, and let $s \in S$ be an arbitrary component. The decision problem that determines whether s is irrelevant or not belongs to the class of $\mathcal{NP}$-Hard computational problems.

Algorithm 1 $x = Mincut(\phi, S)$

```
$m \leftarrow |S|$
$x \leftarrow \vec{0}_m$
for $i = 1$ to $m$ do
    $y \leftarrow x + e_i$
    if $\phi(y) = 0$ then
        $x \leftarrow y$
    end if
end for
return $x$
```

Figure 1: Pseudocode for $Mincut$ method.

Proof. By its definition, s is not irrelevant if there exists a state x such that $\phi(x(s = 0)) \neq \phi(x(s = 1))$. This means that either $x(s = 0)$ is a cutset, or $x(s = 1)$ is a cutset. In both cases, a cutset for the SBS must be found. As a consequence, the recognition of irrelevant components is at least as hard as the determination of a cutset. The result follows from Theorem 2. □

Another complexity result holds for essential components:

Proposition 3. Consider an SBS, given by the triad $\mathcal{S} = (S, p, \phi)$, and let $s \in S$ be an arbitrary component. The decision problem that determines whether s is essential or not belongs to the class of $co - \mathcal{NP}$-Hard computational problems.

Proof. From well-known results in complexity theory (see for instance Arora and Barak (2009)), the general Tautology problem (i.e., finding if an expression in propositional logic is true independently of the values of the individual terms) is co$\mathcal{NP}$-complete.

Let ψ be an arbitrary propositional logic expression on terms $s_1, \ldots, s_{n-1}$ (taking 1 as the True value, and 0 as the False value). Define a structure function ϕ over $s_1, \ldots, s_{n-1}, s$, where if $s = 1$ then ϕ is 1, and where if $s = 0$, then $\phi = 0$ if ψ is True, and $\phi = 1$ if ψ is False.

Then if s is an essential component of ϕ, when s is 0 ϕ must be 0 irrespective of the values of $s_1, \ldots, s_{n-1}$, meaning that ψ is a tautology. If we can solve the essential component problem, we can then solve the Tautology problem with the same computational effort.

This means that determining an essential component is at least as complex as the Tautology problem, which is co$\mathcal{NP}$-complete; so that the essential component determination belongs to the co$\mathcal{NP}$-hard class.

□

In practice, it is highly desired to determine whether a component of a real system is essential or not. However, if we do not have further information about the system, Proposition 3 tells us that we cannot decide efficiently whether the component is essential or not, unless $\mathcal{NP} = co\mathcal{NP}$ (an open question in complexity theory).

The determination of essential components in arbitrary SMBS is straightforward:

Proposition 4. Consider an arbitrary SMBS, given by the triad $\mathcal{S} = (S, p, \phi)$, and let $s \in S$ be an arbitrary component. The decision problem that determines whether s is essential or not can be performed in a single rule-evaluation.

Proof. If $\phi(\vec{1}_m - e_s) = 1$, by its definition s is not essential. On the other hand, if $\phi(\vec{1}_m - e_s) = 0$ by monotonicity we have that $\phi(x - e_s) = 0$ for all possible states $x \in \{0, 1\}^m$, and s is essential. □

4 Monte Carlo algorithms

Monte Carlo simulation methods consist in estimating an unknown measure or merit figure, whose value is the expectation of a given random variable, by generating N samples of this variable and computing the sample mean, variance, and a confidence interval for the value of interest; see Fishman (1996). In the case of reliability estimation, a Monte Carlo estimation works by sampling the state of every component of the system, and evaluating the operational/failed state of the system as a whole.

In this section, we discuss the basic Monte Carlo method. As this basic method runs into performance problems in the case of high reliability values (the so-called rare-event case discussed by Rubino and Tuffin (2009)), two other alternatives designed for improved efficiency, the RVR and the FMC methods, are also presented.

4.1 Crude Monte Carlo (CMC)

The basic Monte Carlo method (also called Crude Monte Carlo, CMC), consists in generating N independent samples $X^1, \ldots, X^N$ of a random variable with finite mean $E(X)$. The Strong Law of Large Numbers implies that the average $\overline{X}_N$ converges almost surely to $E(X)$, with variance $Var(\overline{X}_N) = Var(X)/N$.

For reliability estimation, we consider the random vector of the components' states, $X = (X_1, \ldots, X_m)$, such that $P(X_i = 1) = p_i$ is the component reliability for each i. Then it is necessary to generate N independent samples $X^1, \ldots, X^N$ of X. A point estimation of the reliability is computed by

$$r_{CMC} = \frac{1}{N} \sum_{i=1}^{N} \phi(X^i). \tag{2}$$

This estimator is unbiased, with variance $Var(r_{CMC}) = r_{\mathcal{S}}(1-r_{\mathcal{S}})/N$. One important advantage of CMC is that it does not depend on any particular feature of the system, so that it can be applied to general SBS. A pseudocode of the CMC implementation for pointwise reliability estimation is presented in Algorithm 2.

The precision of the estimator r_{CMC} is mainly determined by its variance. See, for example, that the half width of its Confidence Interval is $z_{\alpha/2} \times Var(r_{CMC})^{1/2}$,

```
Algorithm 2 $r_{CMC} = CMC(\phi, S, p, N)$
Sum ← 0
for $i = 1$ to $N$ do
    $X_i \leftarrow \vec{0}_m$
    for $j = 1$ to $m$ do
        $U_j \sim \mathcal{U}[0, 1]$
        if $U_j < p_j$ then
            $X_i \leftarrow X_i + e_j$
        end if
    end for
    $Sum \leftarrow Sum + \phi(X_i)$
end for
return $r_{CMC} = Sum/N$
```

Figure 2: Pseudocode for CMC method.

where $z_{\alpha/2}$ is the confidence level. Thus, the interval size is mainly determined by the estimator variance or, similarly, by its standard deviation. It is also useful to consider some indicator of relative precision, like, for example, the ratio $Var(r_{CMC})^{1/2}/r_{CMC}$, that is usually accepted as a relative error. For the unreliability estimation this error reads $Var(q_{CMC})^{1/2}/q_{CMC}$, where $Var(q_{CMC}) = Var(r_{CMC})$.

In the case of highly reliable SBSs (i.e, when failures are rare events), this relative error is larger and much more significant if we estimate the small (close to 0) unreliability, rather than the large (close to 1) reliability. This is the reason why in a comparative setting, in which highly reliable systems are considered, comparisons are based on the unreliability. See that, when an SBS becomes more reliable, both terms, $Var(q_{CMC})^{1/2}$ and q_{CMC} tend to zero, however q_{CMC} does it faster than the variance. This is the reason why, as an SBS becomes more reliable, CMC loses efficiency, the relative error grows boundlessly, and variance reduction methods, like the ones introduced in the following sections, are necessary to be able to estimate the unreliability measures with sufficient precision.

4.2 Recursive Variance Reduction (RVR)

The RVR method has originally been proposed to solve network reliability problems, with good results; see Cancela and El Khadiri (1995); Cancela et al. (2012). It is based on a recursive decomposition of the state space of a monotone system by conditioning over the state of the components of a mincut. It can be adapted to the case of an SMBS, but it is not applicable directly to a non monotone system.

The main idea is to find a mincut of the system at each step. The probability that all components in the cutset are down is added to the unreliability estimator; the remaining states are partioned in sub-events, corresponding to the index of the first component of the cutset to be operational. Using conditional probabilities, the method randomly chooses one of these sub-events, and restarts the procedure until a trivial case is met. To present the method, we need to introduce the following notation, closely following the one used by Cancela and El Khadiri (1995), adapted to this context of general SMBS. It is worth to remark that we will identify indistinctly a cutset $x \in \{0,1\}^m$ with the set $A_x = \{i : x_i = 0\}$.

1. The random vector $X = (X_1, \ldots, X_m)$ such that $X_1, \ldots, X_m$ are independent Bernoulli variables and $P(X_i = 1) = p_i$ is the state vector of the SMBS under study.
2. $C = (a_1, \ldots, a_{|C|})$ is a mincut of the system under study. C can be found efficiently using Theorem 4.
3. A_C denotes the event "all the components in C fail".
4. $q_C = P(A_C) = \prod_{j=1}^{|C|}(1 - p_{a_j})$.
5. $A_i = \{a_1, \ldots, a_i\}$ is the subset of the first i elements belonging to C.
6. B_i is the event "a_i is up but a_j fails for all $j < i$".
7. ϕ_{B_i} is the structure function of the system restricted by B_i (i.e, where the a_i component of X is fixed to 1 and the a_j components of X are fixed to 0 for all $j < i$).
8. $P(B_i) = p_{a_i} \prod_{j<i}(1 - p_{a_j})$.
9. V is a discrete random variable such that $P(V = i) = P(B_i|\bar{A}_C) = P(B_i)/(1 - q_C)$, where $\bar{A}_C = \cup_{i=1}^{|C|} B_i$ is the complement of A_C.
10. $Y_i = 1 - E(\phi_{B_i}(X))$ is the unreliability of the system conditional on the event B_i.

Let $X = 1_A$ denote the Bernoulli variable such that $X = 1$ if and only if the event A occurs. Consider now the following random variable:

$$Z = q_C + (1 - q_C) \sum_{i=1}^{|C|} 1_{\{V=i\}} Y_i. \qquad (3)$$

The reader can notice that the subsystems with structure function ϕ_{B_i} are SMBS as well, since the restriction of a monotone function is monotone. Applying the total probability theorem, we can check that Z is an unbiased estimator of q_S. What is more, a mean sample of Z has smaller variance than CMC, therefore it is more accurate (since both are unbiased estimators). The proofs from Cancela and El Khadiri (1995) apply directly, substituting SMBS instead of networks as they do not depend any specificity of the particular K-terminal structure other than monotonicity. The next step consists in observing that the same idea of Expression (3) can be recursively applied to each Y_i, leading to the following recursive operator Q as the unreliability estimator:

$$Q(X) = \begin{cases} 1, & \text{if } \phi(X) \equiv 0; \\ 0, & \text{if } \phi(X) \equiv 1, \text{ or} \\ q_{C_{S'}} + (1 - q_{C_{S'}}) & \sum_{i=1}^{|C_{S'}|} 1_{\{V=i\}} Q_{B_i}(X). \end{cases} \tag{4}$$

where the mincut $C_{S'}$ is recursively found for each subsystem using Theorem 4 over the whole component-set S', and Q_{B_i} is the call to the recursive operator Q over the structure function ϕ_{B_i} defined above (i.e, where the a_i component of X is fixed to 1 and the a_j components of X are fixed to 0 for all $j < i$). Observe that the recursion fixes some elements of the cutset (a number of them in down state, and the first element i that is up as well). In other words, the historical information during the recursion is recorded in the conditional measure. As the subsystems successively considered have diminishing numbers of components, the termination is guaranteed. If $Q_1, \ldots, Q_N$ is an independent and identically distributed sample of $Q(X)$, an unbiased estimator for q_S is:

$$q_{RVR} = \frac{1}{N} \sum_{i=1}^{N} Q_i, \tag{5}$$

The method is presented in pseudocode form in Figure 3.

The block of Lines 1-6 in Algorithm 4 is a halting test that determines if the system can be evaluated trivially (ϕ is constant, either 0 or 1), or not. Function $MinCut$ is called in Line 7, and a mincut C is obtained. The probability of the mincut q_C is found in Line 8 as the product of the failure probabilities of its components. If not all the components of the cutset fail, then there is a first component $a_i, i \in \{1, \ldots, |C|\}$ that is up. The probability distribution vector $p' = (p'_1, \ldots, p'_{|C|})$ is computed in Line 9, where $p'_i = P(V = i)$ and V is a discrete random variable such that $P(V = i) = P(B_i | \bar{A}_C) = P(B_i)/(1 - q_C)$. The recursive part of RVR is precisely Line 10, where Expression (4) is considered. The result is an unbiased reliability estimation of an SMBS, with lower variance than CMC.

4.3 F-Monte Carlo (FMC) in the homogeneous case

We now concentrate in the particular case of homogeneous SMBS, where all components have identical probabilities of operation $p_i = p$. Let $f(x)$ be the function counting the number of bits 0 in the binary word x:

$$f(x) = m - \sum_{i=1}^{m} x_i \tag{6}$$

Consider the following partition of $\{0, 1\}^m$:

$$S_i = \{x \in \{0, 1\}^m : f(x) = i, \phi(x) = 1\}, \forall i = 0, \ldots, m.$$

The collection of sets S_i represents the subset of operational binary words with precisely i bits set to 0. Consider the numbers $F_i = |S_i|$, and let $\mathcal{P}(S)$ be the power-set

Algorithm 3 $q_{RVR} = RVRMC(\phi, S, p, N)$

```
$Sum \leftarrow 0$
for $i = 1$ to $N$ do
    $Sum \leftarrow Sum + RVR(\phi, S, p)$
end for
return $q_{RVR} = Sum/N$
```

Algorithm 4 $Q = RVR(\phi, S, p)$

```
if $\phi(x) = 1 \, \forall x$ then
    return $Q = 0$
end if
if $\phi(x) = 0 \, \forall x$ then
    return $Q = 1$
end if
$C \leftarrow MinCut(\phi, S)$
$q_C \leftarrow \prod_{j=1}^{|C|}(1 - p_{a_j})$
$(V, B_i, A_i, p') \leftarrow SampleCut(C, p)$
return $Q = q_C + (1 - q_C) \times RVR(\phi_{B_i}, S - A_i, p')$
```

Figure 3: Pseudocode for RVR method.

Algorithm 5 $r_{FMC}(p) = FMC(\phi, S, p)$

```
for $i = 0$ to $m$ do
    $Sum_i \leftarrow 0$
    for $j = 1$ to $N$ do
        $X_i^j \leftarrow Choose(m - i, S)$
        $Sum_i \leftarrow Sum_i + \phi(X_i^j)$
    end for
    $\widehat{F}_i \leftarrow \binom{m}{i} Sum_i / N$
end for
$r_{FMC}(p) \leftarrow \sum_{i=0}^{m} \widehat{F}_i p^{m-i}(1-p)^i$
return $r_{FMC}$
```

Figure 4: Pseudocode for FMC method.

S. It is possible to express the reliability of the SMBS as a polynomial in the scalar variable $p \in [0, 1]$, with coefficients F_i:

$$\begin{aligned} r_{\mathcal{S}}(p) &= \sum_{T \in \mathcal{P}(S) : \phi(T) = 1} P(X = T) \\ &= \sum_{i=0}^{m} \sum_{T \in S_i} P(X = T) \\ &= \sum_{i=0}^{m} \sum_{T \in S_i} p^{m-i}(1-p)^i \\ &= \sum_{i=0}^{m} F_i p^{m-i}(1-p)^i. \end{aligned}$$

The function $r_{\mathcal{S}}(p)$ is the *reliability polynomial* of the SMBS S. Then, under identical failure probabilities the reliability polynomial computation is equivalent to finding the F*-vector* $F = (F_0, F_1, \ldots, F_m)$, i.e, solving $m + 1$ counting problems.

The key idea of F-Monte Carlo (FMC) is to pick random states $x \in \{0, 1\}^m$ such that $f(x) = i$. Then, we count the ones that belong to the set S_i in order to estimate F_i. Let us consider independent, identically distributed samples $X_i^1, \ldots, X_i^N$ with exactly $m - i$ elements up, and take the mean sample:

$$\widehat{\phi}_i = \frac{1}{N} \sum_{j=i}^{N} \phi(X_i^j). \tag{7}$$

There are $\binom{m}{i}$ states with $m - i$ elements up. Therefore, $\widehat{F}_i = \binom{m}{i} \widehat{\phi}_i$ is an unbiased estimation for F_i. Finally, an unbiased estimation for the reliability polynomial $r(p)$

is:

$$r_{FMC}(p) = \sum_{i=0}^{m} \hat{F}_i p^{m-i}(1-p)^i.$$

This method has been discussed in Canale et al. (2014) applied to the network reliability context. An obvious limitation is that the method is not applicable when the SMBS is not homogeneous. The pseudocode of the method is shown in Figure 4. The algorithm FMC receives the rule ϕ, the ground-set of components S, and returns a polynomial r_{FMC} in the variable $p \in [0,1]$. In the block of Lines 1-8, a pointwise estimation of the coefficients F_i takes place, and we get $\widehat{F_i}$. Observe that $\widehat{F_i}$ is precisely the proportion of operational states with i bits sets to 0. Therefore, an averaging is considered in order to find a pointwise estimation of that proportion with a sample of size N in a for-loop (Lines 3-6). In Line 4, we choose $m - i$ elements from S using Function $Choose$, and the result is a binary word with those bits set to 1 in the respective order. The number Sum_i is updated in Line 5, and the averaging takes place in Line 7. In Line 9, the reliability polynomial $r_{FMC}(p)$ is found using Expression (7), and it is returned in Line 10.

4.4 Permutational and Splitting–like methods

All the SBS addressed so far are essentially static. However, in order to pursue a variance reduction in the reliability estimation it is possible to transform them into dynamic systems modeled by stochastic processes that, under certain conditions, have the same reliability as the original static SBS. Such transformation gives access to a wide scope of methods, like Permutation Monte Carlo and Splitting, originally conceived to operate over dynamic models. An extensive review of the ideas that gave rise to these methods can be found in Elperin et al. (1991), Glasserman et al. (1996), Garvels (2000), L'Ecuyer et al. (2007), L'Ecuyer et al. (2009), Amrein and Künsch (2011) and Gertsbakh et al. (2014a).

The methods to be introduced now are based on an artificial time, along which the components are *repaired*, provided that they are failed at time $t = 0$. This is the guiding idea of many important research lines, some of which have been also adapted to the case of multiple states per component, like the work in Gertsbakh et al. (2014a). An original and efficient approach, one of whose applications can be found in Gertsbakh et al. (2016), is based on a distribution called D–spectrum, proposed to make a particular description of the sequences of *repairs*, for the case of homogeneous systems. For a complete survey of all of these methods, readers are advised to look over Gertsbakh and Shpungin (2009) and Gertsbakh et al. (2014b).

The transformation of an SBS into a dynamic system, and the analysis of the sequences of *repairs* (permutations), are the basis of three methods discussed in the following subsections. The first two methods have been used in network reliability models, even if in this paper we discuss them in the SBS context; but the third one is a novel proposal in this article, which has not been previously proposed or applied in any context.

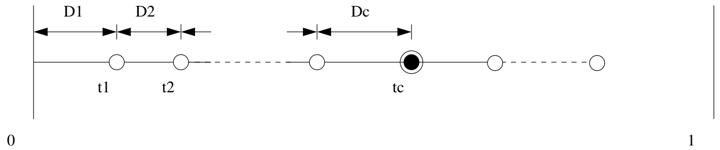

Figure 5: The Creation Process (CP)

4.4.1 Permutation Monte Carlo (PMC)

In the Creation Process (CP), proposed by Elperin et al. (1991), the SBS is no longer modeled by the random vector X, instead, it is modeled by the independent components of the random process $X(t) = (X_1(t), \dots, X_m(t))$, in which:

$$X_i(t) = \begin{cases} 0 & \text{if } t < \tau_i \\ 1 & \text{otherwise,} \end{cases} \qquad i = 1, \dots, m. \tag{8}$$

In words, at $t = 0$ all the components are failed, or simply do not exist, while at times τ_i they are repaired or created. Every τ_i is exponentially distributed with rate $\lambda_i = -\ln(q_i)$ so that, to observe CP at $t = 1$ is the same as observing the static SBS, because $P(X_i(1) = 1) = r_i$ and $P(X_i(1) = 0) = q_i$.

See that, if at time $t = 1$ each component exists with probability $p_i,\ i = 1, \dots, m$, then, at $t = 1$ the SBS is *up* (resp. *down*) with a probability equal to the system reliability (resp. unreliability).

Figure 5 shows a possible sequence of repairs, each one of which occurs at time $\tau_{(i)},\ i = 1, \dots, c$. It is important to distinguish the reference to the order in the sequence (the index in brackets) and the number of component, because time $\tau_{(i)}$ is the repair time of the j–th component, but not necessarily $i = j$. The order of repairs, in terms of the numbers of component, is a permutation of the ground set S.

Time $\tau_{(c)}$ is the repair time of some component —called critical component— such that, from there on the system is *up*. As a consequence:

$$\phi(X(t)) = \begin{cases} 0 & \text{if } t < \tau_{(c)} \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

Thus, the event $\phi(X) = 0$ in the static SBS is the same as event $\tau_{(c)} \geq 1$ in CP, what, in the end, means that $q = P(\tau_{(c)} \geq 1)$. Then, given the following indicator variable:

$$I_P = \begin{cases} 1 & \text{if } \tau_{(c)} \geq 1 \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

the crude or standard unreliability estimation, $\widehat{q}_{CMC}$, is:

$$\widehat{q}_{CMC} = \frac{1}{N} \sum_{i=1}^{N} I_P^{(i)}, \tag{11}$$

where $I_P^{(i)},\ i = 1, \ldots, N$, is a set of N independent copies of variable I_P.

In order to improve the estimation in (11), it is important to find a mathematical description of the time $\tau_{(c)}$. Considering the times between repairs, as shown in Figure 5, time $\tau_{(c)}$ is composed as follows:

$$\tau_{(c)} = \sum_{i=1}^{c} \Delta_i \tag{12}$$

The time Δ_1 is the smallest (earliest) out of a set of m exponentially distributed times, each one of them with rate $\lambda_i,\ i = 1, \ldots, m$. So, the rate of Δ_1 is $\Lambda_1 = \sum_{i=1}^{c} \lambda_i$, whereas the component repaired at time $\tau_{(1)}$ can be sampled from a discrete distribution in which the probability of each component is $\lambda_i/\Lambda_1,\ i = 1, \ldots, m.$ Assuming that component x is sampled at time $\tau_{(1)}$, Δ_2 is exponentially distributed with rate $\Lambda_2 = \Lambda_1 - \lambda_x$, and the component repaired at time $\tau_{(2)}$ follows a discrete distribution in which the probability of each component is $\lambda_i/\Lambda_2\ \ i = 1, \ldots, m,\ i \neq x.$ The mechanism can be extended to all the components in S, however only the first c ones are of interest. Let us call $NextComp(X)$ to a function that, given the value of X at certain moment, returns the number of the next component to be repaired, according to the mechanism just introduced (it is used in the pseudocode of Figure 6).

The random variables $\Delta_1, \ldots, \Delta_c$, and, therefore the variable $\tau_{(c)} = \sum_{i=1}^{c} \Delta_i$, are determined by the permutation $\omega \in \Omega$, where Ω is the set (space) of all the possible permutations. Thus, according to the Total Probability Theorem, the unreliability can be expressed as follows:

$$q = P(\tau_{(c)} \geq 1) = \sum_{\omega} \underbrace{P(\tau_{(c)} \geq 1 \mid \Omega = \omega)}_{\gamma(\omega)} P(\Omega = \omega) \tag{13}$$

This expression is the basis of a Conditional Monte Carlo application known as Permutation Monte Carlo (PMC). The term $\gamma(\omega)$ is the probability that the SBS is up at $t = 1$ when the components are repaired according to the permutation ω. Thus, if for any sampled permutation, the probability $\gamma(\omega)$ can be computed, the following Monte Carlo estimation is possible:

$$\widehat{q}_P = \frac{1}{N} \sum_{i=1}^{N} \gamma(\omega^{(i)}), \tag{14}$$

where $\omega^{(i)},\ i = 1, \ldots, N$, is a set of N independent replications of the permutation ω. It is important to notice that, in order to compute $\gamma(\omega^{(i)})$, it is not necessary to sample the times $\Delta_1, \ldots, \Delta_c$, for every replication, but only the order of repairs (permutation). Once a permutation, ω, is sampled, $\gamma(\omega)$ can be computed according to the distribution determined by Balázs (2005), that is:

$$\gamma(\omega) = P(\Delta_1 + \ldots + \Delta_c > x \mid \omega) = \left(\prod_{i=1}^{c} \Lambda_i \right) \sum_{j=1}^{c} \frac{e^{-\Lambda_j\, x}}{\Lambda_j \prod\limits_{\substack{k=1 \\ k \neq j}}^{c} (\Lambda_k - \Lambda_j)} \tag{15}$$

Algorithm 6 $q_P = PMC(\phi, S, p)$

```
$Sum \leftarrow 0$
for $i = 1$ to $N$ do
    $X_i \leftarrow \vec{0}_m$
    $\omega_i \leftarrow \emptyset$
    while $\phi(X_i) = 0$ do
        $j \leftarrow NextComp(X_i)$
        $X_i \leftarrow X_i + e_j$
        $\omega_i \leftarrow \omega_i \cup j$
    end while
    $Sum \leftarrow Sum + \gamma(\omega_i)$
end for
return $q_P = Sum/N$
```

Figure 6: Pseudocode for Permutation Monte Carlo method.

A pseudocode for the Permutation Monte Carlo algorithm is presented in Figure 6.

It is simple to show that the variance of the estimator $\widehat{q}_P$ in (14) is smaller than the variance of the standard estimator $\widehat{q}_{CMC}$ in (11):

$$
\begin{aligned}
V(\widehat{q}_{CMC}) = q\,(1-q) & \\
&= \sum_{\omega} \gamma(\omega)\, P(\Omega = \omega) \left(1 - \sum_{\omega} \gamma(\omega)\, P(\Omega = \omega) \right) \\
&= \sum_{\omega} \gamma(\omega)\, P(\Omega = \omega) - \left(\sum_{\omega} \gamma(\omega)\, P(\Omega = \omega) \right)^2 \pm \sum_{\omega} \gamma(\omega)^2\, P(\Omega = \omega) \\
&= V(\widehat{q}_P) + \underbrace{E(\gamma(\omega)) - E(\gamma(\omega)^2)}_{\geq 0}
\end{aligned}
$$

what shows that $V(\widehat{q}_P) \leq V(\widehat{q}_{CMC})$.

4.4.2 Splitting/CP

Another CP–based method for estimating the unreliability of an SBS is Splitting/CP. In this method, that has been proposed by Murray et al. (2013), the sequences of repairs are seen as trajectories that start at $t = 0$ and progress towards $t = 1$. The target value is still the probability of the event $\tau_{(c)} \geq 1$, which, in the case of highly reliable systems, is a rare event. Splitting/CP attempts to stimulate the occurrence of this event by artificially increasing the number of trajectories, cloning or splitting them at certain points determined by the cross of intermediate thresholds (one or more) placed between $t = 0$ and $t = 1$. Figure 7 shows a small example with only one intermediate threshold placed at $t = 0.5$. We will refer to it as h_1, and we will call, respectively, h_0 and

h_2 to the bounds at the extremes (even when they do not play the same role as the intermediate threshold, h_1).

The Splitting mechanism posits that: *every time a threshold is crossed, a set of new trajectories is launched from the threshold crossed, all of them preserving the state of the incident trajectory at the crossing point.* In the example shown in Figure 7, every time h_1 is crossed, two new trajectories are launched from the crossing point. This splitting factor (two, in this case) is a parameter that has to be selected in every Splitting implementation.

The successful trajectories are those for which the critical component appears beyond $t = 1$, therefore, the candidates are those which do not contain the critical component, when a threshold is crossed. On the other hand whenever the critical component is sampled on some trajectory, it is immediately stopped and no more components need to be sampled on it. Such is the case of trajectory T_1 in Figure 7.

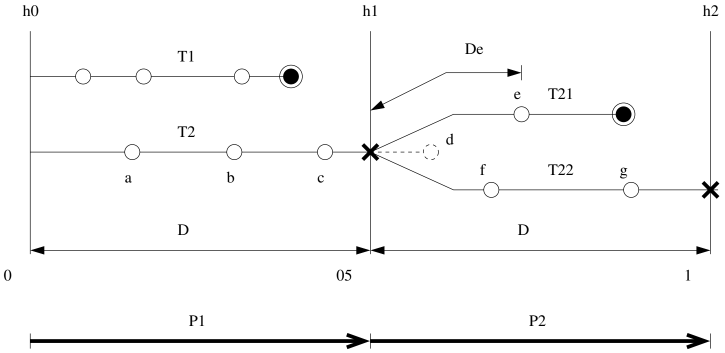

Figure 7: Splitting/CP, small example.

In trajectory T_2, the critical component does not appear earlier than h_1, that is why it is cloned as shown, creating trajectories T_{21} and T_{21}. The state of T_2 at the crossing point is as follows: components a, b and c are already repaired, while the times at which the remaining components will be repaired are all “running”. This is the initial state of both, T_{21} and T_{21}, at $t = 0.5$.

The first component repaired on T_{21} is e, which is not forced to be d again, although it could be. Time Δ_e, should be counted from the repair time of component c, conditioned on the fact that component e must be sampled beyond h_1 (remember that, at $t = 0.5$, the repair times of all the components not sampled yet, are all “running”). However, given the lack of memory of the exponential distribution, Δ_e can be counted from h_1. Trajectory T_{21} is discarded immediately after, because the critical component is sampled earlier than the next threshold, h_2.

Trajectory T_{22} is launched at the same point and with the same initial state as T_{21}. The only difference is that trajectory T_{22} reaches the final threshold, h_2, because the critical component is not shown on it (it appears beyond $t = 1$).

According to the Splitting/CP method, in a setting with N_T intermediate thresholds, the system unreliability reads as follows:

$$q = \prod_{i=1}^{N_T+1} q_{S_i}, \tag{16}$$

where $q_{S_i},\ i = 1, \ldots, N_T + 1$, is the probability that a trajectory started at threshold $i - 1$ reaches threshold i. The final unbiased Splitting/CP estimation, $\widehat{q}_S$, results from replacing each term in (16) with an appropriate estimator, that is:

$$\widehat{q}_S = \prod_{i=1}^{N_T+1} \widehat{q}_{S_i}. \tag{17}$$

Each $\widehat{q}_{S_i},\ i = 1, \ldots, N_T + 1$, is a crude or standard estimator that can be obtained simply as the ratio between the number of trajectories that reach threshold h_i and the total number of trajectories started from threshold h_{i-1} . In the problem shown in Figure 7, where the unreliability estimation is: $\widehat{q}_S = \widehat{q}_{S_1} \times \widehat{q}_{S_2}$, two trajectories are launched from h_0, one of which reach h_1, then $\widehat{q}_{S_1} = 1/2$. Similarly, two new trajectories are launched from h_1, but only one of them reaches h_2, therefore, $\widehat{q}_{S_2} = 1/2$, and, $\widehat{q}_S = 1/4$.

Let us go back to the general setting with N_T intermediate thresholds. Call α_0 to the first Splitting factor, that is, the number of trajectories launched from h_0, and α_i to the number of new trajectories started at threshold $h_i,\ i = 1, \ldots, N_T$, every time h_i is crossed by a candidate trajectory. Let K_i be the total number of trajectories stared at threshold $h_i,\ i = 0, \ldots, N_T$, and R_i the total number of trajectories that reach threshold $h_i,\ i = 1, \ldots, N_T + 1$. The general form of the Splitting/CP estimator is:

$$\begin{aligned} \widehat{q}_S &= \widehat{q}_{S_1}\, \widehat{q}_{S_2} \cdots \widehat{q}_{S_{N_T+1}} \\ &= \frac{R_1}{K_0}\, \frac{R_2}{K_1} \cdots \frac{R_{N_T+1}}{K_{N_T}} \\ &= \frac{R_1}{(\alpha_0)}\, \frac{R_2}{(R_1\alpha_1)} \cdots \frac{R_{N_T+1}}{(R_{N_T}\alpha_{N_T})} \\ &= \frac{1}{\alpha_0\alpha_1 \cdots \alpha_{N_T}} R_{N_T+1} \end{aligned} \tag{18}$$

Figure 8 shows a description of the method of Splitting/CP in the form of pseudocode. There, the vector $\alpha = (\alpha_0, \ldots, \alpha_{N_T})$, is the set of Splitting factors.

Think now that the α_0 independent trajectories launched from h_0 are modeled by α_0 Bernoulli random variables, $\mathtt{1}_{i_0}$, with parameter p_{i_0}, $i_0 = 1, 2, \cdots, \alpha_0$. These variables equal 1 if the corresponding trajectory reaches threshold h_1, and 0 otherwise. Thus, the number R_1 of trajectories crossing threshold h_1 is the number of these Bernoulli variables assuming a value of 1, that is:

$$R_1 \;=\; \sum_{i_0=1}^{\alpha_0} \mathtt{1}_{i_0} \tag{19}$$

When a trajectory modeled by the variable $\mathtt{1}_{i_0}$ crosses threshold h_1, α_1 new trajectories are started right at the crossing point. These new α_1 trajectories are

```
Algorithm 7 $q_{SPL} = SPL(\phi, S, p, \alpha, N_T)$
$Sum \leftarrow 0$
for $i = 1$ to $N$ do
    $Split(\phi, S, p, \alpha, N_T, 0)$
    $Sum \leftarrow Sum + R_{N_T+1}/(\alpha_0\,\alpha_1\,\ldots\,\alpha_{N_T})$
end for
return $q_{SPL} = Sum/N$
function $Split(\phi, S, p, N_T, \alpha, N_T, z)$
for $i = 1$ to $\alpha_z$ do
    Sample repair times, $\tau$, until $\tau > h_{z+1}$, and update $X$
    if $\phi(X) = 0$ then
        $R_{z+1} \leftarrow R_{z+1} + 1$, and cancel the repair beyond $h_{z+1}$
        if $z \leq N_T$ then
            $Split(\phi, S, p, \alpha, N_T, z + 1)$
        end if
    end if
end for
return
```

Figure 8: Pseudocode for Splitting/CP.

modeled by α_1 Bernoulli random variables $\mathtt{1}_{i_0 i_1}$ with parameter $p_{i_0 i_1}$, $i_1 = 1, 2, \cdots, \alpha_1$. Consequently:

$$R_2 \;=\; \sum_{i_0=1}^{\alpha_0} \mathtt{1}_{i_0} \left(\sum_{i_1=1}^{\alpha_1} \mathtt{1}_{i_0 i_1} \right) \;=\; \sum_{i_0=1}^{\alpha_0} \sum_{i_1=1}^{\alpha_1} \mathtt{1}_{i_0} \mathtt{1}_{i_0 i_1} \tag{20}$$

Finally, replacing the value of R_{N_T+1} in (18), the Splitting/CP estimator can be expressed as:

$$\widehat{q}_S \;=\; \frac{1}{\alpha_0 \alpha_1 \cdots \alpha_{N_T}} \sum_{i_0=1}^{\alpha_0} \sum_{i_1=1}^{\alpha_1} \cdots \sum_{i_{N_T}=1}^{\alpha_{N_T}} \mathtt{1}_{i_0} \mathtt{1}_{i_0 i_1} \cdots \mathtt{1}_{i_0 i_1 \cdots i_{N_T}} \tag{21}$$

The Bernoulli random variables are responsible for building the trajectories tree and also for the counting of trajectories (*starting from* and *reaching* thresholds), used to compute the standard Monte Carlo estimates per stage.

4.4.3 Permutation Monte Carlo over Splitting/CP (PMC/SPL)

Splitting is the basis of a multiplicity of methods adapted, each one of them, to a specific context. Such is the case of Botev and Kroese (2012), Botev et al. (2018) and Botev et al. (2016) with Generalized Splitting, Villén-Altamirano and

Villén-Altamirano (1991) and Villén-Altamirano and Villén-Altamirano (2002) where the variant RESTART is approached, and Cancela et al. (2019) where a Splitting–based method, to be used on multi–valued (rather than binary) systems, was developed. Now, a new Splitting variant that somehow combines the algorithms introduced in Sections 4.4.1 and 4.4.2, will be introduced.

Even when there is not a close expression for the variance of the estimator $\widehat{q}_S$ in (17), it is easy to accept that the more precise the estimators $\widehat{q}_{S_i},\ i = 1, \ldots, N_T + 1$, the more precise the estimator $\widehat{q}_S$. We propose to replace the method for obtaining the estimators $\widehat{q}_{S_i}$, by another with lower variance. Specifically, we propose to use Permutation Monte Carlo instead of standard Monte Carlo. This variant will be introduced here, with the aid of Figure 9, which shows in more detail the same example as the one shown in Figure 7.

In $\langle\, 1\, \rangle$ trajectory T_1 is launched. This trajectory does not reach h_1. As part of the standard Splitting mechanism, T_1 would deserve no further attention; there is no crossing point on it, neither new trajectories launched from h_1. Thus, at the end of a standard Splitting application, T_1 would contribute with a 0 to the estimation process of q_{S_1}.

See that the critical component is repaired at a time composed of the sum of four times, namely Δ_1, Δ_2, Δ_3 and Δ_4, which, in turn, are determined by an underlying permutation that we call ω_1. Thus, whenever ω_1 is sampled, the critical component appears beyond h_1 with probability p_1, and earlier than h_1 with probability $1 - p_1$, then:

$$p_1 = P(\Delta_1 + \Delta_2 + \Delta_3 + \Delta_4 > D \mid \Omega = \omega_1). \tag{22}$$

In trajectory T_2, shown in $\langle\, 2\, \rangle$, two repair times were sampled even after threshold h_1 was crossed. This is because we need the critical component to be sampled in every trajectory (we will see the reason of this immediately). In a standard Splitting application, T_2 would contribute with a 1 to the estimation process of q_{S_1}. Now, making a similar analysis as the one made for trajectory T_1, we can say that, if the permutation underlying T_2 is sampled again, let us call it ω_2, the critical component will be beyond h_1 with probability p_2, and earlier than h_1 with probability $1 - p_2$, then:

$$p_2 = P(\Delta_1 + \Delta_2 + \Delta_3 + \Delta_4 + \Delta_5 > D \mid \Omega = \omega_2). \tag{23}$$

Clearly, Δ_1, Δ_2, Δ_3 and Δ_4 in $\langle\, 2\, \rangle$, are not necessarily the same random variables as those in $\langle\, 1\, \rangle$, just because ω_1 and ω_2 are not necessarily equal. After the computation of p_2, the repair times sampled beyond h_1 are deleted from T_2 and considered as if they had never been sampled. They only serve to the purpose of computing of p_2 in (23).

Consider now all the possible permutations $\omega \in \Omega$, and assume that all the associated trajectories include the critical component. Then, the exact probability that a trajectory started at $t = 0$ reaches threshold h_1 is q_{PS_1}:

$$q_{PS_1} = P(\Delta_1 + \ldots + \Delta_c > D) = \sum_{\omega} \underbrace{P(\Delta_1 + \ldots + \Delta_c) > D \mid \Omega = \omega)}_{\gamma_1(\omega)} P(\Omega = \omega) \tag{24}$$

Let us suppose that in the general case, α_0 independent trajectories, $T_i,\ i = 1, \ldots, \alpha_0$, starting at at $t = 0$, are sampled. For each one of them there will be a sequence of times, $\Delta_1, \ldots, \Delta_c$, and an underlying permutation $\omega^{(i)} \in \Omega$. Then, once the probabilities $\gamma_1(\omega^{(i)})$ are computed (making use of expression (15)), the following estimator, which is more accurate than $\widehat{q}_{S_1}$ in (17), can be obtained:

$$\widehat{q}_{PS_1} = \frac{1}{\alpha_0} \sum_{i=1}^{\alpha_0} \gamma_1(\omega^{(i)}). \tag{25}$$

See that the terms $\gamma_1(\omega^{(i)})$ in this sum are the probabilities $p_i,\ i = 1, \ldots, \alpha_0$, the first two of which (pointed out in Figure 9), are p_1 in (22) and p_2 in (23).

In $\langle\, 3 \,\rangle$ and $\langle\, 4 \,\rangle$ we can see trajectories T_{21} and T_{22}, upon which $p_{21} = P(\Delta_4 + \Delta_5 > D)$ and $p_{22} = P(\Delta_4 + \Delta_5 + \Delta_6 > D)$ can be computed according to the same guidelines introduced for $\langle\, 1 \,\rangle$ and $\langle\, 2 \,\rangle$. The only thing to keep in mind is that, in the computations of both, p_{21} and p_{22}, there are three components that have already been sampled in $\langle\, 2 \,\rangle$.

With all this in mind, the example in Figure 9 is solved as follows. Once p_1 and p_2 have been computed by means of Permutation Monte Carlo, the probability that a trajectory started at h_0 reaches threshold h_1, is estimated as:

$$\widehat{q}_{PS_1} = \frac{p_1 + p_2}{2}. \tag{26}$$

Similarly, after the probabilities p_{21} and p_{22} are computed, the probability that a trajectory started at h_1 reaches threshold h_2, is estimated as:

$$\widehat{q}_{PS_2} = \frac{p_{21} + p_{22}}{2}. \tag{27}$$

Finally, the unreliability estimation of the corresponding SBS is:

$$\widehat{q}_{PS} = \widehat{q}_{PS_1} \times \widehat{q}_{PS_2}. \tag{28}$$

In Splitting/CP the trajectories play a double role, they draw the trajectories tree that grows from $t = 0$ towards $t = 1$ and they set the crossing points, whose count is enough to make a standard estimation per stage. Now, in this application of Permutation Monte Carlo over Splitting/CP (PMC/SPL), the trajectories are still responsible for drawing the tree, but there is not counting process of the crossing points. Instead, there is a computation of a probability for every new trajectory launched.

The method of Permutation Monte Carlo over Splitting/CP is described in the form of pseudocode in Figure 10.

Think again that there are α_0 independent trajectories launched from h_0, each one modeled by a Bernoulli random variable 1_{i_0} with parameter p_{i_0}, $i_0 = 1, 2, \cdots, \alpha_0$. The average of these probabilities (all of them, whether they equal 1 or 0) is the Permutation Monte Carlo estimator for the first stage, that is:

$$\widehat{q}_{PS_1} \; = \; \frac{\displaystyle\sum_{i_0=1}^{\alpha_0} p_{i_0}}{\alpha_0} \tag{29}$$

These α_0 random variables equal 1 if the corresponding trajectory reaches threshold h_1, and 0 otherwise. In terms of these variables, $R_1 = \sum_{i_0=1}^{\alpha_0} 1_{i_0}$ and, therefore, the total number of trajectories started from h_1 is $\alpha_1 \sum_{i_0=1}^{\alpha_0} 1_{i_0}$. The sum of the probabilities associated to all these trajectories is $\sum_{i_0=1}^{\alpha_0} 1_{i_0} \left(\sum_{i_1=1}^{\alpha_1} p_{i_0 i_1}\right)$. Finally, the Permutation Monte Carlo estimator for the second stage is the average of these probabilities:

$$\widehat{q}_{PS_2} = \frac{\displaystyle\sum_{i_0=1}^{\alpha_0} \sum_{i_1=1}^{\alpha_1} 1_{i_0} p_{i_0 i_1}}{\displaystyle\alpha_1 \sum_{i_0=1}^{\alpha_0} 1_{i_0}} \tag{30}$$

A similar analysis for the j–th stage, $j = 2, \ldots, N_T + 1$, produces the following estimator:

$$\widehat{q}_{PS_j} = \frac{\displaystyle\sum_{i_0=1}^{\alpha_0} \sum_{i_1=1}^{\alpha_1} \cdots \sum_{i_j=1}^{\alpha_{j-1}} 1_{i_0} 1_{i_0 i_1} \cdots p_{i_0 \ldots i_{j-1}}}{\displaystyle\alpha_{j-1} \sum_{i_0=1}^{\alpha_0} \sum_{i_1=1}^{\alpha_1} \cdots \sum_{i_{j-2}=1}^{\alpha_{j-2}} 1_{i_0} 1_{i_0 i_1} \cdots 1_{i_0 i_1 \cdots i_{j-2}}} \tag{31}$$

Finally, an unbiased unreliability estimator for the SBS is: $\widehat{q}_{PS} = \widehat{q}_{PS_1} \widehat{q}_{PS_2} \cdots \widehat{q}_{PS_{N_T+1}}$.

5 Experimental Analysis

In this section we compare the effectiveness of the different Monte Carlo algorithms discussed in the previous section.

We carried out experiments over eight main network test configurations (covering twenty four systems when considering different elementary reliabilities).

Tests 1 and 2 are based on the K-Terminal reliability problem applied to the Arpanet network, depicted in Figure 11, where the black nodes build the set K.

Tests 3 and 4 are based on the All-Terminal reliability problem applied to the network depicted in Figure 12; it corresponds to the RAU network (Red Académica Universitaria/Universitary Academic Network), the Internet network linking all universities and research centers in Uruguay. All nodes in the RAU network are black, since all of them are considered as terminals.

Tests 5 and 7 are based on the K-Terminal reliability problem applied to the Easton–Wong Network (EW), a topology taken from Easton and Wong (1980) and shown in Figure 13, where the five external black nodes are taken as terminals.

Tests 6 and 8 are based on the Source-Terminal reliability problem applied to the "ship13" problem (Reduced Graph), a topology found in pag. 189 of the book by Stoer (1992). This network, called here Ship, is composed of 159 nodes and 328 links. Two nodes that are sufficiently far from each other, were selected to play the role of s and t (there is a shortest path of 10 links between them).

Tests 1, 3, 5 and 7 consider classical reliability cases, whereas Tests 2, 4, 6 and 8 correspond to diameter constrained cases. They will be denoted, respectively, as CLR and DCR cases.

Finally, Test 9 is an instance of the k-out-of-m problem.

The details of the nine test cases are:

1. K-Terminal CLR, on the Arpanet with $p_e = 0.9, 0.99$ and 0.999 (three different homogeneous systems).

2. K-Terminal DCR, on the Arpanet, $d = 7$, with $p_e = 0.9, 0.99$ and 0.999 (three different homogeneous systems).

3. All-Terminal CLR, on the RAU with $p_e = 0.9, 0.99$ and 0.999 (three different homogeneous systems).

4. All-Terminal DCR, on the RAU, $d = 4$, with $p_e = 0.9, 0.99$ and 0.999 (three different homogeneous systems).

5. K-Terminal CLR, on the EW with $p_e = 0.9, 0.99$ and 0.999 (three different homogeneous systems).

6. Source-Terminal CLR, on the Ship with $p_e = 0.9, 0.99$ and 0.999 (three different homogeneous systems).

7. K-Terminal DCR, on the EW with $p_e = 0.9, 0.99$ and 0.999 (three different homogeneous systems).

8. Source-Terminal DCR, on the Ship with $p_e = 0.9, 0.99$ and 0.999 (three different homogeneous systems).

9. 3-out-of-29 with $p_i = 0.1 + 0.8 \times (i-1)/28, i = 1..29$ (heterogeneous system).

The algorithms were coded in C and C++. The tests were run on an x64 Intel Core i7 2.6 GHz computer with 32 GB RAM. Source codes are available at http://www2.um.edu.uy/ieem-papers/sbs.zip.

We apply CMC, FMC, RVR, PMC, Splitting/CP and PMC/SPL to the selected examples of SMBS. Since FMC estimates the reliability polynomial, it is only suitable for homogeneous scenarios (i.e., with identical probabilities of operation p in all components). Finding this polynomial is trivial for k-out-of-m systems; $F_i = 0$ for all $i > m - k$ and $F_i = \binom{m}{i}$ for all $i \leq k$.

Observe that CMC, FMC, RVR, PMC, Splitting/CP and PMC/SPL return unbiased reliability estimators. In Monte Carlo methods, increasing the sample size improves the quality of the estimators at the expense of more computing time. Therefore, a fair comparison should consider both, the precision of the estimation (measured by the variance) and the computational effort. Suppose we have two methods A and B that provide unbiased estimations of the reliability. The relative efficiency of A with respect to B is the ratio $t_B V(B)/(t_A V(A))$, where t_A and t_B are the respective mean computational times (in the same units), and $V(A)$ and $V(B)$ the variances (or mean square errors) of the estimators of the respective methods A and B; see Fishman (1996).

The six algorithms are run for the first eight tests. However, as said before, FMC is not defined for heterogeneous systems, hence the ninth test compares CMC versus RVR, PMC, Splitting/CP and PMC/SPL.

Let us look more in depth at the ninth test. The number of components under operation in a k-out-of-m system is a sum of independent Bernoulli variables with different success probabilities. This is a Poisson-Binomial distribution, and a closed formula is provided in Fernandez and Williams (2010). If $p = (p_1, \ldots, p_m)$ denotes the probability of operation for each component and $X_i \sim Ber(p_i)$ represents the random operation for component i, then $p(h) = P(\sum_{i=1}^{m} X_i = h)$ is the probability mass function of the number of components under operation. Consider the cumulative probability of the tail, $Q(x) = \sum_{j=x}^{m} p(j)$. The exact reliability of a k-out-of-m system, according to Fernandez and Williams (2010), is precisely:

$$r = Q(k) = 1 - \frac{k}{m+1} - \frac{1}{m+1} \sum_{i=1}^{m} \frac{1 - \mathrm{e}^{-j2\pi ik/(m+1)}}{1 - \mathrm{e}^{-j2\pi i/(m+1)}} \times \prod_{h=1}^{m} \Big[p_h \mathrm{e}^{j2\pi i/(m+1)} + 1 - p_h \Big],$$

with $j^2 = -1$.

In Monte Carlo simulation it is impossible to select a unique sample size to make a fair comparison among different methods if such comparison is based on the computational time or the variance (one or the other). However, since the relative efficiency becomes a constant after a significant number of replications, the only requirement for selecting a proper sample size in a comparative setting, is that it is large enough to let the product between variance and simulation time become a constant (in the sense that it does not change if the sample size is increased).

In Tests 1 to 8, the average of 100 independent runs of both CMC and RVR is computed, using sample sizes $N = 10^6$ (for CMC) and $N = 10^4$ (for RVR), as well as the average of 30 independent runs of FMC (using $N = 10^5$ as the CMC sample size for estimating each F_i). In Test 9, both CMC and RVR, run with the same sample size $N = 10^6$. In all the tests the sample size for PMC is $N = 10^6$.

In every single replication of a Splitting–like method, a number H of trajectories is launched at $t = 0$. Accepting that a number of runs N is performed, each one of them based on H trajectories initiated, a significant parameter to measure the effort of the whole simulation is, therefore, $N \times H$ (which is the total number of trajectories actually launched). In all the tests $N \times H = 10^5$, except for the tests on the Ship network in which $N \times H = 10^4$. The number of intermediate thresholds, N_T, was selected, after a set of pilot runs in every experiment, attempting to reach the least value of the product between variance and simulation time. The values of N_T used ranged between 1 and 26.

The results are presented in Tables 1, 2, 3 and 4. Average unreliabilities, computing times, sampling variances and the efficiencies relative to CMC, are shown in different rows.

The first observation is that for relatively low values of p, namely 0.9, the basic CMC method is one of the best options. Both RVR and FMC have high computational times in relation to the CMC algorithm; and for $p = 0.9$, the variances do not vary as much, leading to an overall lower relative efficiency for these two methods. In many of the tests for $p = 0.9$, Splitting/CP and PMC/SPL are in the order of CMC (slightly better), because the low reliability settings require an extremely low number of intermediate thresholds, say 1 or 2, what makes these implementations quite similar to CMC, with the additional benefit of cloning trajectories. The results for PMC are also slightly better than CMC for $p = 0.9$. The fact is that PMC has bounded relative error and is known to be an extremely efficient method for small networks. One of the limitations of PMC over large networks is that the structure function has to be evaluated each time a link is repaired (the larger the network, the higher the number of links to be repaired). Another limitation of PMC is related to the formula shown in (15), which has two products that grow exponentially on the number of links. Thus, depending on the data type selected for the variables involved, some overflow problems may occur. We have a tool to compute the distribution shown in (15) as a convolution, by means of a recursive formula, but it is extremely low and consumes high amounts of memory. Some alternatives that we consider to alleviate this problem are subject of future work.

On the other hand, when p grows, the execution times do not vary much; but the variance reductions attained are considerable, and the relative efficiency of all the methods with respect to CMC grows quickly, attaining considerably large values. The general observation is that, as expected, efficiency grows with p, the highest values being attained at $p = 0.999$, in accordance with previous results from the literature; see Cancela and El Khadiri (1995); Rubino and Tuffin (2009).

A general comparison among all methods is not so clear. Only slight and varied trends can be observed, except for the case of the largest networks (EW and Ship, CLR) for which RVR is, by far, the most efficient method. In the case of the Arpanet, K–Terminal reliability, PMC and the Splitting–like methods appear as the most efficient for CLR, and RVR for DCR, whereas for the RAU network All-Terminal reliability, PMC, Slitting/CP and PMC/SPL are the most efficient in both, CLR and DCR.

It should be noted that PMC/SPL, that was proposed in this article as an improvement of Splitting/CP, outperforms Splitting/CP in almost all tests.

Finally, in Test 9, that corresponds to a non–homogeneous case, the best performance corresponds to PMC. An accurate PMC estimation should be expected on this model, because, as said earlier, the computational effort of PMC is determined by the number of components —that in this case is quite low— and the cost of the structure function, which in this case is extremely light, as it is only a comparison to find out whether the number of operational components is less than 3.

6 Conclusions

Several real-world systems can be mathematically modeled by stochastic binary systems (SBS). Finding cutsets, essential and irrelevant components are hard computational problems in arbitrary SBS. In the particular case of stochastic monotone binary systems (SMBS), the task of finding cutsets (or pathsets) can be produced be means of m rule-evaluations, being m the number of components involved in the system. This fact serves to generalize the Recursive Variance

Reduction (RVR) method for the reliability evaluation of SMBS. In the case of homogeneous SMBS, the FMC method can also be applied.

The computational experiments show that for low reliabilities, Splitting/CP and PMC/SPL are the better methods. Nevertheless, the standard CMC method is also one of the most efficient, as it has low computational cost and a precision that is similar to the more sophisticated methods. Splitting/CP and PMC/SPL performance is slightly better than that of CMC, because for low reliabilities they only require 1 or 2 intermediate thresholds, what makes their structures quite similar to CMC, but they still have the benefit of cloning trajectories.

For high reliability values, all the methods implemented are much more efficient than CMC; offsetting the high computational costs per iteration by achieving higher precision, but there is no clear trend in favor of any of the methods, except for the case of larger networks where RVR is, by far, the most efficient method. For the other network models, the trends are diverse and there is no clear winner.

While it is possible to formally prove that RVR and PMC achieve variance reductions over CMC, no such proof is still available for the other implementations. This might be a matter for future work.

Depending on the context, a possibly important limitation of FMC is that it is not applicable for non–homogeneous systems.

Work in progress includes finding reliability bounds and studying efficient representations for general SBSs. Other future work should cover the study of alternative ways to find mincuts that efficiently exploit the structure of the SMBS, and their impact on the performance of the reliability estimation. Also a more efficient mechanism for computing the distribution shown in (15) is subject of future work.

An additional observation is that while this work is concentrated on static models, their reliability can also be seen as the steady-state behavior of a Markovian or semi-Markovian system (see for instance Perman et al. (1997), Carazas et al. (2011) and Wu et al. (2020)) where the states of the individual components are not static, but evolve in time following some probability distributions. Future work could study in detail the relationship of such dynamic models with the static ones, and see if other Monte Carlo methods could lead to improved estimations of the measures of interest.

Acknowledgment

This work was partially supported by Project ANII FCE_1_2019_1_156693 *Teoría y Construcción de Redes de Máxima Confiabilidad*, MATHAMSUD 19–MATH-03 *RAREDEP - Rare events analysis in multi-component systems with dependent components* and STIC-AMSUD 19–STIC-01 *ACCON - Algorithms for the capacity crunch problem in optical networks*.

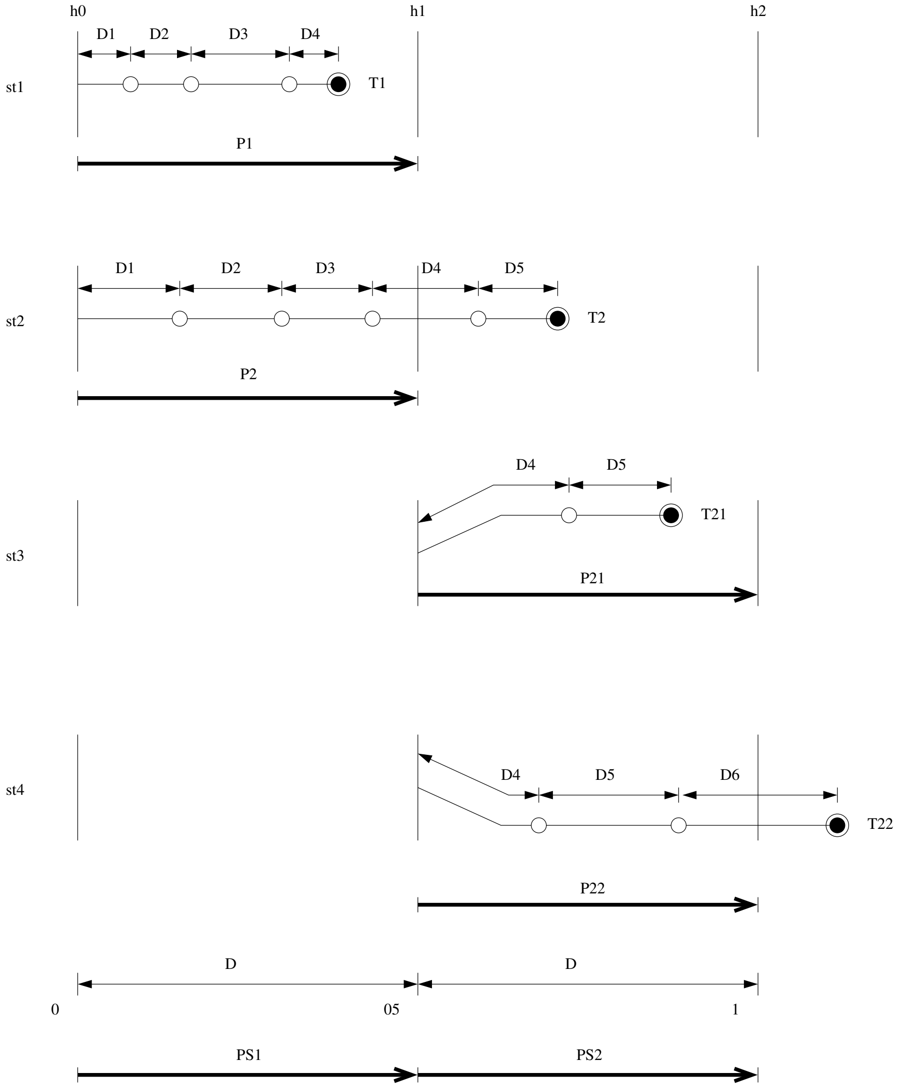

Figure 9: Permutation Monte Carlo over Splitting/CP

```
Algorithm 8 $q_{PS} = PS(\phi, S, p, \alpha, N_T)$
Prod $\leftarrow 1$
for $i = 1$ to $N$ do
    $Split(\phi, S, p, \alpha, N_T, 0)$
    for $j = 1$ to $N_T$ do
        $Prod \leftarrow Prod \times (Sum_j / Num_j)$
    end for
end for
return $q_{PS} = Sum/N$
function $Split(\phi, S, p, N_T, \alpha, N_T, z)$
for $i = 1$ to $\alpha_z$ do
    Sample a permutation, $\omega$, up to the critical component, and update $X$
    $Sum_z \leftarrow Sum_z + \gamma(\omega)$
    $Num_z \leftarrow Num_z + 1$
    if $\phi(X) = 0 \wedge z \leq N_T$ then
        Cancel all the repairs beyond $h_{z+1}$
        $Split(\phi, S, p, \alpha, N_T, z + 1)$
    end if
end for
return
```

Figure 10: Pseudocode for Permutation Monte Carlo over Splitting/CP.

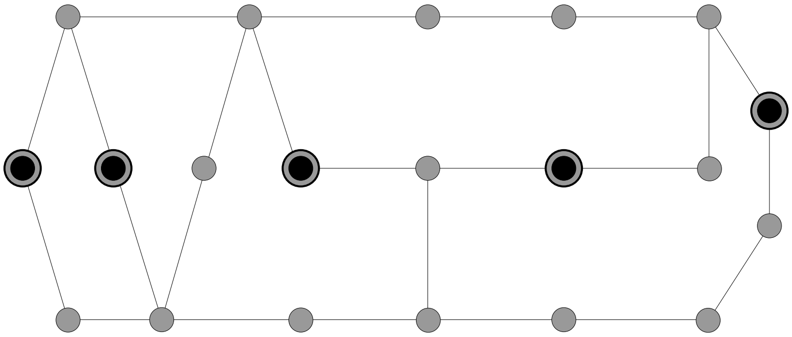

Figure 11: Arpanet network: tests 1 and 2

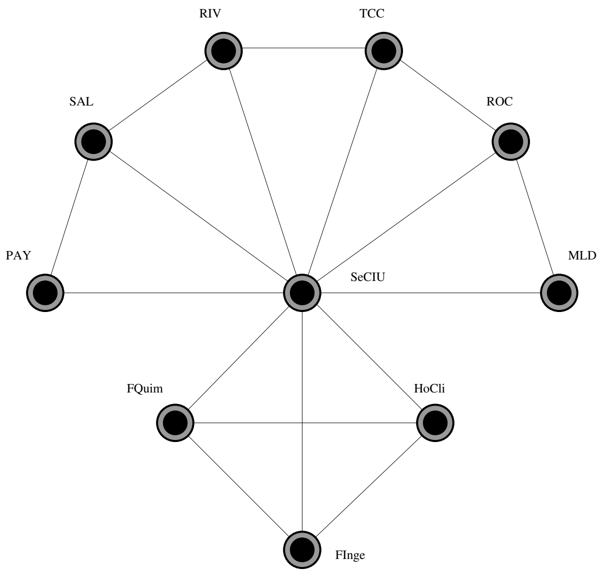

Figure 12: RAU network: tests 3 and 4

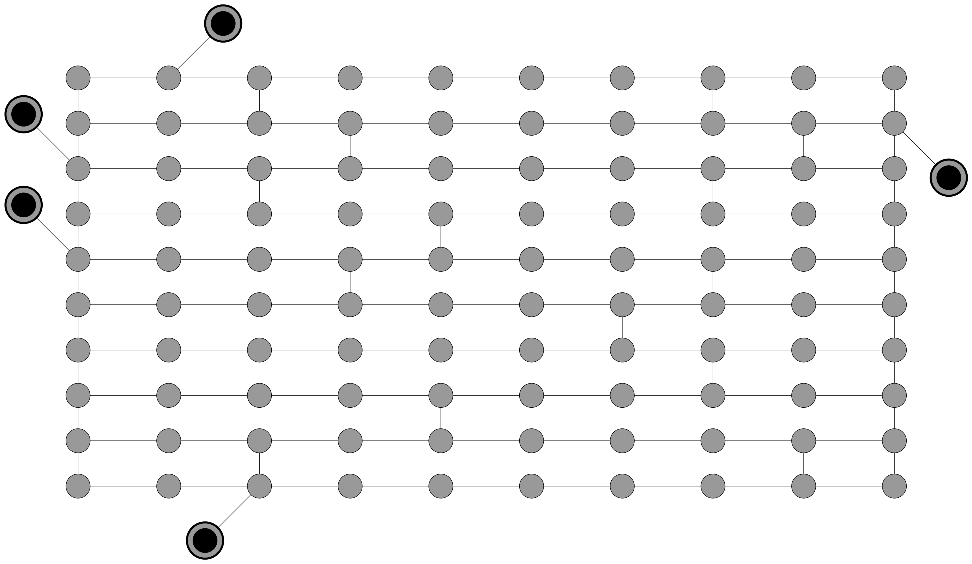

Figure 13: Easton Wong network: tests 5 and 7

Problem	**CLR**			**DCR, d=7**		
Edge reliab.	0.9	0.99	0.999	0.9	0.99	0.999
CMC						
Unrel.	1.26E-01	9.41E-04	8.00E-06	4.83E-01	5.03E-02	4.87EE-03
Variance	1.10E-07	9.40E-10	8.00E-12	2.50E-07	4.78E-08	4.84E-09
Time (ms)	5662	6100	5489	9733	13181	13300
RVR						
Unrel.	1.25E-01	9.53E-04	9.00E-06	4.82E-01	5.02E-02	5.00E-03
Variance	9.89E-07	7.49E-10	5.18E-13	6.04E-07	2.87E-09	3.31E-12
Time (ms)	4625	5031	5109	8540	10070	10161
Rel.Eff.	0.14	1.52	16.59	0.47	21.80	1913.88
FMC						
Unrel.	1.25E-01	9.61E-04	9.10E-06	4.28E-01	5.02E-02	5.00E-03
Variance	1.51E-07	1.90E-10	2.79E-14	3.94E-07	6.01E-08	9.66E-10
Time (ms)	11735	11735	11735	13906	13906	13906
Rel.Eff.	0.35	2.57	134.13	0.44	0.75	4.79
PMC						
Unrel.	1.26E-01	9.55E-04	9.04E-06	4.82E-01	5.02E-02	5.01E-03
Variance	2.92E-08	1.93E-11	2.50E-15	1.04E-07	7.50E-09	9.71E-11
Time (ms)	12651	12604	12581	20902	20953	21218
Rel.Eff	1.69	23.60	1296.21	1.12	4.01	31.25
Splitting/CP						
Unrel.	1.25E-01	9.57E-04	8.99E-06	4.81E-01	4.98E-02	5.03E-03
Variance	3.25E-07	3.35E-11	4.10E-15	1.52E-06	5.36E-08	1.52E-09
Time (ms)	1040	6990	13130	1295	3743	3614
Rel.Eff	1.84	24.47	815.75	1.23	3.14	11.75
PMC/SPL						
Unrel.	1.25E-01	9.51E-04	9.04E-06	4.84E-01	5.04E-02	4.99E-03
Variance	2.19E-07	3.15E-11	4.30E-15	6.49E-07	7.01E-08	1.12E-09
Time (ms)	1937	6845	8776	4096	2771	2153
Rel.Eff	1.47	26.59	1163.7	0.92	3.24	26.79

Table 1: Results for Tests 1 and 2

Problem	**CLR**			**DCR, d=4**		
Edge reliab.	0.9	0.99	0.999	0.9	0.99	0.999
CMC						
Unrel.	3.00E-02	2.15E-04	2.00E-06	4.02E-02	2.21E-04	2.00E-06
Variance	2.91E-08	2.15E-10	2.00E-12	3.84E-08	2.31E-10	2.00E-12
Time (ms)	3056	2732	2609	10107	8233	8193
RVR						
Unrel.	3.00E-02	2.12E-04	2.01E-06	4.04E-02	2.18E-04	2.01E-06
Variance	2.65-07	2.73E-10	1.62E-13	5.25E-07	2.35E-10	1.53E-13
Time (ms)	1642	1538	1456	5182	5181	4864
Rel.Eff.	0.20	1.40	22.12	0.14	1.56	22.02
FMC						
Unrel.	2.98E-02	2.09E-04	2.01E-06	4.02E-02	2.20E-04	1.99E-06
Variance	1.89E-08	1.65E-11	2.26E-15	4.14E-08	2.04E-11	2.53E-15
Time (ms)	6266	6266	6266	10438	10438	10438
Rel.Eff.	0.75	5.68	368.49	0.90	8.93	620.51
PMC						
Unrel.	2.98E-02	2.10E-04	2.01E-06	4.03E-02	2.20E-04	2.02E-06
Variance	6.23E-09	2.20E-12	2.62E-16	6.81E-09	2.20E-12	2.62E-16
Time (ms)	6275	6276	6254	12631	12582	12580
Rel.Eff	2.27	42.58	3184.65	4.51	68.74	4971.73
Splitting/CP						
Unrel.	2.97E-02	2.10E-04	2.01E-06	4.02E-02	2.18E-04	2.02E-06
Variance	3.12E-08	2.89E-12	3.98E-16	7.01E-08	2.26E-12	2.69E-16
Time (ms)	1278	3987	5896	3759	31924	45829
Rel.Eff	2.23	50.97	2223.72	1.47	26.38	1329.22
PMC/SPL						
Unrel.	3.00E-02	2.10E-04	1.99E-06	4.04E-02	2.19E-04	1.98E-06
Variance	4.93E-08	2.21E-12	2.38E-16	5.98E-08	4.24E-12	5.22E-15
Time (ms)	957	5177	5054	4280	15666	8766
Rel.Eff	1.89	51.33	4338.19	1.53	28.62	358.11

Table 2: Results for Tests 3 and 4

Problem	**EW CLR**			**Ship CLR**		
Edge reliab.	0.9	0.99	0.999	0.9	0.99	0.999
CMC						
Unrel.	4.81E-01	4.96E-02	5.04E-03	3.82E-03	2.00E-06	0.00E+00
Variance	2.50E-07	4.72E-08	5.02E-09	3.80E-09	4.00E-12	0.00E+00
Time (ms)	24219	14546	13469	38203	25140	21937
RVR						
Unrel.	4.83E-01	4.93E-02	4.99E-03	4.21E-03	1.01E-06	1.00E-09
Variance	1.14E-04	4.68E-08	5.57E-11	1.97E-06	1.15E-16	9.33E-24
Time (ms)	1922	1857	1843	15727	17698	18158
Rel.Eff.	0.028	7.90	658.21	0.0047	49332.79	-
Splitting/CP						
Unrel.	4.80E-01	4.89E-02	4.97E-03	3.79E-03	3.02E-06	3.06E-09
Variance	7.34E-07	1.23E-07	6.36E-10	1.17E-08	3.09E-15	1.06E-20
Time (ms)	7868	5545	20418	5963	30473	15474
Rel.Eff	1.05	1.00	5.20	2.09	1067.95	-
	EW DCR, d=20			**Ship DCR, d=10**		
Edge reliab.	0.9	0.99	0.999	0.9	0.99	0.999
CMC						
Unrel.	8.37E-01	1.03E-01	9.93E-03	1.53E-01	1.28E-03	1.90E-05
Variance	1.37E-07	9.31E-08	9.83E-09	1.30E-07	1.28E-09	1.90E-11
Time (ms)	33266	47578	48187	47297	37281	30875
RVR						
Unrel.	8.35E-01	1.05E-01	1.00E-02	1.55E-01	1.24E-03	9.67E-06
Variance	5.38E-05	3.32E-06	5.81E-09	3.41E-04	2.64E-07	1.43E-10
Time (ms)	4568	5125	5065	19648	22997	23480
Rel.Eff.	0.02	0.26	16.10	0.00	0.01	0.17
Splitting/CP						
Unrel.	8.35E-01	1.05E-01	1.00E-02	1.53E-01	1.27E-03	1.25E-05
Variance	3.42E-06	8.90E-07	9.80E-09	1.01E-06	3.36E-10	1.14E-13
Time (ms)	1268	4667	12517	13632	22468	25884
Rel.Eff	1.05	1.07	3.86	0.45	6.32	198.80

Table 3: Results for Tests 5, 6, 7 and 8

Metric	**CMC**	**RVR**	**PMC**	**Splitting/CP**	**PMC/SPL**
Unrel.	4.25E-08	3.41E-08	3.21E-08	3.21E-08	3.23E-08
Variance	4.80E-14	2.22E-16	2.72E-22	1.20E-19	1.27E-19
Time (ms)	251	2690	704	8434	14360
Rel. Efficiency	1.00	20.17	6.29E7	11904.20	6606.28

Table 4: Results for Test 9

References

Amrein, M., Künsch, H.R., 2011. A variant of importance splitting for rare event estimation: Fixed number of successes. *ACM Trans. Model. Comput. Simul.* 21, 2, 13:1–13:20.

Arora, S., Barak, B., 2009. *Complexity Theory: A Modern Approach*. Cambridge University Press.

Balázs, M., 2005. Sum of independent exponentials. https://people.maths.bris.ac.uk/ mb13434/sumexp.pdf.

Ball, M.O., 1980. Complexity of network reliability computations. *Networks* 10, 2, 153–165.

Ball, M.O., 1986. Computational complexity of network reliability analysis: An overview. *IEEE Transactions on Reliability* 35, 3, 230 –239.

Botev, Z., L'Ecuyer, P., Tuffin, B., 2018. Reliability Estimation for Networks with Minimal Flow Demand and Random Link Capacities. Pp:1-18.

Botev, Z.I., Kroese, D.P., 2012. Efficient monte carlo simulation via the generalized splitting method. *Statistics and Computing* 22, 1, 1–16.

Botev, Z.I., L'Ecuyer, P., Simard, R., Tuffin, B., 2016. Static network reliability estimation under the Marshall-Olkin copula. *ACM Transactions on Modeling and Computer Simulation* 26, 2, Article 14.

Canale, E., Cancela, H., Piccini, J., Robledo, F., Romero, P., Rubino, G., Sartor, P., 2015a. Recursive variance reduction method in stochastic monotone binary systems. In *Reliable Networks Design and Modeling (RNDM), 2015 7th International Workshop on*, pp. 135–141.

Canale, E., Cancela, H., Robledo, F., Romero, P., Sartor, P., 2015b. Diameter constrained reliability: Complexity, distinguished topologies and asymptotic behavior. *Networks* 66, 4, 296–305.

Canale, E., Cancela, H., Robledo, F., Romero, P., Sartor, P., 2015c. Full complexity analysis of the diameter-constrained reliability. *International Transactions in Operational Research* 22, 5, 811–821.

Canale, E., Cancela, H., Robledo, F., Rubino, G., Sartor, P., 2013. On computing the 2-diameter -constrained k -reliability of networks. *International Transactions in Operational Research* 20, 1, 49–58. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1475-3995.2012.00864.x.

Canale, E., Robledo, F., Romero, P., Sartor, P., 2014. Monte Carlo methods in diameter-constrained reliability. *Optical Switching and Networking* 14, Part 2, 0, 134 – 148. Special Issue on RNDM 2013.

Cancela, H., El Khadiri, M., 1995. A recursive variance-reduction algorithm for estimating communication-network reliability. *IEEE Transactions on Reliability* 44, 4, 595–602.

Cancela, H., Ferreira, G., Guerberoff, G., Robledo, F., Romero, P., 2018. Building reliability bounds in stochastic binary systems. In *2018 10th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 1–7.

Cancela, H., Khadiri, M.E., 2003. The recursive variance-reduction simulation algorithm for network reliability evaluation. *IEEE Transactions on Reliability* 52, 2, 207–212.

Cancela, H., Khadiri, M.E., Rubino, G., 2012. A new simulation method based on the RVR principle for the rare event network reliability problem. *Annals OR* 196, 1, 111–136.

Cancela, H., Murray, L., Rubino, G., 2019. Efficient estimation of stochastic flow network reliability. *IEEE Transactions on Reliability* 68, 3, 954–970.

Cancela, H., Robledo, F., Rubino, G., Sartor, P., 2014. Efficient estimation of distance-dependent metrics in edge-failing networks. *International Transactions in Operational Research* 21, 2, 199–213. https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12065.

Cancela, H., Urquhart, M., 2002. RVR simulation techniques for residual connectedness network reliability evaluation. *IEEE Transactions on Computers* 51, 4, 439–443.

Carazas, F., Salazar, C., Souza, G., 2011. Availability analysis of heat recovery steam generators used in thermal power plants. *Energy* 36, 6, 3855–3870.

Colbourn, C.J., 1999. Reliability Issues In Telecommunications Network Planning. In *Telecommunications network planning, chapter 9*, Kluwer Academic Publishers, pp. 135–146.

Cook, S.A., 1971. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, ACM, New York, NY, USA, pp. 151–158.

Dash, R., Barpanda, N., Tripathy, P., Tripathy, C., 2012. Network reliability optimization problem of interconnection network under node-edge failure model. *Applied Soft Computing* 12, 8, 2322 – 2328.

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

Easton, M., Wong, C., 1980. Sequential destruction method for Monte Carlo evaluation of system reliability. *IEEE Transactions on Reliability* R–29, 1.

Elperin, T., Gertsbakh, I.B., Lomonosov, M., 1991. Estimation of network reliability using graph evolution models. *IEEE Transactions on Reliability* 40, 5, 572–581.

Endharta, A., Yun, W., Ko, Y., 2018. Reliability evaluation of circular k-out-of-n: G balanced systems through minimal path sets. *Reliability Engineering and System Safety* 180, 226–236.

Fernandez, M., Williams, S., 2010. Closed-form expression for the poisson-binomial probability density function. *Aerospace and Electronic Systems, IEEE Transactions on* 46, 2, 803–817.

Fishman, G., 1996. *Monte Carlo:.* Springer Series in Operations Research and Financial Engineering. Springer.

Garvels, M.J.J., 2000. The Splitting Method in Rare Event Simulation. Ph.D. thesis, Faculty of mathematical Science, University of Twente, The Netherlands.

Gertsbakh, I., Rubinstein, R., Shpungin, Y., Vaisman, R., 2014a. Permutational methods for performance analysis of stochastic flow networks. *Probability in the Engineering and Informational Sciences* 28, 1, 21–38.

Gertsbakh, I.B., Shpungin, Y., 2009. *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo* (1st edn.). CRC Press, Inc., USA.

Gertsbakh, I.B., Shpungin, Y., Vaisman, R., 2014b. *Ternary Networks - Reliability and Monte Carlo*. Springer Briefs in Electrical and Computer Engineering. Springer.

Gertsbakh, I.B., Shpungin, Y., Vaisman, R., 2016. D-spectrum and reliability of a binary system with ternary components. *Probability in the Engineering and Informational Sciences* 30, 1, 25–39.

Glasserman, P., Heidelberger, P., Shahabuddin, P., Zajic, T., 1996. Splitting for rare event simulation: Analysis of simple cases. In *Proceedings of the 28th conference on Winter simulation, WSC 1996, Coronado, CA, USA, December 8-11, 1996*, pp. 302–308.

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

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

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

Karp, R.M., 1972. Reducibility among combinatorial problems. In Miller, R.E. and Thatcher, J.W. (eds), *Complexity of Computer Computations*. Plenum Press, pp. 85–103.

Kobayashi, K., Morohosi, H., Oyama, T., 2009. Applying path-counting methods for measuring the robustness of the network-structured system. *International Transactions in Operational Research* 16, 3, 371–389. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1475-3995.2008.00688.x.

L'Ecuyer, P., Demers, V., Tuffin, B., 2007. Rare events, splitting, and Quasi-Monte Carlo. *ACM Trans. Model. Comput. Simul.* 17, 2.

L'Ecuyer, P., Le Gland, F., Lezaud, P., Tuffin, B., 2009. Splitting techniques. In Rubino, G. and Tuffin, B. (eds), *Rare Event Simulation using Monte Carlo Methods*. John Wiley & Sons, chapter 3, pp. 39–61.

L'Ecuyer, P., Rubino, G., Saggadi, S., Tuffin, B., 2011. Approximate Zero-Variance Importance Sampling for Statistic Network Reliability Estimation. *IEEE Transactions on Reliability - TR* 60, 3, 590–604.

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

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

Masri, H., Krichen, S., Guitouni, A., 2019. Metaheuristics for solving the biobjective single-path multicommodity communication flow problem. *International Transactions in Operational Research* 26, 2, 589–614. https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12378.

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

Murray, L., Cancela, H., Rubino, G., 2013. A splitting algorithm for network reliability. *IIE Transactions* 45, 2, 177–189.

Pérez-Rosés, H., 2018. Sixty years of network reliability. *Mathematics in Computer Science* 12, 3, 275–293.

Perman, M., Senegacnik, A., Tuma, M., 1997. Semi-markov models with an application to power-plant reliability analysis. *IEEE Transactions on Reliability* 46, 4, 526–532.

Petingi, L., Rodríguez, J., 2001. Reliability of networks with delay constraints. In *Congressus Numerantium*, Vol. 152, pp. 117–123.

Provan, S.J., Ball, M.O., 1983. The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected. *SIAM Journal on Computing* 12, 4, 777–788.

Raad, D., Sinske, A., Van Vuuren, J., 2009. Robust multi-objective optimization for water distribution system design using a meta-metaheuristic. *International Transactions in Operational Research* 16, 5, 595–626. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1475-3995.2009.00705.x.

Rath, S., Gendreau, M., Gutjahr, W.J., 2016. Bi-objective stochastic programming models for determining depot locations in disaster relief operations. *International Transactions in Operational Research* 23, 6, 997–1023. https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12163.

Romero, P., 2016. Duality in stochastic binary systems. In *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 85–91.

Romero, P., 2019. Challenges in stochastic binary systems. In *2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 1–8.

Rosenthal, A., 1977. Computing the reliability of complex networks. *SIAM Journal on Applied Mathematics* 32, 2, 384–393.

Rubino, G., Tuffin, B., 2009. *Rare event simulation using Monte Carlo methods*. Wiley.

Saling, K.G., White Jr., K.P., 2013. Integrating probabilistic design and rare-event simulation into the requirements engineering process for high-reliability systems. *International Transactions in Operational Research* 20, 4, 515–531. https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12023.

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

Stoer, M., 1992. *Design of survivable networks*. Lecture notes in mathematics. Springer-Verlag.

Tien, I., Kiureghian, A.D., 2016. Algorithms for bayesian network modeling and reliability assessment of infrastructure systems. *Reliability Engineering & System Safety* 156, 134 – 147.

Villén-Altamirano, M., Villén-Altamirano, J., 1991. Restart: A method for accelerating rare events simulations. In *Proceedings of the 13th International Teletraffic Congress*, North–Holland, pp. 71–76.

Villén-Altamirano, M., Villén-Altamirano, J., 2002. Analysis of restart simulation: Theoretical basis and sensitivity study. *European Transactions on Telecommunications* 13, 4, 373–385. https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.4460130409.

Wang, G., Peng, R., Xing, L., 2018. Reliability evaluation of unrepairable k-out-of-n: G systems with phased-mission requirements based on record values. *Reliability Engineering and System Safety* 178, 191–197.

Wu, B., Maya, B., Limnios, N., 2020. Using semi-markov chains to solve semi-markov processes. *Methodology and Computing in Applied Probability*. doi: 10.1007/s11009-020-09820-y.