Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

00 (2021) 1–31

Reliability and Importance Measure Analysis of Networks with Shared Risk Link Groups

Radislav Vaisman[a,*], Yuting Sun[a]

[a] School of Mathematics and Physics, The University of Queensland Brisbane 4072, Australia

Abstract

We consider the problem of assessing the reliability and the Birnbaum importance measure of complex networks under the shared link risk group (SRLG) failure scenario. With a view to the fact that SRLG failures can cause a global system breakdown, the reliability and the importance measure analysis is of fundamental importance to the study of critical infrastructures such as sensor and cross-layer networks, supply chains, and other complex systems that support the essential functioning of our society and economy. The mathematical complexity of the reliability and the importance measure calculation implies that one has to rely on approximation techniques, since no analytical method for solving this problem in reasonable computation time is known to exist. This study will build upon the Permutation Monte Carlo paradigm. The major advantage of the proposed solution is that it allows us to obtain reliable estimates of both the network reliability and the Birnbaum importance measure using the same algorithmic machinery. We show that the suggested algorithm is easy to implement and that the method is scalable to meet real-life network sizes.

Keywords: Shared risk link groups, reliability, Birnbaum importance measure, complex networks, optical communication

1. Introduction

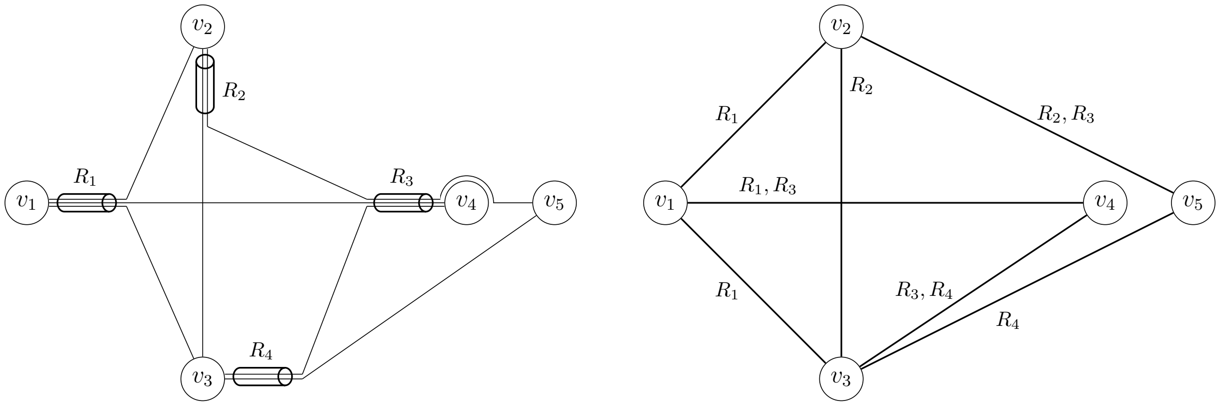
We investigate the problem of *reliability* and *importance measure* (IM) analysis of complex networks where users communicate with each other via end-to-end paths, under the *shared risk link group* (SRLG) failure scenario [1]. In this setting, an assemblage of network links can share a specific vulnerability, that is, these links belong to a single SRLG. As a consequence, a certain failure event can cause a simultaneous downfall of several network edges at the same time. Shared risk problems are common in transportation and communication network studies. For example, the Howard Street Tunnel fire which occurred in Baltimore on July 18, 2001, had a severe impact on many essential services. Among these are streets closure, rerouting of bus lines, suspension of train services, and power outages [2]. When dealing with complex fiber and optical networks, the AT&T's study showed that a network link may belong to over 100 SRLGs and that every SRLG can associate together many network links [3]. As a consequence, a failure in one or several SRLGs, can result in a global failure that will cause a disconnection of network peers, and thus compromise the integrity of the corresponding critical infrastructure. Due to the problem importance, the task of designing reliable networks, namely, networks that are resilient to failure events, was extensively considered in previous studies. However, to the best of our knowledge, the *combined* analysis of network reliability and IM, which we call the SRLG-RI, was not examined previously. This study addresses the gap by utilizing a Monte Carlo

*Corresponding author

Email addresses: r.vaisman@uq.edu.au (Radislav Vaisman), skye.sun@uq.edu.au (Yuting Sun)

based machinery, which is suitable for the analysis of *both* the network's reliability and the IM. While the reliability assessment allows us to analyze the system resilience to failures, the corresponding IM analysis provides a way to *rank* SRLG components according to their *importance* [4, 5]. In this study, component importance is proportional to its contribution to the overall network reliability. Therefore, the IM analysis allows to optimally rectify potential hazards and increase the overall system reliability in the most *effective* and *economical* (financial) fashion, by allocating available resources to the set of most important components.

The SRLG setting, in which a single failure event can affect several links, was first introduced by Koch et. al. [1]. An illustrative example of a fiber network is depicted in Figure 1 (a); here fibers are aggregated into cables that are usually placed in ducts and berried under the ground. A duct can be damaged due to some breakdown event such as construction work or an earthquake. In this case, all fibers in this duct fail and we say that these fibers are associated with (at least) one risk group, that is, they share a particular risk factor. In real life, there are numerous reasons that can cause SRLG failures. For example, it is common for several optical channels to use a single fiber. In addition, fibers (that can even be associated with different network providers), can use the same conduit. Moreover, fibers usually share a communication infrastructure (repeaters). Finally, modern networks are often multilayered, that is, it is common to observe virtual private networks on top of a fiber network. Namely, there can be shared links at any level.



(a) Physical fiber link topology

(b) SRLG fiber link topology

Figure 1: A small fiber link network with five vertices and four cables. The set of vertices can represent data centers, local networks, virtual machines, switches, routers, points of presence, etc..

Figure 1(a) and Figure 1(b), show the physical layout and the corresponding SRLG topology of a fiber network, respectively. There are five data-centers (v_1, v_2, v_3, v_4, v_5) and four cables denoted by R_1, R_2, R_3, and R_4. Each cable may contain several fibers, while a given fiber may belong to several cables, as shown in Figure 1(a). There are four SRLGs in this example, namely, $R_1 = \{(v_1, v_2), (v_1, v_4), (v_1, v_3)\}$, $R_2 = \{(v_2, v_3), (v_2, v_5)\}$, $R_3 = \{(v_2, v_5), (v_1, v_4), (v_3, v_4)\}$, and $R_4 = \{(v_3, v_4), (v_3, v_5)\}$. For instance, a failure of R_1 will lead to the failure of links (v_1, v_2), (v_1, v_3), and (v_1, v_4), simultaneously. In addition, by noting that the (v_2, v_5) link belongs to both the R_2 and the R_3, we conclude that a failure of either R_2 or R_3 (or both), will interrupt the connection between v_2 and v_5.

Due to the importance of the SRLG problem in communication networks, it attracted a significant amount of research [6, 7, 8, 9, 10, 11, 12]. The load control optimization under the SRLG setting and the enumeration of regional failures in links that are caused by disasters, were considered by Liu et. al. [7], and by Tapolcai et. al. [8], respectively. Special attention was drawn to the *(diverse) routing* problems [11], where the corresponding optimization task is computationally hard since it was shown to be NP-complete [9, 10]. As a result of the NP computational complexity, previous research focused mainly on heuristic procedures, evolutionary algorithms, and integer programming methods [11, 12]. The concept of IM is fundamental in the reliability theory. Therefore, the extent of the available literature that deals with various types of IMs is truly colossal. For example, Zheng et. al. consider IM analysis in smart electric power grid systems [13]. Zheng et. al. evaluate the availability of smarts grids and show how one can find the system's vulnerabilities. A data-driven framework which can be used for identifying critical components in complex technical infrastructures was proposed by Lu et. al. [14], where the authors combine IM analysis with modern machine learning methods. Lisnianski et. al. consider both the availability and the performability analysis in large-scale multi-state systems [15, 16]. Specifically, the authors consider a Markov multi-state model for large-scale highly responsive distributed systems and develop the corresponding measure and an L_Z-transform based algorithm. For additional recent studies, we refer to Dui et. al. [17], Do et. al. [18], and Shi et. al. [19]. In these works, Dui et. al. provide an analysis of IMs while also considering the system lifetime, Do et. al. study a conditional-based importance measure, which takes into account economic aspects such as maintenance costs, and Shi et. al. introduce a maintenance optimization and the corresponding algorithm which uses IM for optimal maintenance grouping in multi-component systems. Finally, Bistouni et. al. [20], examine the analysis of reliability and IM for Ethernet ring mesh networks, and Chang et. al. [21], consider the reliability estimation and sensitivity analysis of multi-state manufacturing network with joint buffers using simulation. Nevertheless, despite the problem's importance and the extensive amount of previous work, to the best of our knowledge, the task of the simultaneous analysis of both the reliability and the IM for general graphs, namely, the SRLG-RI problem, was not previously addressed.

In this study, we focus on the SRLG-RI problem which can be stated as follows. Given a network and a set of SRLGs, where each SRLG failure occurs with some predefined probability, find the probability that the network is fully connected. That is, find the probability that all network peers are able to communicate with each other. This question corresponds to the classic network reliability problem [22, 23]. In addition, the IM problem concerns with finding an *importance ranking* of SRLG components. From the practical perspective, the IM analysis allows us to determine the set of specific SRLGs, whose reliability should be intensified with a view to increasing the overall network resilience in the most effective and economical fashion.

Similar to routing optimization tasks that belong to the NP complexity class, SRLG-RI is also computationally hard. However, SRLG-RI is not an optimization, but it is rather a *counting* problem. In particular, SRLG-RI belongs to the #P complexity class [24, 25], since one can show that SRLG-RI is a generalization of the *k-terminal reliability* problem [25]; the complexity of SRLG-RI will be further discussed in Section 2. For some #P complete problems, there exist good approximation schemes [26, 27]. In addition, when dealing with network reliability, there are several methods that can handle small-sized graphs and a few specific graph topologies such as series-parallel and directed acyclic networks [28, 29]. However, there is no known efficient approximation scheme for the general network reliability problem [27]. Since SRLG-RI generalizes the k-terminal network reliability problem, no analytical method for solving SRLG-RI in reasonable computation time is known to exist and as a consequence, one has to rely on approximation techniques.

Our objective is to handle real-life networks, and therefore, we would like to apply a method that is fast from the computational point of view. In order to satisfy this requirement, we propose to analyze a network invariant called the *spectra* [30, 31, 32], since the latter allows us to obtain both the network reliability and to perform the corresponding IM analysis using the *same* algorithmic machinery. While the analytical calculation of spectra is computationally hard, we propose to estimate it using the *Permutation Monte Carlo* (PMC) approach [22]. As we show in Section 3, the PMC algorithm can be used for both the reliability and the IM analysis. In particular, as soon as the spectra object is available, one can estimate the reliability of the network for *any* failure probability of SRLG components in polynomial time in the number of SRLGs, and obtain the network's components IM via the so-called *Birnbaum Measure of Component Importance* [33, 34]. Since this measure corresponds to the importance ranking of SRLGs, a network designer who wishes to increase the overall network reliability, can perform a smart and economical decision making by determining the set of components of maximum importance (these components will have the largest Birnbaum measure values).

The major contribution of this study is as follows.

1. Our first contribution is that we address the gap of analyzing complex network reliability and IM under SRLG failure scenario. The IM analysis allows to rank SRLG components according to their importance and thus the network administrator can increase the reliability of the set of the most important components, while introducing a maximal improvement to the overall network reliability and utilizing the least possible economical effort.
2. Our second contribution is that we show that one can apply a well-established technique (PMC), that is simple to implement and that can handle real-life networks. It is important to note that the proposed procedure does not require any parameter tuning, except for the sample size. According to the AT&T study, the computational aspect is especially important for communication fiber and optical networks that can have hundreds of SRLGs. In addition, we note that the same algorithmic method is used for both the reliability and the IM analysis. The

simplicity of the proposed method allows to introduce a rigorous efficiency analysis (please see Theorem 1 in Section 3.2).

3. A very important aspect of the proposed method is that as soon as the spectra calculation is completed, one can readily use the spectra object for the analysis of both the reliability and the IM for *any* SRLG's failure probability (provided that every SRLG fails with the same probability). In particular, the network reliability can be obtained in polynomial time in the number of SRLGs, and the IM analysis can be completed using a simple statistical procedure. The latter will be detailed in Section 3 and Section 4.
4. Finally, we provide a research software package that can handle real-life SRLG-RI problems, and is capable of achieving good solutions while using reasonable computation time. To the best of our knowledge, there exists no other freely available or non-proprietary software that can operate under the SRLG-RI setting.

To summarize, this study proposes an efficient method for analyzing reliability and IM of complex networks under the SRLG failure scenario. Specifically, we present a general technique for the *estimation* of reliability and IM of networks with shared link risk groups. Our procedure also allows us to identify the set of risks of major importance. Having in mind that each SRLG incorporates the dependence between its corresponding links, our methodology assumes independence between SRLGs. While the method is designed with a view to handling real-life networks, it has one major and one minor limitation that are going to be detailed in forthcoming sections. The major limitation is that our method is designed for the scenario, in which every SRLG component has the *same* failure probability. However, this assumption is reasonable when working with communication networks, since we can expect to deal with comparable reliability characteristics of electrical components. In addition, this assumption allows us to design a simple and computationally efficient algorithm for both the reliability and the IM analysis. The minor limitation is that the method is not suitable for an adequate estimation of spectra that involves very small (spectra) components. While, the latter can be resolved by utilizing techniques that are similar to the ones discussed in Vaisman et. al. [25, 35], the resulting algorithmic complexity will force the user to compromise on the corresponding computational efficiency.

The rest of the paper is organized as follows. In Section 2 we formally define the SRLG-RI problem setting. The proposed solution procedure is described in Section 3. In particular, we give a rigorous description of the PMC algorithm and explain how it can be used for the estimation of both the reliability and the IM of complex networks under the SRLG failure scenario. In Section 4 we present an extensive experimental study that demonstrates the performance of the proposed methods when applied to several synthetic graphs and one real-life network. Finally, Section 5 summarizes our findings and outlines limitations and possible directions for future research.

2. Problem definition

In this section, we formally define the SRLG-RI problem. To start with, consider a (complex) network, which can be represented using a finite undirected graph $G = (V, E)$, where V and E are the vertex and the edge sets, respectively. When working with communication networks, vertices can exemplify data centers, local networks, virtual machines, switches, routers, points of presence, etc.. For example, in Figure 1, we have the vertex set $V = \{v_1, v_2, v_3, v_4, v_5\}$, and the edge set $E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_5), (v_3, v_4), (v_3, v_5)\}$. An SRLG $R \subseteq E$ is a collection of links that share the same vulnerability. The set of *all* SRLGs is denoted by $\mathcal{R}$. In the example from Figure 1, $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$.

An important ingredient of the reliability analysis is the *structure function* $\psi : \{0, 1\}^n \to \{0, 1\}$ [22, 30, 36], where $|\mathcal{R}| = n$. The function's input is a binary vector of length n, namely $\mathbf{x} = (x_1, \ldots, x_n)$, where $x_i \in \{0, 1\}$ is an indicator variable that signifies that the ith SRLG component (R_i) for $i \in \{1, \ldots, n\}$ is operational ($x_i = 1$), or failed ($x_i = 0$). Alternatively, we say that a component is operational or failed if it is in the *up* or in the *down* state, respectively. The structure function outputs $\psi(\mathbf{x}) = 1$ if the network is operational (the network is in the *up* state), or, $\psi(\mathbf{x}) = 0$ if the network is not operational (the network is in the *down* state).

In this study we work with the following failure regime; if an edge $e \in E$ belongs to two or more different SRLGs, an edge will be in the *down* state if at least one SRLG is in the *down* state. A failure regime is determined by the definition of the corresponding structure function. Under our setting, the structure function is defined as follows. Let $\mathcal{R}' \subseteq \mathcal{R}$ be a set of SRLGs, and define $G(\mathcal{R}') = (V, E')$ to be a subgraph of $G = (V, E)$, where

$$E' = \{e \in E \ : \ \exists \mathcal{R}'' \in \mathcal{R}' \text{ such that } e \in Z', \text{ and } \forall \mathcal{R}''' \in \mathcal{R} \setminus \mathcal{R}', \ e \notin \mathcal{R}'''\}. \qquad (1)$$

In other words, $G(\mathcal{R}')$ is a subgraph that contains all vertices of the V set, and all edges from the E set that are not in the *down* state. Then, the structure function is defined via

$$\psi(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } \mathcal{R}' = \{R_i \in \mathcal{R} \ : \ x_i = 1\} \text{ and } G(\mathcal{R}') \text{ is fully connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Remark 1 (Failure regime)**.** In practice, it can be beneficial to consider different failure regimes. For example, one can define an edge $e \in E$ to be in the *up* state, if it belongs to at least one SRLG that is in the *up* state. In addition, we might want to modify the network's *up* state criteria. For instance, we can declare that the network is in the *up* state if 80% of vertices are connected. Moreover, one might be interested in a congestion scenario, which is of major importance when dealing with communication networks. The latter can be achieved by fixing all physical links (that belong to a virtual link), to share the same SRLG. Of course, a change of a failure regime will require a redefinition of the structure function in (2). However, it is important to note that all methods developed in this paper can be applied to various failure regimes, provided that the structure function is properly defined; some examples are given in Section 4.

Next, we define $\mathbf{q} = (q_1, \ldots, q_n)$, to be a vector of failure probabilities, where $q_i \in [0, 1]$ for $1 \leq i \leq n$. Here, q_i stands for the failure probability of the ith SRLG component $R_i \in \mathcal{R}$, or, in other words, q_i is the probability of the ith SRLG to be in the *down* state. For convenience, we also define the corresponding (success) probability vector $\mathbf{p} = (p_1, \ldots, p_n)$, where $\mathbf{p} \overset{\text{def}}{=} \mathbf{1} - \mathbf{q} = (1 - q_1, \ldots, 1 - q_n)$. Consequently, for $1 \leq i \leq n$, p_i is the probability of R_i to be in the *up* state. The network reliability and the unreliability is denoted by $\Psi(p_1, \ldots, p_n) \overset{\text{def}}{=} r(G, \mathcal{R}, \mathbf{p}) \overset{\text{def}}{=} r(G, \mathcal{R}, \mathbf{q})$ and by $\overline{\Psi}(p_1, \ldots, p_n) \overset{\text{def}}{=} \bar{r}(G, \mathcal{R}, \mathbf{p}) \overset{\text{def}}{=} \bar{r}(G, \mathcal{R}, \mathbf{q})$, respectively.

Table 1 The nomenclature.

Notation	Meaning
V	vertex set
E	edge set
$e = (u, v);\ u, v \in V$	edge between two vertices (u and v)
$G = (V, E)$	undirected graph that represents a complex network
$\mathbf{x} = (x_1, \ldots, x_n)$	system state vector, $x_i \in \{0, 1\}$ for all $1 \leq i \leq n$; $x_i = 1$ and $x_i = 0$ signify the *up* and the *down* state of component i, respectively
$R \subseteq E$	shared risk link group
$\mathcal{R}$	set of all shared risk link groups; $\lvert\mathcal{R}\rvert = n$
$\psi : \{0, 1\}^n \to \{0, 1\}$	structure function
$\mathbf{q} = (q_1, \ldots, q_n)$	vector of failure probabilities
$\mathbf{p} = (p_1, \ldots, p_n)$	vector of success probabilities
$q_i = 1 - p_i \in [0, 1]$	failure probability of the SRLG $R_i \in \mathcal{R}$
$r(G, \mathcal{R}, \mathbf{p}) \overset{\text{def}}{=} r(G, \mathcal{R}, \mathbf{q})$	network reliability (the probability that the network is in the *up* state)
$\Psi(p_1, \ldots, p_n) \overset{\text{def}}{=} r(G, \mathcal{R}, \mathbf{p})$	network reliability
$r(G, \mathcal{R}, p) \overset{\text{def}}{=} r(G, \mathcal{R}, q)$	network reliability where $p = p_1 = p_2 = \cdots = p_n$
$\bar{r}(G, \mathcal{R}, \mathbf{p}) \overset{\text{def}}{=} \bar{r}(G, \mathcal{R}, \mathbf{q})$	network unreliability
$\overline{\Psi}(p_1, \ldots, p_n) \overset{\text{def}}{=} \bar{r}(G, \mathcal{R}, \mathbf{p})$	network unreliability
$\bar{r}(G, \mathcal{R}, p) \overset{\text{def}}{=} \bar{r}(G, \mathcal{R}, q)$	network unreliability where $p = p_1 = p_2 = \cdots = p_n$
BIM_i	Birnbaum measure of component importance of R_i for $1 \leq i \leq n$

In this work, we are interested in a special case for which $q_i = q_j = q$ for all $1 \leq i, j \leq n$. Under this setting, the reliability and the unreliability notation is simplified to $r(G, \mathcal{R}, p)$ and $r(G, \mathcal{R}, q)$, and to $\overline{r}(G, \mathcal{R}, p)$ and $\overline{r}(G, \mathcal{R}, q)$, respectively. For the rest of this paper, we use $\mathbf{p}$ and $\mathbf{q}$, p and q, and, p_i and q_i interchangeably. Finally, for each SRLG $R_i \in \mathcal{R}$, we define the corresponding *Birnbaum Measure of Component Importance* (BIM) via BIM_i for all $1 \leq i \leq n$. These BIMs are constructed in such a way that a decrease in the failure rate of the most important component, namely, of the component that has the largest BIM value, will cause the largest increase of the overall network reliability. Formally, the BIM of component i is defined via [33]

$$\text{BIM}_i = \frac{\partial \Psi(p_1, \ldots, p_n)}{\partial p_i} = \Psi(p_1, \ldots, p_{i-1}, 1, p_{i+1}, \ldots, p_n) - \Psi(p_1, \ldots, p_{i-1}, 0, p_{i+1}, \ldots, p_n). \quad (3)$$

Determining the importance of SRLGs constitutes the network IM analysis. The above notation is compactly described in Table 1, and we can proceed with the formal definition of the SRLG-RI problem.

Definition 1 (The SRLG-RI problem)**.** Given an undirected graph $G = (V, E)$, a set of SRLGs $\mathcal{R} = (R_1, \ldots, R_n)$, and an SRLG failure probability $q \in [0, 1]$, such that, $\mathbb{P}(\{\ R_i \text{ is in failed state}\}) = q$ for all $1 \leq i \leq n$, the solution of the SRLG-RI problem instance $(G, \mathcal{R}, p)$, is composed from answers to the following questions.

1. The system reliability (or unreliability) calculation. Namely, calculating

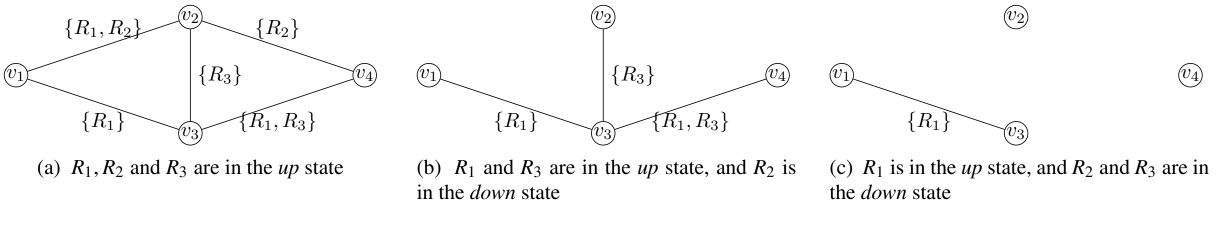
$$r(G, \mathcal{R}, p) = \mathbb{P}(\psi(X_1, \ldots, X_n) = 1), \quad (\text{or } \overline{r}(G, \mathcal{R}, p) = \mathbb{P}(\psi(X_1, \ldots, X_n) = 0)),$$

where X_i is a Bernoulli random variable, that is, $X_i \sim \mathsf{Bernoulli}(p)$ for $1 \leq i \leq n$.

2. The system IM calculation. Namely, calculating $\{\text{BIM}_i\}_{i=1}^n$ for $\{R_i\}_{i=1}^n$.

We next proceed with an instructive illustration of reliability and IM analysis when applied to a small network instance.

Example 1 (The bridge network)**.** Consider the bridge network in Figure 2(a). The network consists of four vertices and five edges. In addition, there are three SRLGs. Specifically $\mathcal{R} = \{R_1, R_2, R_3\}$, where $R_1 = \{(v_1, v_2), (v_1, v_3), (v_3, v_4)\}$, $R_2 = \{(v_1, v_2), (v_2, v_4)\}$, and $R_3 = \{(v_2, v_3)\}$. In this example, there are three possible failure events that correspond to each SRLG. Suppose that all SRLGs are operational (Figure 2 (a)). We can see that if say the R_2 group fails, then, the (v_1, v_2) and the (v_2, v_4) edges fail (Figure 2 (b)). Note that the network is still connected in the sense that all graph vertices form a single connected component. Suppose now that the SRLG R_3 fails, that is, we have two SRLGs in the *down* state, specifically, the R_2 and the R_3. In this case, edges (v_2, v_3) and (v_3, v_4) fail, and the corresponding network becomes disconnected (Figure 2 (c)). Namely, after the failure of R_2 and R_3, the networks enters the *down* state.



(a) R_1, R_2 and R_3 are in the *up* state

(b) R_1 and R_3 are in the *up* state, and R_2 is in the *down* state

(c) R_1 is in the *up* state, and R_2 and R_3 are in the *down* state

Figure 2: A bridge network with four vertices and five edges.

Next, let us assume that the probability of SRLGs R_1, R_2 and R_3 to be in the *up* state, is equal to p_1, p_2, and p_3, respectively. The bridge network in Figure 2 is very small and therefore it is also instructive in the sense that we can provide a complete characterization of the *structure function* $\psi(x_1, x_2, x_3)$. The bridge network reliability can be derived from Table 2; specifically, the network reliability is equal to

$$r(G, \mathcal{R}, (p_1, p_2, p_3)) \overset{\text{def}}{=} \mathbb{P}(\psi(X_1, X_2, X_3) = 1) = p_1\ q_2\ p_3 + p_1\ p_2\ q_3 + p_1\ p_2\ p_3. \quad (4)$$

Table 2 The bridge's network structure function.

$x_1\ x_2\ x_3$	ψ	$\Psi(p_1,p_2,p_3) \overset{\text{def}}{=} \mathbb{P}(X_1 = x_1, X_2 = x_2, X_3 = x_3)$
0 0 0	0	$q_1\ q_2\ q_3$
0 0 1	0	$q_1\ q_2\ p_3$
0 1 0	0	$q_1\ p_2\ q_3$
0 1 1	0	$q_1\ p_2\ p_3$
1 0 0	0	$p_1\ q_2\ q_3$
1 0 1	1	$p_1\ q_2\ p_3$
1 1 0	1	$p_1\ p_2\ q_3$
1 1 1	1	$p_1\ p_2\ p_3$

Since we are concerned with the special case for which $p_1 = p_2 = p_3 = p$, (4) simplifies to

$$r(G,\mathcal{R},q) = 2p^2q + p^3. \tag{5}$$

Next, we proceed with the IM analysis of the bridge network. From (3) and (4), we arrive at

$$\begin{aligned} \text{BIM}_1 &= q_2p_3 + p_2q_3 + p_2p_3 \underbrace{=}_{p_1=p_2=p_3=p} p^2 + 2pq, \\ \text{BIM}_2 &= -p_1p_3 + p_1q_3 + p_1p_3 \underbrace{=}_{p_1=p_2=p_3=p} pq, \\ \text{BIM}_3 &= p_1q_2 - p_1p_2 + p_1p_2 \underbrace{=}_{p_1=p_2=p_3=p} pq. \end{aligned} \tag{6}$$

Since it holds that $\text{BIM}_1 \geq \text{BIM}_2$ and $\text{BIM}_1 \geq \text{BIM}_3$, we conclude that the first SRLG, namely (R_1), is of major importance. In other words, if we would like to increase the overall network reliability but can only intensify the reliability of one SRLG, the R_1 SRLG will be the best candidate. At this stage, it is instructive to see the motivation for computing BIMs. Suppose that the SRLG component R_i $(1 \leq i \leq n)$, which fails with probability $q_i = 1 - p_i$, is replaced with a more reliable one. Specifically, assume that the new component's reliability is $p_i + \Delta$, where $\Delta > 0$. Then, the new overall system reliability will be increased by $\text{BIM}_i \times \Delta$ factor [22]. This means that the component with maximum BIM value has the largest contribution to the system reliability. In the bridge network example, by increasing p_1 by Δ, and combining this with (4), we arrive at

$$\begin{aligned} r(G,\mathcal{R},(p_1+\Delta,p_2,p_3)) &= (p_1+\Delta)\ q_2\ p_3 + (p_1+\Delta)\ p_2\ q_3 + (p_1+\Delta)\ p_2\ p_3 \\ &= p_1\ q_2\ p_3 + p_1\ p_2\ q_3 + p_1\ p_2\ p_3 + \Delta(q_2\ p_3 + p_2\ q_3 + p_2\ p_3) \\ &\underbrace{=}_{p_1=p_2=p_3=p} r(G,\mathcal{R},q) + \Delta(\underbrace{p^2+2pq}_{\text{BIM}_1}). \end{aligned}$$

Unfortunately, for larger real-life problems, the full enumeration of all system states and the exact calculation of the corresponding structure function as detailed in Table 2, is not computationally feasible and one has to rely on approximation methods. In fact, the SRLG-RI problem is in #P [24].

Remark 2 (The complexity of the SRLG-RI problem)**.** The SRLG-RI problem belongs to the #P complexity class [24], since this problem is a generalization of k-terminal reliability problem. To see this, assign a unique SRLG to each link. The problem is then simplified to the well-known k-terminal reliability problem, which is in #P [25, 27].

In order to understand the merit of the PMC approach that will be discussed in Section 3, it is instructive to consider a Crude Monte Carlo solution to the SRLG-RI problem.

2.1. The Crude Monte Carlo solution

We proceed with the design of a crude Monte Carlo (CMC) algorithm for the SRLG-RI problem. Let $(G, \mathcal{R}, p)$ be an SRLG-RI instance and let $X_i \sim \mathsf{Bernoulli}(p)$ be a set of random variables that stand for the state of R_i for $1 \le i \le n$. Then, the calculation of the value of the structure function $\psi(X_1, \ldots, X_n)$ can be computed in polynomial time in n, $|V|$, and $|E|$. The latter can be performed by constructing $G(\mathcal{R}') = (V, E')$, a subgraph of $G = (V, E)$, where $\mathcal{R}' = \{R_i \in \mathcal{R} \ : \ X_i = 1\}$ and E' is defined via (1). Finally, the output of the structure function can be calculated by utilizing an algorithm that verifies the resulting subgraph connectivity, for example, using BFS or DFS [37]. Since $\psi(X_1, \ldots, X_n)$ is an indicator function, the system reliability can be written as an expected value via

$$r(G, \mathcal{R}, p) \overset{\text{def}}{=} \mathbb{P}(\psi(X_1, \ldots, X_n) = 1) = \mathbb{E}[\psi(X_1, \ldots, X_n)].$$

This implies the corresponding CMC procedure, which is summarized in Algorithm 1.

Algorithm 1: The CMC algorithm for the estimation of $r(G, \mathcal{R}, p)$

```
Input: $G = (V, E)$, $\mathcal{R} = (R_1, \ldots, R_n)$, $p \in [0, 1]$, and a sample size $N \in \mathbb{N}$
Output: $\hat{\ell}_{\text{CMC}}$ — unbiased estimator of $r(G, \mathcal{R}, p)$
$\hat{\ell}_{\text{CMC}} \leftarrow 0$
for $t \leftarrow 1$ to $N$ do
    Generate $X_i^{(t)} \sim \mathsf{Bernoulli}(p)$ for $1 \le i \le n$.
    $\mathbf{X}_t \leftarrow \left(X_1^{(t)}, \ldots, X_n^{(t)}\right)$
end
return $\widehat{\ell}_{\text{CMC}} \leftarrow \frac{1}{N} \sum_{t=1}^{N} \psi(\mathbf{X}_t)$
```

An estimator for BIM_i for $1 \le i \le n$ can be obtained in a similar fashion via a CMC procedure. Specifically, we can use CMC to estimate $\mathbb{E}[\psi(X_1, \ldots, X_{i-1}, 1, X_{i+1}, \ldots, X_n)]$ and $\mathbb{E}[\psi(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots, X_n)]$ and combine these with (3).

Unfortunately, Algorithm 1 can be very inefficient when dealing with the following scenarios.

1. Suppose that we would like to solve the SRLG-RI problem for m different values of p. In this case, one would require to run m CMC algorithms; for large m, this will be computationally prohibitive.
2. The second problem is concerned with highly reliable networks. In particular, suppose that $1 - p = q \ll 1$. In this case, $\psi(X_1, \ldots, X_n)$ is equal to 1 with high probability, and one cannot estimate the system unreliability in a satisfactory fashion. More formally, note that $\{Y_t = 1 - \psi(\mathbf{X}_t)\}_{t=1}^{N}$ is a set of independent Bernoulli random variable such that $Y_t \sim \mathsf{Bernoulli}(\ell)$, where $\ell = \overline{r}(G, \mathcal{R}, p)$ is the true system unreliability. Then, $\text{Var}\left(\hat{\ell}_{\text{CMC}}\right) = \frac{\ell(1-\ell)}{N}$ holds, and the relative error (RE) [38] of the estimator $\widehat{\ell}_{\text{CMC}}$ is

$$\text{RE}\left(\hat{\ell}_{\text{CMC}}\right) = \frac{\sqrt{\text{Var}\left(\hat{\ell}_{\text{CMC}}\right)}}{\mathbb{E}\left(\hat{\ell}_{\text{CMC}}\right)} = \frac{\sqrt{\ell(1-\ell)/N}}{\ell} \underbrace{\approx}_{\ell \ll 1} 1/\sqrt{N\ell}. \tag{7}$$

   The RE in (7) imposes a serious challenge. To see this, consider the rare-event probability $\ell \approx 10^{-15}$, and suppose that we are interested in a (very) modest 20% RE. It is possible to verify from (7), that the required sample size N is about 2.5×10^{16}.

In order to resolve both problems, we propose to apply the spectra approach, which is detailed in Section 3. We next show that when the failure probability of each SRLG is identical, one can overcome the rare-event problem by using the spectra approach [30]. In addition, under the same failure probability setting, we show that the spectra machinery allows us to identify important BIMs [34], and thus resolve the problem of network IM analysis, too.

3. Methods

To start with, consider an SRLG-RI problem instance $(G, \mathcal{R}, p)$, and let $\mathcal{A}_i = \{\mathcal{R}' \subseteq \mathcal{R} \ : \ |\mathcal{R}'| = i\}$ be the set of all subsets of risk groups of cardinality i for $1 \le i \le n$. In addition, define

$$\mathcal{B}_i = \{\mathcal{R}' \subseteq \mathcal{R} \ : \ |\mathcal{R}'| = i \text{ and } G(\mathcal{R}') \text{ is fully connected}\},$$

to be a subset of $\mathcal{A}_i$. Then, it holds that

$$r(G, \mathcal{R}, p) = \sum_{i=1}^{n} |\mathcal{B}_i| \, p^i \, q^{n-i}, \tag{8}$$

where p and $q = 1 - p$ are the probabilities of an SRLG $R \in \mathcal{R}$ to be in the *up* and the *down* state, respectively. Next, we define an important combinatorial object called the *spectra* [23, 30, 39, 40].

Definition 2 (SRLG spectra)**.** Given an instance of SRLG-RI problem $(G, \mathcal{R}, p)$, where $|\mathcal{R}| = n$, the SRLG spectra is defined by

$$\mathbf{s}(G, \mathcal{R}) = (s_1, \ldots, s_n),$$

where $s_i = |\mathcal{B}_i|/|\mathcal{A}_i|$ for $1 \le i \le n$.

Using the fact that $|\mathcal{A}_i| = \binom{n}{i}$ for $1 \le i \le n$, we can rewrite (8) via

$$r(G, \mathcal{R}, p) = \sum_{i=1}^{n} \binom{n}{i} s_i \, p^i \, q^{n-i}. \tag{9}$$

Moreover, from Definition 2, we conclude that $|\mathcal{A}_i|$ and $|\mathcal{B}_i|$ are independent of the SRLG component failure probability q for $1 \le i \le n$. Therefore, $\mathbf{s}(G, \mathcal{R})$ is a *system invariant*, and thus, provided that $\mathbf{s}(G, \mathcal{R})$ is available, one can calculate the $r(G, \mathcal{R}, p)$ via (9), in $O(n)$ time for any $p \in [0, 1]$. In practice, reliability engineers are more concerned with network's unreliability. With this in mind, it is important to note that the spectra can be used for calculating the network unreliability using $\mathbf{1} - \mathbf{s}(G, \mathcal{R}) = (1 - s_1, \ldots, 1 - s_n)$, since the following holds

$$\bar{r}(G, \mathcal{R}, p) = 1 - r(G, \mathcal{R}, p) = \underbrace{\sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i}}_{\text{binomial theorem} \to (p+q)^n = 1} - \sum_{i=1}^{n} \binom{n}{i} s_i \, p^i q^{n-i} = q^n + \sum_{i=1}^{n} \binom{n}{i} (1 - s_i) \, p^i q^{n-i}.$$

Example 2 (The bridge network reliability via spectra)**.** Consider the bridge network from Example 1 and note that it is possible to verify that $\mathcal{B}_1 = \{\emptyset\}$, $\mathcal{B}_2 = \{\{R_1, R_2\}, \{R_1, R_3\}\}$, and $\mathcal{B}_3 = \{\{R_1, R_2, R_3\}\}$. In addition, since $|\mathcal{A}_1| = \binom{3}{1} = 3$, $|\mathcal{A}_2| = \binom{3}{2} = 3$, $|\mathcal{A}_3| = \binom{3}{3} = 1$, the spectra is equal to

$$\mathbf{s}(G, \mathcal{R}) = \left(\frac{|\mathcal{B}_1|}{|\mathcal{A}_1|}, \frac{|\mathcal{B}_2|}{|\mathcal{A}_2|}, \frac{|\mathcal{B}_3|}{|\mathcal{A}_3|} \right) = \left(0, \frac{2}{3}, 1 \right). \tag{10}$$

By combining (9) and (10), we immediately arrive at the exact network reliability, which is given by

$$r(G, \mathcal{R}, p) = \sum_{i=1}^{3} \binom{n}{i} s_i \, p^i \, q^{n-i} = \binom{3}{1} 0 \, p \, q^2 + \binom{3}{2} \frac{2}{3} \, p^2 q + \binom{3}{3} 1 \, p^3 = 2p^2 q + p^3,$$

and one can verify that this result is equal to the direct reliability calculation from (5).

If we take a careful look at the spectra components in (10), it becomes apparent that estimating the spectra is much easier than estimating the network reliability directly via Algorithm 1, because, these components do not have very small non-zero numbers. In particular, as soon as we obtain a reliable estimator of the spectra, the system reliability or unreliability can be calculated 1) for *any* value of p and 2) regardless of the condition $q \ll 1$; both 1) and 2) were discussed at the end of Section 2. That is, the spectra approach helps to void the need to repeat the CMC algorithm for each p, and addresses the rare-event issue.

The spectra object can be used for calculating the BIM components, too. Let $\mathcal{B}_i^{(j)}$ be a subset of $\mathcal{B}_i$, such that the SRLG R_j is in the *up* state in $\mathcal{B}_i$, namely, define

$$\mathcal{B}_i^{(j)} = \left\{\mathcal{R}' \subseteq \mathcal{R} \ : \ R_j \in \mathcal{R}', \ |\mathcal{R}'| = i \text{ and } G(\mathcal{R}') \text{ is fully connected}\right\}.$$

Similar to the SRLG spectra from Definition 2, we define the BIM spectra as follows.

Definition 3 (SRLG BIM spectra of component j for $1 \leq j \leq n$)**.** Given an instance of SRLG-RI problem $(G, \mathcal{R}, p)$, where $|\mathcal{R}| = n$, the SRLG-BIM spectra for component j is defined by

$$\mathbf{s}^{(\text{bim}_j)}(G, \mathcal{R}) = \left(s_1^{(\text{bim}_j)}, \dots, s_n^{(\text{bim}_j)}\right),$$

where $s_i^{(\text{bim}_j)} = |\mathcal{B}_i^{(j)}|/|\mathcal{A}_i|$ for $1 \leq i \leq n$.

Combining the SRLG spectra and the SRLG BIM spectra from Definition 2 and Definition 3, one can obtain the set of BIM components. The corresponding result is summarized in Lemma 1 [22].

Lemma 1 (BIM calculation using spectra [22])**.** *Let* $\mathbf{s}(G, \mathcal{R}) = (s_1, \dots, s_n)$ *and* $\mathbf{s}^{(\text{bim}_j)}(G, \mathcal{R}) = \left(s_1^{(\text{bim}_j)}, \dots, s_n^{(\text{bim}_j)}\right)$ *be the SRLG spectra and the SRLG BIM spectra of component j for* $1 \leq j \leq n$*. Then, it holds that*

$$\text{BIM}_j = \sum_{i=1}^{n} \binom{n}{i} s_i^{(\text{bim}_j)} p^{i-1} q^{n-i} - \binom{n}{i} (s_i - s_i^{(\text{bim}_j)}) p^i q^{n-i-1}. \tag{11}$$

Proof. First, note that

$$\Psi(p_1, \dots, p_{j-1}, 1, p_{j+1}, \dots, p_n) = \sum_{i=1}^{n} |\mathcal{B}_i^{(j)}| \, p^{i-1} \, q^{n-i}, \tag{12}$$

where we have p^{i-1} instead of p^i, since the jth component is always in the *up* state. Let $\mathcal{B}_i^{(\neg j)}$ be a subset of $\mathcal{B}_i$, such that the SRLG R_j is in the *down* state in $\mathcal{B}_i$, namely, $\mathcal{B}_i^{(\neg j)} = \left\{\mathcal{R}' \subseteq \mathcal{R} \ : \ R_j \notin \mathcal{R}', \ |\mathcal{R}'| = i \text{ and } G(\mathcal{R}') \text{ is fully connected}\right\}$. Then, it holds that

$$\Psi(p_1, \dots, p_{j-1}, 0, p_{j+1}, \dots, p_n) = \sum_{i=1}^{n} |\mathcal{B}_i^{(\neg j)}| \, p^i \, q^{n-i-1}, \tag{13}$$

where we have q^{n-i-1} instead of q^{n-i}, since the jth component is always in the *down* state. Finally, since $\mathcal{B}_i^{(j)}$ and $\mathcal{B}_i^{(\neg j)}$ is a partition of $\mathcal{B}_i$, we have that $\left|\mathcal{B}_i^{(\neg j)}\right| = |\mathcal{B}_i| - \left|\mathcal{B}_i^{(j)}\right|$, and by combining (12) and (13) with the definition of BIM_j in (3), we arrive at

$$\begin{aligned}
\text{BIM}_j &\overset{\text{def}}{=} \Psi(p_1, \dots, p_{j-1}, 1, p_{j+1}, \dots, p_n) - \Psi(p_1, \dots, p_{j-1}, 0, p_{j+1}, \dots, p_n) \\
&= \sum_{i=1}^{n} |\mathcal{B}_i^{(j)}| \, p^{i-1} \, q^{n-i} - \sum_{i=1}^{n} |\mathcal{B}_i^{(\neg j)}| \, p^i \, q^{n-i-1} = \sum_{i=1}^{n} |\mathcal{B}_i^{(j)}| \, p^{i-1} \, q^{n-i} - (|\mathcal{B}_i| - |\mathcal{B}_i^{(j)}|) p^i q^{n-i-1} \\
&= \sum_{i=1}^{n} \binom{n}{i} s_i^{(\text{bim}_j)} p^{i-1} q^{n-i} - \binom{n}{i} (s_i - s_i^{(\text{bim}_j)}) p^i q^{n-i-1},
\end{aligned}$$

and thus, (11) follows. □

Example 3 (The bridge network BIMs calculation with spectra)**.** For the bridge example, one can verify that

$$\left|\mathcal{B}_1^{(1)}\right| = 0, \quad \left|\mathcal{B}_2^{(1)}\right| = 2, \quad \left|\mathcal{B}_3^{(1)}\right| = 1,$$

and that

$$\left|\mathcal{B}_1^{(2)}\right| = \left|\mathcal{B}_1^{(3)}\right| = 0, \quad \left|\mathcal{B}_2^{(2)}\right| = \left|\mathcal{B}_2^{(3)}\right| = 1, \quad \left|\mathcal{B}_3^{(2)}\right| = \left|\mathcal{B}_3^{(3)}\right| = 1.$$

We can now calculate all BIM components via (11) as follows

$$\text{BIM}_1 = \sum_{i=1}^{3} \left|\mathcal{B}_i^{(1)}\right| p^{i-1}q^{n-i} - (|\mathcal{B}_i| - \left|\mathcal{B}_i^{(1)}\right|)p^i q^{n-i-1} = \underbrace{2pq - 0}_{i=2} + \underbrace{p^2 - 0}_{i=3} = 2pq + p^2,$$

$$\text{BIM}_2 = \text{BIM}_3 = \sum_{i=1}^{3} \left|\mathcal{B}_i^{(2)}\right| p^{i-1}q^{n-i} - (|\mathcal{B}_i| - \left|\mathcal{B}_i^{(2)}\right|)p^i q^{n-i-1} = \underbrace{pq - p^2}_{i=2} + \underbrace{p^2 - 0}_{i=3} = pq.$$

As expected, the above BIM calculation via spectra, agrees with the direct calculation in (6).

Unfortunately, for a general network, and in particular for larger networks, the spectra cannot be obtained analytically. Therefore, for the rest of this section we show how one can estimate the spectra components, namely, how to estimate both the $\mathbf{s}(G, \mathcal{R})$ and the $\mathbf{s}^{(\text{bim}_j)}(G, \mathcal{R})$ for $1 \leq j \leq n$.

Remark 3 (Estimating spectra via CMC)**.** It is possible to design a simple CMC algorithm for estimating both spectra objects. Specifically, an estimator for the s_i and for the $s_i^{(\text{bim}_j)}$ for some fixed $i, j \in \{1, \ldots, n\}$, can be obtained by defining the uniform distribution over the $\mathcal{A}_i$ set, and estimating the expected values of the random variables

$$\mathbb{1}_{\{\mathcal{A}' \in \mathcal{A}_i \text{ and } G(\mathcal{A}') \text{ is fully connected}\}} \quad \text{and} \quad \mathbb{1}_{\{\mathcal{A}' \in \mathcal{A}_i \text{ and } G(\mathcal{A}') \text{ is fully connected and } R_j \in \mathcal{A}'\}},$$

since it holds that

$$\mathbb{E}\left[\mathbb{1}_{\{\mathcal{A}' \in \mathcal{A}_i \text{ and } G(\mathcal{A}') \text{ is fully connected}\}}\right] = \frac{|\mathcal{B}_i|}{|\mathcal{A}_i|} \overset{\text{def}}{=} s_i, \quad \text{and} \quad \mathbb{E}\left[\mathbb{1}_{\{\mathcal{A}' \in \mathcal{A}_i \text{ and } G(\mathcal{A}') \text{ is fully connected and } R_j \in \mathcal{A}'\}}\right] = \frac{\left|\mathcal{B}_i^{(j)}\right|}{|\mathcal{A}_i|} \overset{\text{def}}{=} s_i^{(\text{bim}_j)}.$$

However, the cost of such CMC algorithm can be computationally prohibitive, since the CMC approach will require an execution of n CMC algorithms for estimating the SRLG spectra. Moreover, if we wish to estimate $s_i^{(\text{bim}_j)}$ separately for all $i, j \in \{1, \ldots, n\}$, we will additionally require n^2 CMC algorithm executions. Instead, similar to Gertsbakh et. al. [30], we propose to apply the PMC approach, which will considerably reduce the required computational effort and consequently open a way to handle real-sized networks.

3.1. The PMC method

The PMC method, which is summarized in Algorithm 2, has the advantage that both spectra objects are estimated in one run, and in contrast to the CMC approach, there is no rejection sampling involved. We start with defining the uniform distribution on the set of all permutations of the $\{1, \ldots, n\}$ set. Let $\mathbf{\Pi} = (\Pi_1, \ldots, \Pi_n)$ be a random permutation, and define S_i and $S_i^{(j)}$ to be indicator random variables for $1 \leq i, j \leq n$ such as

$$S_i = \begin{cases} 1 & \text{if } G(\{R_{\Pi_1}, \ldots, R_{\Pi_i}\}) \text{ is fully connected,} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$S_i^{(j)} = \begin{cases} 1 & \text{if } G(\{R_{\Pi_1}, \ldots, R_{\Pi_i}\}) \text{ is fully connected, and } j \in \{\Pi_1, \ldots, \Pi_i\}, \\ 0 & \text{otherwise.} \end{cases}$$

Noting that for every connected subgraph $G' \in \mathcal{B}_i$, there exist $i!(n-i)!$ corresponding SRLG permutations, and combining this with the fact that there are $n!$ possible permutations over the $\{1, \ldots, n\}$ set, we arrive at

$$\mathbb{E}[S_i] = \sum_{\boldsymbol{\pi}} S_i \mathbb{P}(\boldsymbol{\pi}) = \frac{1}{n!} \sum_{\boldsymbol{\pi}} S_i = \frac{1}{n!} |\mathcal{B}_i| i!(n-i)! = |\mathcal{B}_i| \binom{n}{i}^{-1} = \frac{|\mathcal{B}_i|}{|\mathcal{A}_i|} \overset{\text{def}}{=} s_i, \tag{14}$$

and

$$\mathbb{E}\left[S_i^{(j)}\right] = \sum_{\pi} S_i^{(j)}\mathbb{P}(\boldsymbol{\pi}) = \frac{1}{n!}\sum_{\pi} S_i^{(j)} = \frac{1}{n!}\left|\mathcal{B}_i^{(j)}\right| i!(n-i)! = \left|\mathcal{B}_i^{(j)}\right|\binom{n}{i}^{-1} = \frac{\left|\mathcal{B}_i^{(j)}\right|}{|\mathcal{A}_i|} \overset{\text{def}}{=} s_i^{(\text{bim}_j)}. \tag{15}$$

The basic idea of the PMC algorithm is a straightforward generalization of the procedure from Gertsbakh et. al. [23]. The SRLG PMC method is summarized in Algorithm 2, and the resulting spectra objects unbiasedness follows from (14) and (15).

```
Algorithm 2: The PMC algorithm for the estimation of $\mathbf{s}(G,\mathcal{R})$ and of $\mathbf{s}^{(\text{bim}_j)}(G,\mathcal{R})$ for all $1 \le j \le n$
   Input: $G = (V,E)$, $\mathcal{R} = (R_1,\dots,R_n)$, and a sample size $N \in \mathbb{N}$
   Output: $\widehat{\mathbf{s}}(G,\mathcal{R}) = (\widehat{s}_1,\dots,\widehat{s}_n)$ — unbiased estimator of the spectra object $\mathbf{s}(G,\mathcal{R})$, and
           $\widehat{\mathbf{s}}^{(\text{bim}_j)}(G,\mathcal{R}) = \left(\widehat{s}_1^{(\text{bim}_j)},\dots,\widehat{s}_n^{(\text{bim}_j)}\right)$ — unbiased estimator of the BIM spectra object for all $1 \le j \le n$
   /* initialization                                                          */
 1 for $i \leftarrow 1$ to $n$ do
 2     $\widehat{s}_i \leftarrow 0$
 3     for $j \leftarrow 1$ to $n$ do
 4         $\widehat{s}_i^{(\text{bim}_j)} \leftarrow 0$
 5     end
 6 end
   /* The PMC main loop                                                       */
 7 for $t \leftarrow 1$ to $N$ do
 8     Generate a permutation $\mathbf{\Pi} = (\Pi_1,\dots,\Pi_n)$ uniformly at random.
 9     $\mathcal{R}' \leftarrow \emptyset$
10     for $i \leftarrow 1$ to $n$ do
11         $\mathcal{R}' \leftarrow \mathcal{R}' \cup R_{\Pi_i}$
12         if $G(\mathcal{R}')$ is fully connected then
13             $\widehat{s}_i \leftarrow \widehat{s}_i + 1$
14             for $j \leftarrow 1$ to $i$ do
15                 $\widehat{s}_i^{(\text{bim}_{\Pi_j})} \leftarrow \widehat{s}_i^{(\text{bim}_{\Pi_j})} + 1$
16             end
17         end
18     end
19 end
20 $\widehat{\mathbf{s}}(G,\mathcal{R}) \leftarrow \frac{1}{N}(\widehat{s}_1,\dots,\widehat{s}_n)$
21 for $j \leftarrow 1$ to $n$ do
22     $\widehat{\mathbf{s}}^{(\text{bim}_j)}(G,\mathcal{R}) \leftarrow \frac{1}{N}\left(\widehat{s}_1^{(\text{bim}_j)},\dots,\widehat{s}_n^{(\text{bim}_j)}\right)$
23 end
24 return $\widehat{\mathbf{s}}(G,\mathcal{R})$, and $\widehat{\mathbf{s}}^{(\text{bim}_j)}(G,\mathcal{R})$ for all $1 \le j \le n$
```

Algorithm 2 requires $O(n^2)$ space to maintain the spectra objects. The initialization step in lines 1-6 takes $O(n^2)$ time, (but can be performed in $O(1)$ time by allocating a memory block initialized with zeros via the `calloc` instruction). Similarly, line 20 and lines 21-23 can be performed in $O(n)$ and $O(n^2)$ time, respectively. We next proceed with the main PMC loop. If we ignore the connectivity verification in line 12, the loop in lines 10-18 runs in $O(n^2)$ time; to see this, note that each $\widehat{s}_i^{(\text{bim}_j)}$ for $1 \le i, j \le n$ is updated at most once during the execution of lines 10-18.

In order to check the connectivity in line 12, we use the *disjoint-set* data structure which allows us to track the number of connected components in a graph while adding its links. In each iteration of the main PMC loop, we will need to spend $O(|V| + |E|\,\alpha(|E|,|V|))$ time to make $|E|$ disjoint set operations; here, $\alpha(\cdot,\cdot)$ is the inverse Ackermann function [41, 42]. Under our failure regime setting, for each edge, we are also required to track the number of SRLGs that enter the *up* state. This means that when line 11 ($\mathcal{R}' \leftarrow \mathcal{R}' \cup R_{\Pi_i}$), is executed, we traverse over all edges in R_{Π_i} and check if these edges can be added to the subgraph. Therefore, the overall complexity of the loop in lines 10-18 is $\mathcal{O}(n^2) + \mathcal{O}(\sum_{i=1}^{n} |R_i|) + \mathcal{O}(|V| + |E|\,\alpha(|E|, |V|)) = \mathcal{O}(n^2 + n|E| + |V| + |E|\alpha(|E|, |V|))$. This also governs the algorithm execution runtime and thus the overall runtime complexity is $\mathcal{O}(N \times (n^2 + |V| + |E|(n + \alpha(|E|, |V|))))$.

As soon as the $\widehat{\mathbf{s}}(G, \mathcal{R})$, and the $\widehat{\mathbf{s}}^{(\text{bim}_j)}(G, \mathcal{R})$ for all $1 \le j \le n$ objects are available, we can use (9) and (11) to obtain the following estimators of the system reliability and IM.

1. The reliability estimator

$$\widehat{r}(G, \mathcal{R}, p) = \sum_{i=1}^{n} \binom{n}{i} \widehat{s_i}\, p^i\, q^{n-i}. \tag{16}$$

2. The BIM estimators for $1 \le j \le n$

$$\widehat{\text{BIM}}_j = \sum_{i=1}^{n} \binom{n}{i} \widehat{s_i^{(\text{bim}_j)}} p^{i-1} q^{n-i} - \binom{n}{i} (\widehat{s_i} - \widehat{s_i^{(\text{bim}_j)}}) p^i q^{n-i-1}. \tag{17}$$

Example 4 (PMC algorithm for the bridge network)**.** Table 3 summarizes all possible permutations over the bridge network SRLGs. Ones and zeros stand for the corresponding subgraph's connectivity and disconnectedness, respectively. For example, consider the S_1 entry which is always zero; this means that a failure of two SRLGs will always lead to network disconnection. Careful consideration of all SRLG permutations in the table shows that there is a possibility to connect the network using two groups R_1 and R_2 or R_1 and R_3. However, the pair of groups R_2 and R_3 in the *up* state, is not sufficient for the network connectivity.

Table 3 Permutations of SRLGs of the bridge network.

π	S_1	S_2	S_3	$S_1^{(1)}$	$S_2^{(1)}$	$S_3^{(1)}$	$S_1^{(2)}$	$S_2^{(2)}$	$S_3^{(2)}$	$S_1^{(3)}$	$S_2^{(3)}$	$S_3^{(3)}$
$\pi = (1, 2, 3)$	0	1	1	0	1	1	0	1	1	0	0	1
$\pi = (1, 3, 2)$	0	1	1	0	1	1	0	0	1	0	1	1
$\pi = (2, 1, 3)$	0	1	1	0	1	1	0	1	1	0	0	1
$\pi = (2, 3, 1)$	0	0	1	0	0	1	0	0	1	0	0	1
$\pi = (3, 1, 2)$	0	1	1	0	1	1	0	0	1	0	1	1
$\pi = (3, 2, 1)$	0	0	1	0	0	1	0	0	1	0	0	1

Because there are only 6 possible permutations, it is possible to (exactly) derive the spectra objects from Table 3. In particular, we have that $\mathbf{s}(G, \mathcal{R}) = (0/6,\ 4/6,\ 6/6) = (0,\ 2/3,\ 1)$, (note that $\mathbf{s}(G, \mathcal{R})$ is equal to the spectra object in (10)), $\mathbf{s}^{(\text{bim}_1)}(G, \mathcal{R}) = (0, 4/6, 6/6) = (0, 2/3, 1)$, and $\mathbf{s}^{(\text{bim}_2)}(G, \mathcal{R}) = \mathbf{s}^{(\text{bim}_3)}(G, \mathcal{R}) = (0, 2/6, 6/6) = (0, 1/3, 1)$. Using the fact that $s_i = |\mathcal{B}_i|/|\mathcal{A}_i|$ and that $s_i^{(\text{bim}_j)} = |\mathcal{B}_i^{(j)}|/|\mathcal{A}_i|$, we arrive at

$$|\mathcal{B}_1| = \binom{3}{1} s_1 = 0,\ |\mathcal{B}_2| = \binom{3}{2} s_2 = 2,\ |\mathcal{B}_2| = \binom{3}{3} s_3 = 1,\ \left|\mathcal{B}_1^{(1)}\right| = \binom{3}{1} s_1^{(\text{bim}_1)} = 0,\ \left|\mathcal{B}_2^{(1)}\right| = \binom{3}{2} s_2^{(\text{bim}_1)} = 2,$$

$$\left|\mathcal{B}_3^{(1)}\right| = \binom{3}{3} s_3^{(\text{bim}_1)} = 1,\ \left|\mathcal{B}_1^{(2)}\right| = \left|\mathcal{B}_1^{(3)}\right| = \binom{3}{1} s_1^{(\text{bim}_2)} = 0,\ \left|\mathcal{B}_2^{(2)}\right| = \left|\mathcal{B}_2^{(3)}\right| = \binom{3}{2} s_2^{(\text{bim}_2)} = 1,\ \left|\mathcal{B}_3^{(2)}\right| = \left|\mathcal{B}_3^{(3)}\right| = \binom{3}{3} s_3^{(\text{bim}_2)} = 1.$$

By comparing the cardinalities of $\left|\mathcal{B}_i^{(j)}\right|$ for $1 \le i, j \le 3$ with the cardinalities in Example 3, one can verify that the PMC approach provides the same results.

When applying the PMC algorithm in practice, one should take into account two technical issues, that are discussed next.

3.2. Reliability estimation in practice

While we saw that Algorithm 2 provides unbiased estimators for reliability and IM, we should also discuss the limitation of this method. Intuitively, Algorithm 2 will be very accurate provided that the spectra components (s_i for $1 \leq i \leq n$), are not very small. In this case, Algorithm 2 can estimate these efficiently. Otherwise, one should resort to different approximation techniques such as importance sampling or multilevel splitting [38]. More formally, the efficiency of Algorithm 2 for the estimation of the $\mathbf{s}(G,\mathcal{R})$ object and the corresponding reliability calculation, is discussed in Theorem 1. In particular, Theorem 1 establishes the required conditions for the PMC Algorithm 2 to be a fully polynomial randomized approximation scheme (FPRAS) for the reliability problem. Obtaining such a result is very desirable for any #P problem. One can prove that a randomized algorithm is an FPRAS, if its output's coefficient of variation (CV) is bounded by a polynomial in the algorithm input size [43].

Theorem 1 (PMC efficiency)**.** Let $(G,\mathcal{R},p)$ be an SRLG-RI instance where $|\mathcal{R}| = n$ and let $\widehat{\mathbf{s}}(G,\mathcal{R}) = (\widehat{s_1},\dots,\widehat{s_n})$ be the spectra estimator of $\mathbf{s}(G,\mathcal{R}) = (s_1,\dots,s_n)$ obtained via PMC Algorithm 2. Finally, let $\widehat{r}(G,\mathcal{R},p)$ be the reliability estimator that was calculated using (16), for any $p \in [0,1]$. Then, provided that the minimal non-zero component of $\mathbf{s}(G,\mathcal{R})$ is at least equal to $1/\mathcal{P}(n)$, where $\mathcal{P}(n)$ is a polynomial in n, it holds that

$$\mathrm{CV} \overset{\text{def}}{=} \frac{\mathbb{E}\left[\widehat{r}(G,\mathcal{R},p)^2\right]}{\mathbb{E}\left[\widehat{r}(G,\mathcal{R},p)\right]^2} \leq n\mathcal{P}(n)^2.$$

Proof. The proof is by bounding of the first and the second moment of the random variable $\widehat{r}(G,\mathcal{R},p)$. In particular, by combining (14) and (16), we arrive at

$$\mathbb{E}\left[\widehat{r}(G,\mathcal{R},p)\right] = \mathbb{E}\left[\sum_{i=1}^{n}\binom{n}{i}S_i\,p^i\,q^{n-i}\right] = \sum_{i=1}^{n}\binom{n}{i}\mathbb{E}[S_i]\,p^i\,q^{n-i} \underset{(14)}{=} \sum_{i=1}^{n}\binom{n}{i}s_i\,p^i\,q^{n-i} = r(G,\mathcal{R},p). \tag{18}$$

Next, we consider an upper bound for the second moment of $\widehat{r}(G,\mathcal{R},p)$. Specifically, recall that $S_i \sim \mathsf{Bernoulli}(s_i)$, so $\mathbb{E}\left[S_i^2\right] = s_i \leq 1$ for $1 \leq i \leq n$. Then, it holds that

$$\begin{aligned}
\mathbb{E}\left[\widehat{r}(G,\mathcal{R},p)^2\right] &= \mathbb{E}\left[\left(\sum_{i=1}^{n}\binom{n}{i}S_i\,p^i\,q^{n-i}\right)^2\right] \underset{\text{(Jensen inequality [43])}}{\leq} \mathbb{E}\left[n\sum_{i=1}^{n}\binom{n}{i}^2 S_i^2\,p^{2i}\,q^{2(n-i)}\right] \\
&= n\sum_{i=1}^{n}\binom{n}{i}^2\mathbb{E}\left[S_i^2\right]\,p^{2i}\,q^{2(n-i)} \underset{(\mathbb{E}[S_i^2]\leq 1)}{\leq} n\sum_{i=1}^{n}\binom{n}{i}^2 1\,p^{2i}\cdot q^{2(n-i)} = n\left(\sum_{i=1}^{n}\binom{n}{i}^2\,p^{2i}\,q^{2(n-i)}\right),
\end{aligned} \tag{19}$$

Finally, let $\underline{s}$ be the minimal *non zero* component of $\mathbf{s}(G,\mathcal{R})$, and combine (18) and (19) to obtain

$$\begin{aligned}
\mathrm{CV} &= \frac{\mathbb{E}\left[\widehat{r}(G,\mathcal{R},p)^2\right]}{\mathbb{E}\left[\widehat{r}(G,\mathcal{R},p)\right]^2} \leq \frac{n\left(\sum_{i=1}^{n}\binom{n}{i}^2 p^{2i}\,q^{2(n-i)}\right)}{\left(\sum_{i=1}^{n}\binom{n}{i}s_i\,p^i\,q^{n-i}\right)^2} \leq \frac{n\left(\sum_{i=1}^{n}\binom{n}{i}^2 p^{2i}\,q^{2(n-i)}\right)}{\left(\sum_{i=1}^{n}\binom{n}{i}\underline{s}\,p^i\,q^{n-i}\right)^2} \leq \frac{n\left(\sum_{i=1}^{n}\binom{n}{i}^2 p^{2i}\,q^{2(n-i)}\right)}{\underline{s}^2\left(\sum_{i=1}^{n}\binom{n}{i}p^i\,q^{n-i}\right)^2} \\
&\leq \frac{n\left(\sum_{i=1}^{n}\binom{n}{i}^2 p^{2i}\,q^{2(n-i)}\right)}{\underline{s}^2\left(\sum_{i=1}^{n}\binom{n}{i}^2 p^{2i}\,q^{2(n-i)}\right)} \leq \frac{n}{\underline{s}^2} \underset{(\underline{s}\geq 1/\mathcal{P}(n))}{\leq} n\mathcal{P}(n)^2,
\end{aligned}$$

and thus, to complete the proof. □

From the practical point of view, the rare-event issue can be resolved either by increasing the sample size N in Algorithm 2 (which is not generally practical), or by applying more advanced variance minimization techniques. Please see Vaisman et. al. [35] for details.

3.3. BIM estimation in practice

Our numerical evaluation in Section 4 implies that the reliability estimator (16), is very useful in practice and that its RE shows good stability. However, we found that the BIM estimator (17) can be very unstable in the sense of its RE, and therefore, it is not advisable to apply it directly. Instead, we propose to exploit the following important property of the BIM spectra [22], which is summarized in Theorem 2. While this result is due to Gertsbakh [22], we failed to find a formal proof to the second part of Theorem 2, so we provide the full proof next.

Theorem 2 (Gertsbakh [22])**.** *Let* $\mathbf{s}^{(\mathrm{bim}_j)}(G, \mathcal{R}) = \left(s_1^{(\mathrm{bim}_j)}, \ldots, s_n^{(\mathrm{bim}_j)}\right)$ *be a BIM spectra. Then, for some fixed* j_1 *and* j_2 *such that* $1 \le j_1 \ne j_2 \le n$*, the following holds.*

1. *If for all* $1 \le i \le n$*,* $s_i^{(\mathrm{bim}_{j_1})} \ge s_i^{(\mathrm{bim}_{j_2})}$*, then we have that* $\mathrm{BIM}_{j_1} \ge \mathrm{BIM}_{j_2}$*.*
2. *Suppose that* 1*. does not hold, and let* $1 \le k \le n$*, be the maximal index such that* $s_k^{(\mathrm{bim}_{j_1})} \ne s_k^{(\mathrm{bim}_{j_2})}$*. Without loss of generality, suppose that* $s_k^{(\mathrm{bim}_{j_1})} > s_k^{(\mathrm{bim}_{j_2})}$ *holds. Then, there exists* $p_0 \in [0, 1]$*, such that* $\mathrm{BIM}_{j_1} > \mathrm{BIM}_{j_2}$ *for all* $p > p_0$*.*

Proof. In order to prove the first part, note that

$$\begin{aligned}
\mathrm{BIM}_{j_1} - \mathrm{BIM}_{j_2} &= \sum_{i=1}^{n} \binom{n}{i} s_i^{(\mathrm{bim}_{j_1})} p^{i-1} q^{n-i} - \binom{n}{i}\left(s_i - s_i^{(\mathrm{bim}_{j_1})}\right) p^i q^{n-i-1} \\
&\quad - \sum_{i=1}^{n} \binom{n}{i} s_i^{(\mathrm{bim}_{j_2})} p^{i-1} q^{n-i} - \binom{n}{i}\left(s_i - s_i^{(\mathrm{bim}_{j_2})}\right) p^i q^{n-i-1} \\
&= \sum_{i=1}^{n} \binom{n}{i}\left(s_i^{(\mathrm{bim}_{j_1})} - s_i^{(\mathrm{bim}_{j_2})}\right) p^{i-1} q^{n-i} - \binom{n}{i}\left(s_i^{(\mathrm{bim}_{j_2})} - s_i^{(\mathrm{bim}_{j_1})}\right) p^i q^{n-i-1} \\
&= \sum_{i=1}^{n} \binom{n}{i}\left(q s_i^{(\mathrm{bim}_{j_1})} - q s_i^{(\mathrm{bim}_{j_2})}\right) p^{i-1} q^{n-i-1} - \binom{n}{i}\left(p s_i^{(\mathrm{bim}_{j_2})} - p s_i^{(\mathrm{bim}_{j_1})}\right) p^{i-1} q^{n-i-1} \\
&= \sum_{i=1}^{n} \binom{n}{i}\left(s_i^{(\mathrm{bim}_{j_1})} - s_i^{(\mathrm{bim}_{j_2})}\right) p^{i-1} q^{n-i-1},
\end{aligned}$$

holds. Therefore, provided that $s_i^{(\mathrm{bim}_{j_1})} \ge s_i^{(\mathrm{bim}_{j_2})}$ for all $1 \le i \le n$, we arrive at

$$\sum_{i=1}^{n} \binom{n}{i} \left(\underbrace{s_i^{(\mathrm{bim}_{j_1})} - s_i^{(\mathrm{bim}_{j_2})}}_{\ge 0} \right) p^{i-1} q^{n-i-1} = \mathrm{BIM}_{j_1} - \mathrm{BIM}_{j_2} \ge 0 \Rightarrow \mathrm{BIM}_{j_1} \ge \mathrm{BIM}_{j_2}.$$

For the second part, assume that there exists $1 \le k \le n-1$, where k is the maximal index such that $s_k^{(\mathrm{bim}_{j_1})} > s_k^{(\mathrm{bim}_{j_2})}$. Let $\Delta = s_k^{(\mathrm{bim}_{j_1})} - s_k^{(\mathrm{bim}_{j_2})}$ (note that $\Delta > 0$), and consider the expression $\mathrm{BIM}_{j_1} - \mathrm{BIM}_{j_2}$. Then, it holds that

$$\begin{aligned}
\mathrm{BIM}_{j_1} - \mathrm{BIM}_{j_2} &= \sum_{i=1}^{k} \binom{n}{i}\left(s_i^{(\mathrm{bim}_{j_1})} - s_i^{(\mathrm{bim}_{j_2})}\right) p^{i-1} q^{n-i-1} + \underbrace{\sum_{i=k+1}^{n} \binom{n}{i}\left(s_i^{(\mathrm{bim}_{j_1})} - s_i^{(\mathrm{bim}_{j_2})}\right) p^{i-1} q^{n-i-1}}_{=0} \\
&= \sum_{i=1}^{k-1} \binom{n}{i}\left(s_i^{(\mathrm{bim}_{j_1})} - s_i^{(\mathrm{bim}_{j_2})}\right) p^{i-1} q^{n-i-1} + \binom{n}{k} \Delta p^{k-1} q^{n-k-1} \\
&\underbrace{\ge}_{s_i^{(\mathrm{bim}_{j_1})} - s_i^{(\mathrm{bim}_{j_2})} \ge -1} \binom{n}{k} \Delta p^{k-1} q^{n-k-1} - \sum_{i=1}^{k-1} \binom{n}{i} p^{i-1} q^{n-i-1} \ge \Delta p^{k-1} q^{n-k-1} - \sum_{i=0}^{k-1} \binom{n}{i} p^{i-1} q^{n-i-1} \\
&\ge \Delta p^{k-1} q^{n-k-1} - n^n \sum_{i=0}^{k-1} \binom{k-1}{i} p^{i-1} q^{n-i-1} = \Delta p^{k-1} q^{n-k-1} - \frac{n^n q^{n-k}}{p} \underbrace{\sum_{i=0}^{k-1} \binom{k-1}{i} p^i q^{k-1-i}}_{=1} = \Delta p^{k-1} q^{n-k-1} - \frac{n^n q^{n-k}}{p}.
\end{aligned}$$

Therefore, in order to show that there exists p_0, such that for any $p > p_0$, $\text{BIM}_{j_1} > \text{BIM}_{j_2}$, it is sufficient to see that there exists p_0, such that for any $p > p_0$, $\Delta p^{k-1}q^{n-k-1} - \frac{n^n q^{n-k}}{p} > 0$. Now, it holds that

$$\Delta p^{k-1}q^{n-k-1} - \frac{n^n q^{n-k}}{p} > 0 \iff \frac{\Delta p^k q^{n-k}}{q} > n^n q^{n-k} \iff \frac{p^k}{q} > \frac{n^n}{\Delta} \iff \frac{p^k}{1-p} > \frac{n^n}{\Delta}.$$

By noting that n^n and Δ are constants, and combining this with the fact that $\lim_{p\to 1^-} \frac{p^k}{1-p} = \infty$, the theorem follows. □

It is important to note that Theorem 2 is not directly applicable, since we only have access to *estimators* of BIM spectra components. However, Corollary 1 can be utilized to deliver statistically significant results regarding the importance of BIM components.

Corollary 1. *Suppose that* $s_{n-1}^{(\text{bim}_{j_1})} > s_{n-1}^{(\text{bim}_{j_2})}$ *holds. Then, there exists* $p_0 \in [0,1]$, *such that* $\text{BIM}_{j_1} > \text{BIM}_{j_2}$ *for all* $p > p_0$.

Proof. This corollary is an immediate consequence of part 2 of Theorem 2, since $s_n^{(\text{bim}_j)} = 1$ for all $1 \le j \le n$. □

Statistically significant results regarding the importance of BIM components can be achieved for example, by obtaining confidence intervals on $\widehat{s}_{n-1}^{(\text{bim}_j)}$ for all $1 \le j \le n$. In addition, ANOVA-like testing machinery is available to compare these $\widehat{s}_{n-1}^{(\text{bim}_j)}$s. This technique allows us to avoid the stability problems in (17), and our numerical results show that this approach is both accurate and useful.

4. Experimental study

In this section, we focus on the performance evaluation of the proposed PMC method. For each model under consideration, we apply the PMC algorithm and estimate both the network reliability and the IM for a range of SRLG failure probabilities. Our experimental study shows that the PMC paradigm is very effective as compared to CMC, and that it is scalable in practice in the sense that the algorithm is capable of dealing with real-sized network instances. Specifically, we consider the following experiments.

1. Our first case study is the bridge network from Section 2. The availability of the corresponding analytical solution, allows us to benchmark the accuracy of the PMC algorithm. In addition, we show the necessity of applying Corollary 1 for an efficient estimation of important BIM components.
2. For the second case study, we consider a very structured network called the *wheel* graph. For this particular topology, one can construct different sized networks and benchmark the algorithmic performance using the available analytical solution. In order to test the performance of the proposed method when applied to larger real-life instances, we explore wheel graphs with several hundreds of vertices, edges, and SRLGs.
3. In the third case study, we demonstrate the performance of Algorithm 2 when applied to a simulated internet graph [44]. While analytical results are no longer available, we construct the set of SRLGs in such a way, that highest valued BIM components will be known in advance. In addition, the purpose of this case study is to show that the proposed method is capable of dealing with different failure regimes as discussed in Remark 1.
4. The forth case study shows that the proposed method is suitable for handling a large real-life instance. In particular, we test the method performance on the 20 US-based service providers network. This data was collected as part of the *Internet Atlas Project* [45, 46].
5. Finally, we consider the PMC algorithm performance when applied to the 20 US-based service providers network while using a different network failure regime. This case study reveals a possible limitation of the PMC algorithm, by demonstrating a scenario for which the method can fail.

The experimental setup: We implemented the PMC algorithm in `C++` package called *SrlgRI*. The software is freely available on the author's website under `https://people.smp.uq.edu.au/RadislavVaisman/#software`. The package was compiled using `GNU g++` with full optimization for speed (utilizing the `-O3` flag). We instrumented the timing measure directly into the code and the provided research software is single-threaded. However, parallelization of Algorithm 2 would be relatively easy to include. All tests were executed on Intel Core i7-6920HQ CPU 2.90GHz processor with 32GB of RAM running 64 bit Ubuntu 20.04 LTS. The PMC algorithm requires a single input parameter N — the sample size. The sample size ranges from $N = 10^3$ to $N = 10^6$ and depends on the number of SRLG components under consideration. In addition, for each case study, we perform N_{sim} independent experiments (independent runs of Algorithm 2), in order to report the corresponding estimated RE, and confidence intervals when applicable. With a view to ensuring reproducibility, each test case was executed using a sequence of fixed seeds: $(1001, 1002, \ldots, 1000 + N_{\text{sim}})$.

Having in mind that the output of Algorithm 2 is random, we use standard accuracy measures for a simulated experiment that outputs a random variable W. Specifically, let W be a random variable which stands for the output of a stochastic experiment and let $W_1, \ldots, W_{N_{\text{sim}}}$ be independent realizations of W. Let us consider the sample mean $\widehat{\ell}$ such that $\widehat{\ell} = N_{\text{sim}}^{-1} \sum_{t=1}^{N_{\text{sim}}} W_t$, and the sample variance $\widehat{\sigma}^2$, which is given by $\widehat{\sigma}^2 = (N_{\text{sim}} - 1)^{-1} \sum_{t=1}^{N_{\text{sim}}} (W_t - \widehat{\ell})^2$. Then, we use the following accuracy measures.

1. When the analytical solution is available, we report the *absolute error*. For the estimator $\widehat{\ell}$ of the true quantity ℓ, the absolute error is defined via
$$\delta_{\text{abs}} \overset{\text{def}}{=} \frac{\left|\widehat{\ell} - \ell\right|}{\ell}.$$
2. When we are not aware of analytical solution, we report the estimated RE, which is given by
$$\widehat{RE} = \frac{\widehat{\sigma} / \sqrt{N_{\text{sim}}}}{\widehat{\ell}}.$$
3. When applicable, specifically, when we would like to report the accuracy of individual spectra components, we also report the 95% *relative confidence interval* (CI) of a random variable $\widehat{\ell}$ [47]. The CI is defined via
$$\left(\widehat{\ell} \times \left(1 \pm 2\, z_{1-\alpha/2} \frac{\widehat{\sigma} / \sqrt{N_{\text{sim}}}}{\widehat{\ell}}\right)\right),$$
where $1 - \alpha/2 = 0.95$, and $z_{1-\alpha/2}$ is the $(1 - \alpha/2)$ quantile of the standard normal distribution.

4.1. The bridge network

While the bridge network is a toy example, we include it with a view to highlighting several major ideas used in this section. Namely, we show the importance of the stability of the $\widehat{\text{RE}}$, and introduce the statistical methodology which is used to report the BIM results. For the bridge network, we set $N = 1,000$ and $N_{\text{sim}} = 100$. The 95% relative CIs for the spectra objects are as follows

$$\widehat{\mathbf{s}}(G, \mathcal{R}) = \Big((0.0, 0.0),\ (0.6602, 0.6705),\ (1.0, 1.0)\Big), \tag{20}$$

$$\begin{pmatrix} \widehat{\mathbf{s}}^{(\text{bim}_1)}(G, \mathcal{R}) \\ \widehat{\mathbf{s}}^{(\text{bim}_2)}(G, \mathcal{R}) \\ \widehat{\mathbf{s}}^{(\text{bim}_3)}(G, \mathcal{R}) \end{pmatrix} = \begin{pmatrix} (0.0, 0.0), & (0.6601, 0.6705), & (1.0, 1.0) \\ (0.0, 0.0), & (0.3277, 0.3374), & (1.0, 1.0) \\ (0.0, 0.0), & (0.3275, 0.3381), & (1.0, 1.0) \end{pmatrix}. \tag{21}$$

Table 4 presents the point estimator of the bridge network unreliability $\widehat{r}(G, \mathcal{R}, q)$, the $\widehat{\text{RE}}$, and the absolute error δ_{abs}, for various values of SRLG unreliability q. Table 4 is instructive in the sense that one can observe that the $\widehat{\text{RE}}$ is very stable. In particular, the $\widehat{\text{RE}}$ does not grow when q (and $\widehat{r}(G, \mathcal{R}, q)$), become very small. We would also like to remind that it will be computationally prohibitive to estimate the network unreliability via CMC for very small values of q (such as $q = 1 \times 10^{-20}$). In addition, the absolute error (which is available for this small case study), is also very small and stable. The stability of the $\widehat{\text{RE}}$ and the absolute error is not very surprising, since the spectra components in (20) are very close to the analytical values obtained in Example 2.

Table 4 The bridge network: the average unreliability estimators and the corresponding $\widehat{\text{RE}}$s and δ_{abs}s for various values of SRLG failure probability q using the $N = 1,000$ sample size and $N_{\text{sim}} = 100$. The average CPU time for a single PMC run is 5×10^{-4} seconds.

q	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$	δ_{abs}
5×10^{-2}	5.256×10^{-2}	3.378×10^{-3}	3.455×10^{-3}
1×10^{-2}	1.014×10^{-2}	3.804×10^{-3}	3.892×10^{-3}
5×10^{-4}	5.023×10^{-4}	3.913×10^{-3}	4.004×10^{-3}
1×10^{-4}	1.004×10^{-4}	3.918×10^{-3}	4.009×10^{-3}
5×10^{-6}	5.020×10^{-6}	3.919×10^{-3}	4.010×10^{-3}
1×10^{-6}	1.004×10^{-6}	3.919×10^{-3}	4.010×10^{-3}
5×10^{-8}	5.020×10^{-8}	3.919×10^{-3}	4.010×10^{-3}
1×10^{-8}	1.004×10^{-8}	3.919×10^{-3}	4.010×10^{-3}
5×10^{-10}	5.020×10^{-10}	3.919×10^{-3}	4.010×10^{-3}
1×10^{-10}	1.004×10^{-10}	3.919×10^{-3}	4.010×10^{-3}
5×10^{-12}	5.020×10^{-12}	3.919×10^{-3}	4.010×10^{-3}
1×10^{-12}	1.004×10^{-12}	3.919×10^{-3}	4.010×10^{-3}
5×10^{-14}	5.020×10^{-14}	3.919×10^{-3}	4.010×10^{-3}
1×10^{-14}	1.004×10^{-14}	3.919×10^{-3}	4.010×10^{-3}
5×10^{-16}	5.020×10^{-16}	3.919×10^{-3}	4.010×10^{-3}
1×10^{-16}	1.004×10^{-16}	3.919×10^{-3}	4.010×10^{-3}
5×10^{-18}	5.020×10^{-18}	3.919×10^{-3}	4.010×10^{-3}
1×10^{-18}	1.004×10^{-18}	3.919×10^{-3}	4.010×10^{-3}
5×10^{-20}	5.020×10^{-20}	3.919×10^{-3}	4.010×10^{-3}
1×10^{-20}	1.004×10^{-20}	3.919×10^{-3}	4.010×10^{-3}

However, when we try to exploit the fact that the BIM spectra in (21) is also close to the analytical values (see Example 4), the estimator (17) is very unstable. Table 5 shows that both the $\widehat{\text{RE}}$ and the δ_{abs} escalate rapidly, and the obtained BIM estimator become improper for any adequate IM analysis.

Table 5 The bridge network: average BIM estimators $\left(\widehat{\text{BIM}}_1, \widehat{\text{BIM}}_2, \text{ and } \widehat{\text{BIM}}_3\right)$ obtained via (17), and the corresponding $\widehat{\text{RE}}$s and δ_{abs}s for various values of SRLG failure probability q using the $N = 1,000$ sample size and $N_{\text{sim}} = 100$.

	BIM_1			BIM_2			BIM_3		
q	$\widehat{\text{BIM}}_1$	$\widehat{\text{RE}}$	δ_{abs}	$\widehat{\text{BIM}}_2$	$\widehat{\text{RE}}$	δ_{abs}	$\widehat{\text{BIM}}_3$	$\widehat{\text{RE}}$	δ_{abs}
5×10^{-2}	0.997	1.870×10^{-4}	1.900×10^{-4}	0.049	0.076	2.800×10^{-2}	0.050	0.069	4.420×10^{-2}
1×10^{-2}	0.999	3.900×10^{-5}	4.000×10^{-5}	0.011	0.346	1.560×10^{-1}	0.012	0.298	2.370×10^{-1}
5×10^{-4}	1	1.970×10^{-6}	1.750×10^{-6}	0.002	1.917	0.320×10^{1}	0.003	1.271	0.482×10^{1}
1×10^{-4}	1	3.930×10^{-7}	1.000×10^{-8}	0.002	2.367	1.600×10^{1}	0.003	1.473	2.410×10^{1}
5×10^{-6}	1	1.970×10^{-8}	2.500×10^{-11}	0.002	2.507	3.200×10^{2}	0.002	1.531	4.820×10^{2}
1×10^{-6}	1	3.930×10^{-9}	1.000×10^{-12}	0.002	2.513	1.600×10^{3}	0.002	1.534	2.410×10^{3}
5×10^{-8}	1	1.970×10^{-10}	0	0.002	2.515	3.200×10^{4}	0.002	1.534	4.820×10^{4}
1×10^{-8}	1	3.930×10^{-11}	0	0.002	2.515	1.600×10^{5}	0.002	1.534	2.410×10^{5}
5×10^{-10}	1	1.970×10^{-12}	0	0.002	2.515	3.200×10^{6}	0.002	1.534	4.820×10^{6}
1×10^{-10}	1	3.930×10^{-13}	0	0.002	2.515	1.600×10^{7}	0.002	1.534	2.410×10^{7}
5×10^{-12}	1	1.970×10^{-14}	0	0.002	2.515	3.200×10^{8}	0.002	1.534	4.820×10^{8}
1×10^{-12}	1	3.930×10^{-15}	0	0.002	2.515	1.600×10^{9}	0.002	1.534	2.410×10^{9}
5×10^{-14}	1	1.970×10^{-16}	0	0.002	2.515	3.200×10^{10}	0.002	1.534	4.820×10^{10}
1×10^{-14}	1	3.930×10^{-17}	0	0.002	2.515	1.600×10^{11}	0.002	1.534	2.410×10^{11}
5×10^{-16}	1	1.920×10^{-18}	0	0.002	2.515	3.200×10^{12}	0.002	1.534	4.820×10^{12}
1×10^{-16}	1	0	0	0.002	2.515	1.600×10^{13}	0.002	1.534	2.410×10^{13}
5×10^{-18}	1	0	0	0.002	2.515	3.200×10^{14}	0.002	1.534	4.820×10^{14}
1×10^{-18}	1	0	0	0.002	2.515	1.600×10^{15}	0.002	1.534	2.410×10^{15}
5×10^{-20}	1	0	0	0.002	2.515	3.200×10^{16}	0.002	1.534	4.820×10^{16}
1×10^{-20}	1	0	0	0.002	2.515	1.600×10^{17}	0.002	1.534	2.410×10^{17}

Due to the negative result in Table 5, we proceed with the application of Corollary 1. Note that a comparison of the second column in (21) shows that $\widehat{\mathbf{s}}^{(\text{bim}_1)}(G, \mathcal{R})$ is larger than $\widehat{\mathbf{s}}^{(\text{bim}_2)}(G, \mathcal{R})$ and $\widehat{\mathbf{s}}^{(\text{bim}_3)}(G, \mathcal{R})$. According to Corollary 1, this means that R_1 is the most important component. The importance of R_1 is statistically significant. The latter is formally verified by the execution of the *Kruskal–Wallis* one-way analysis of variance test [48]. The obtained $p-$value is 1.348×10^{-43}, so the *null* hypothesis (the medians of all groups are equal), is rejected. Figure 3 shows the corresponding BIM results, specifically, the values of $\widehat{s}_2^{(\text{bim}_j)}$ for $j = 1, 2, 3$.

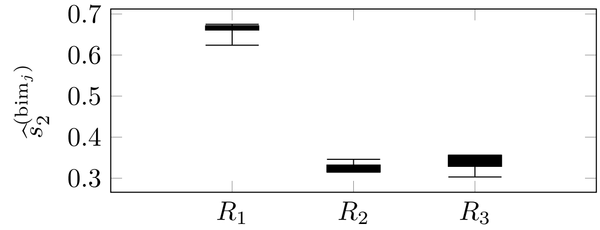


Figure 3: The bridge network: BIM results obtained using the $N = 1,000$ sample size and $N_{\text{sim}} = 100$.

Remark 4 (Statistical significance)**.** Though this section, we use the Kruskal–Wallis test. However, it is worth noting that one could choose to use CIs (as in (21)), that are readily available, too.

4.2. The wheel graph

In order to benchmark the PMC algorithm performance on larger instances, and in particular to verify the accuracy of the reliability estimation, and the applicability of Corollary 1, we consider a very structured *wheel* graph $\mathcal{W}(m)$. The $\mathcal{W}(m)$ network has m vertices and $2(m-1)$ edges. The central vertex v_1 is connected to a ring of size $m-1$. In our case study, all edges from the central vertex belong to R_1 and each edge in the ring is associated with its own SRLG, namely, with $R_2, \ldots, R_m$. The $\mathcal{W}(5)$ graph is depicted in Figure 4.

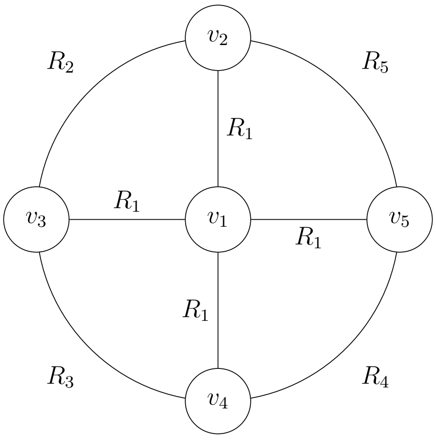


Figure 4: The $\mathcal{W}(5)$ wheel graph with five vertices, eight edges, and 5 SRLGs.

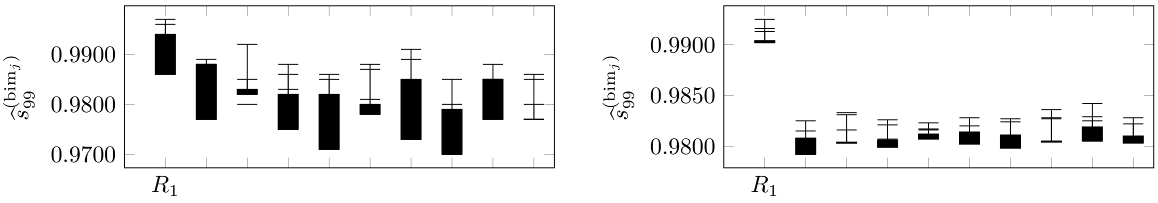
For a wheel graph, the network unreliability is equal to $\mathbb{P}(R_1$ is in the *down* state$) = q$. The latter allows benchmarking the absolute error of the network unreliability estimator. Moreover, R_1 is the most important component regardless of the wheel size, since this is the only component whose removal will cause the network disconnection. Next, we consider two wheel instances, the $\mathcal{W}(100)$ and the $\mathcal{W}(300)$, and execute the PMC algorithm with the $N = 1,000$ and the $N = 10,000$ sample sizes. As for the bridge network example, we set the number of independent runs to be $N_{\text{sim}} = 100$.

4.2.1. The $\mathcal{W}(100)$ graph

Table 6 summarizes the unreliability results for the $\mathcal{W}(100)$ graph. In particular, the table shows the stability of both the $\widehat{\mathrm{RE}}$s and the δ_{abs}s. Figure 5 depicts the corresponding BIM results for the first ten components with the largest $\widehat{s}_{99}^{(\mathrm{bim}_j)}$ values. There is clear evidence of the importance of R_1. As expected, by increasing the sample size N from $N = 1,000$ to $N = 10,000$, we can obtain a decrease in the $\widehat{\mathrm{RE}}$, in the δ_{abs} (Table 6), and in the $p-$value (Figure 5(b)).

Table 6 The $\mathcal{W}(100)$ graph: average unreliability estimators and the corresponding $\widehat{\mathrm{RE}}$s and δ_{abs}s for various values of SRLG failure probability q. The average CPU time ($N_{\mathrm{sim}} = 100$) for a single PMC run with $N = 1,000$ and $N = 10,000$ sample size is 0.0622 and 0.5994 seconds, respectively.

		$N = 1,000$			$N = 10,000$	
q	$\widehat{r}(G,\mathcal{R},q)$	$\widehat{\mathrm{RE}}$	δ_{abs}	$\widehat{r}(G,\mathcal{R},q)$	$\widehat{\mathrm{RE}}$	δ_{abs}
5×10^{-2}	4.932×10^{-2}	1.183×10^{-2}	1.357×10^{-2}	5.003×10^{-2}	3.744×10^{-3}	5.236×10^{-4}
1×10^{-2}	9.704×10^{-3}	2.212×10^{-2}	2.961×10^{-2}	1.001×10^{-2}	6.451×10^{-3}	1.462×10^{-3}
5×10^{-4}	4.799×10^{-4}	3.190×10^{-2}	4.024×10^{-2}	4.996×10^{-4}	9.865×10^{-3}	8.673×10^{-4}
1×10^{-4}	9.592×10^{-5}	3.255×10^{-2}	4.085×10^{-2}	9.989×10^{-5}	1.009×10^{-2}	1.052×10^{-3}
5×10^{-6}	4.795×10^{-6}	3.270×10^{-2}	4.099×10^{-2}	4.995×10^{-6}	1.014×10^{-2}	1.098×10^{-3}
1×10^{-6}	9.590×10^{-7}	3.271×10^{-2}	4.100×10^{-2}	9.989×10^{-7}	1.014×10^{-2}	1.100×10^{-3}
5×10^{-8}	4.795×10^{-8}	3.271×10^{-2}	4.100×10^{-2}	4.995×10^{-8}	1.014×10^{-2}	1.100×10^{-3}
1×10^{-8}	9.590×10^{-9}	3.271×10^{-2}	4.100×10^{-2}	9.989×10^{-9}	1.014×10^{-2}	1.100×10^{-3}
5×10^{-10}	4.795×10^{-10}	3.271×10^{-2}	4.100×10^{-2}	4.995×10^{-10}	1.014×10^{-2}	1.100×10^{-3}
1×10^{-10}	9.590×10^{-11}	3.271×10^{-2}	4.100×10^{-2}	9.989×10^{-11}	1.014×10^{-2}	1.100×10^{-3}
5×10^{-12}	4.795×10^{-12}	3.271×10^{-2}	4.100×10^{-2}	4.995×10^{-12}	1.014×10^{-2}	1.100×10^{-3}
1×10^{-12}	9.590×10^{-13}	3.271×10^{-2}	4.100×10^{-2}	9.989×10^{-13}	1.014×10^{-2}	1.100×10^{-3}
5×10^{-14}	4.795×10^{-14}	3.271×10^{-2}	4.100×10^{-2}	4.995×10^{-14}	1.014×10^{-2}	1.100×10^{-3}
1×10^{-14}	9.590×10^{-15}	3.271×10^{-2}	4.100×10^{-2}	9.989×10^{-15}	1.014×10^{-2}	1.100×10^{-3}
5×10^{-16}	4.795×10^{-16}	3.271×10^{-2}	4.100×10^{-2}	4.995×10^{-16}	1.014×10^{-2}	1.100×10^{-3}
1×10^{-16}	9.590×10^{-17}	3.271×10^{-2}	4.100×10^{-2}	9.989×10^{-17}	1.014×10^{-2}	1.100×10^{-3}
5×10^{-18}	4.795×10^{-18}	3.271×10^{-2}	4.100×10^{-2}	4.995×10^{-18}	1.014×10^{-2}	1.100×10^{-3}
1×10^{-18}	9.590×10^{-19}	3.271×10^{-2}	4.100×10^{-2}	9.989×10^{-19}	1.014×10^{-2}	1.100×10^{-3}
5×10^{-20}	4.795×10^{-20}	3.271×10^{-2}	4.100×10^{-2}	4.995×10^{-20}	1.014×10^{-2}	1.100×10^{-3}
1×10^{-20}	9.590×10^{-21}	3.271×10^{-2}	4.100×10^{-2}	9.989×10^{-21}	1.014×10^{-2}	1.100×10^{-3}



(a) The $\mathcal{W}(100)$ graph BIM results using $N = 1,000$ sample size. The $s_{99}^{(\mathrm{bim}_1)}$ appears to be larger than $s_{99}^{(\mathrm{bim}_j)}$ for $2 \leq j \leq 100$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p − value is 6.147×10^{-24}.

(b) The $\mathcal{W}(100)$ graph BIM results using $N = 10,000$ sample size. The $s_{99}^{(\mathrm{bim}_1)}$ appears to be larger than $s_{99}^{(\mathrm{bim}_j)}$ for $2 \leq j \leq 100$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p − value is 8.876×10^{-29}.

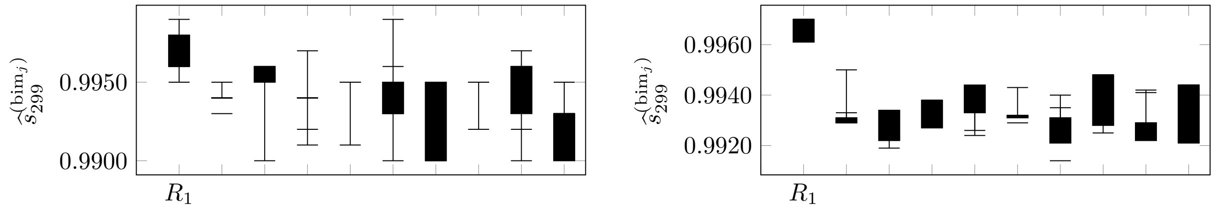
Figure 5: The $\mathcal{W}(100)$ graph: BIM results obtained using the $N = 1,000$ and the $N = 10,000$ sample sizes, and $N_{\mathrm{sim}} = 100$.

4.2.2. The $\mathcal{W}(300)$ graph

Similar to the $\mathcal{W}(100)$ case, we proceed with the $\mathcal{W}(300)$ instance. Table 7 summarizes the unreliability results for the $\mathcal{W}(300)$ graph. As for the $\mathcal{W}(100)$ network, we can see that $\widehat{\mathrm{RE}}$s and δ_{abs}s show stability. Figure 6 depicts the corresponding BIM results for the first ten components with the largest $\widehat{s}_{299}^{(\mathrm{bim}_j)}$ values. This example is instructive in the sense that we had to increase the sample size N from 1,000 to 10,000, in order to achieve a statistically significant result with respect to the importance of R_1. We conjecture that bigger sample size is required because of the increased number of SRLG components that have to be estimated, namely, 300 in the $\mathcal{W}(300)$ graph as compared to 100 in the $\mathcal{W}(100)$ graph.

Table 7 The $\mathcal{W}(300)$ graph: average unreliability estimators and the corresponding $\widehat{\text{RE}}$s and δ_{abs}s for various values of SRLG failure probability q. The average CPU time ($N_{\text{sim}} = 100$) for a single PMC run with $N = 1,000$ and $N = 10,000$ sample size is 0.3273 and 3.2976 seconds, respectively.

		$N = 1,000$			$N = 10,000$	
q	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$	δ_{abs}	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$	δ_{abs}
5×10^{-2}	5.002×10^{-2}	1.323×10^{-2}	3.539×10^{-4}	4.991×10^{-2}	4.220×10^{-3}	1.750×10^{-3}
1×10^{-2}	1.016×10^{-2}	2.295×10^{-2}	1.578×10^{-2}	9.991×10^{-3}	9.250×10^{-3}	8.694×10^{-4}
5×10^{-4}	5.252×10^{-4}	4.862×10^{-2}	5.042×10^{-2}	5.145×10^{-4}	1.746×10^{-2}	2.905×10^{-2}
1×10^{-4}	1.057×10^{-4}	5.165×10^{-2}	5.712×10^{-2}	1.031×10^{-4}	1.831×10^{-2}	3.139×10^{-2}
5×10^{-6}	5.295×10^{-6}	5.242×10^{-2}	5.890×10^{-2}	5.160×10^{-6}	1.852×10^{-2}	3.197×10^{-2}
1×10^{-6}	1.059×10^{-6}	5.245×10^{-2}	5.898×10^{-2}	1.032×10^{-6}	1.853×10^{-2}	3.199×10^{-2}
5×10^{-8}	5.295×10^{-8}	5.246×10^{-2}	5.900×10^{-2}	5.160×10^{-8}	1.853×10^{-2}	3.200×10^{-2}
1×10^{-8}	1.059×10^{-8}	5.246×10^{-2}	5.900×10^{-2}	1.032×10^{-8}	1.853×10^{-2}	3.200×10^{-2}
5×10^{-10}	5.295×10^{-10}	5.246×10^{-2}	5.900×10^{-2}	5.160×10^{-10}	1.853×10^{-2}	3.200×10^{-2}
1×10^{-10}	1.059×10^{-10}	5.246×10^{-2}	5.900×10^{-2}	1.032×10^{-10}	1.853×10^{-2}	3.200×10^{-2}
5×10^{-12}	5.295×10^{-12}	5.246×10^{-2}	5.900×10^{-2}	5.160×10^{-12}	1.853×10^{-2}	3.200×10^{-2}
1×10^{-12}	1.059×10^{-12}	5.246×10^{-2}	5.900×10^{-2}	1.032×10^{-12}	1.853×10^{-2}	3.200×10^{-2}
5×10^{-14}	5.295×10^{-14}	5.246×10^{-2}	5.900×10^{-2}	5.160×10^{-14}	1.853×10^{-2}	3.200×10^{-2}
1×10^{-14}	1.059×10^{-14}	5.246×10^{-2}	5.900×10^{-2}	1.032×10^{-14}	1.853×10^{-2}	3.200×10^{-2}
5×10^{-16}	5.295×10^{-16}	5.246×10^{-2}	5.900×10^{-2}	5.160×10^{-16}	1.853×10^{-2}	3.200×10^{-2}
1×10^{-16}	1.059×10^{-16}	5.246×10^{-2}	5.900×10^{-2}	1.032×10^{-16}	1.853×10^{-2}	3.200×10^{-2}
5×10^{-18}	5.295×10^{-18}	5.246×10^{-2}	5.900×10^{-2}	5.160×10^{-18}	1.853×10^{-2}	3.200×10^{-2}
1×10^{-18}	1.059×10^{-18}	5.246×10^{-2}	5.900×10^{-2}	1.032×10^{-18}	1.853×10^{-2}	3.200×10^{-2}
5×10^{-20}	5.295×10^{-20}	5.246×10^{-2}	5.900×10^{-2}	5.160×10^{-20}	1.853×10^{-2}	3.200×10^{-2}
1×10^{-20}	1.059×10^{-20}	5.246×10^{-2}	5.900×10^{-2}	1.032×10^{-20}	1.853×10^{-2}	3.200×10^{-2}



(a) The $\mathcal{W}(300)$ graph BIM results using $N = 1,000$ sample size. It is not clear if the $s_{299}^{(\text{bim}_1)}$ is indeed larger than $s_{99}^{(\text{bim}_j)}$ for $2 \le j \le 300$. The result is not statistically significant. In particular, with the $N = 1,000$ sample size, we fail to reject the *null* hypothesis of the Kruskal–Wallis test; the corresponding p − value is 0.5787.

(b) The $\mathcal{W}(300)$ graph BIM results using $N = 10,000$ sample size. The $s_{299}^{(\text{bim}_1)}$ appears to be larger than $s_{99}^{(\text{bim}_j)}$ for $2 \le j \le 300$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p − value is 6.237×10^{-7}.

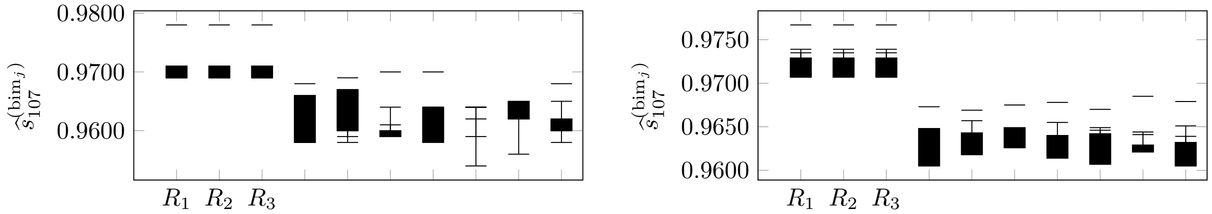
Figure 6: The $\mathcal{W}(300)$ graph: BIM results obtained using the $N = 1,000$ and the $N = 10,000$ sample sizes, and $N_{\text{sim}} = 100$.

4.3. The random internet network

In this case study, we consider a random internet graph topology [44]. The generated graph has 200 vertices, 288 edges, and 108 SRLGs. In this network, we only consider links between the so-called T and M vertices (these vertices are transit providers), and all other edges are assumed to be reliable. We assigned 70 links to the first 3 SRLGs, namely, to R_1, R_2, and R_3. For each R_i ($i \in \{1, 2, 3\}$), 70 edges were chosen uniformly at random from the set of 288 edges. For the remaining set of SRLGs, we assign one edge to each R_i for $i \in \{4, \ldots, 108\}$, where an edge for R_i was selected uniformly at random from the set of 288 edges, too. We run the PMC algorithm with $N = 1,000$, $N = 10,000$, and $N_{\text{sim}} = 100$. Table 8 summarizes the unreliability results. As for the bridge and the wheel case studies, the $\widehat{\text{RE}}$ shows stability. Figure 7 depicts the corresponding BIM results for the first ten components with the largest $\widehat{s}_{107}^{(\text{bim}_j)}$ values, and one can verify that (as expected), there is clear statistically-significant evidence of the importance of the R_1, the R_2 and the R_3.

Table 8 The random internet network: average unreliability estimators and the corresponding $\widehat{\text{RE}}$s for various values of SRLG failure probability q. The average CPU time ($N_{\text{sim}} = 100$) for a single PMC run with $N = 1,000$ and $N = 10,000$ sample size is 0.0973 and 0.9904 seconds, respectively.

	$N = 1,000$		$N = 10,000$	
q	$\widehat{r}(G,\mathcal{R},q)$	$\widehat{\text{RE}}$	$\widehat{r}(G,\mathcal{R},q)$	$\widehat{\text{RE}}$
5×10^{-2}	1.485×10^{-1}	6.483×10^{-3}	1.513×10^{-1}	1.963×10^{-3}
1×10^{-2}	2.942×10^{-2}	1.242×10^{-2}	3.001×10^{-2}	3.797×10^{-3}
5×10^{-4}	1.471×10^{-3}	1.751×10^{-2}	1.493×10^{-3}	6.116×10^{-3}
1×10^{-4}	2.942×10^{-4}	1.785×10^{-2}	2.986×10^{-4}	6.271×10^{-3}
5×10^{-6}	1.471×10^{-5}	1.794×10^{-2}	1.493×10^{-5}	6.309×10^{-3}
1×10^{-6}	2.942×10^{-6}	1.794×10^{-2}	2.986×10^{-6}	6.311×10^{-3}
5×10^{-8}	1.471×10^{-7}	1.794×10^{-2}	1.493×10^{-7}	6.311×10^{-3}
1×10^{-8}	2.942×10^{-8}	1.794×10^{-2}	2.986×10^{-8}	6.311×10^{-3}
5×10^{-10}	1.471×10^{-9}	1.794×10^{-2}	1.493×10^{-9}	6.311×10^{-3}
1×10^{-10}	2.942×10^{-10}	1.794×10^{-2}	2.986×10^{-10}	6.311×10^{-3}
5×10^{-12}	1.471×10^{-11}	1.794×10^{-2}	1.493×10^{-11}	6.311×10^{-3}
1×10^{-12}	2.942×10^{-12}	1.794×10^{-2}	2.986×10^{-12}	6.311×10^{-3}
5×10^{-14}	1.471×10^{-13}	1.794×10^{-2}	1.493×10^{-13}	6.311×10^{-3}
1×10^{-14}	2.942×10^{-14}	1.794×10^{-2}	2.986×10^{-14}	6.311×10^{-3}
5×10^{-16}	1.471×10^{-15}	1.794×10^{-2}	1.493×10^{-15}	6.311×10^{-3}
1×10^{-16}	2.942×10^{-16}	1.794×10^{-2}	2.986×10^{-16}	6.311×10^{-3}
5×10^{-18}	1.471×10^{-17}	1.794×10^{-2}	1.493×10^{-17}	6.311×10^{-3}
1×10^{-18}	2.942×10^{-18}	1.794×10^{-2}	2.986×10^{-18}	6.311×10^{-3}
5×10^{-20}	1.471×10^{-19}	1.794×10^{-2}	1.493×10^{-19}	6.311×10^{-3}
1×10^{-20}	2.942×10^{-20}	1.794×10^{-2}	2.986×10^{-20}	6.311×10^{-3}



(a) The random internet network BIM results using $N = 1,000$ sample size. The $s_{107}^{(\text{bim}_j)}$s for $1 \leq j \leq 3$ appear to be larger than $s_{107}^{(\text{bim}_j)}$ for $4 \leq j \leq 108$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding $p-$ value is 1.337×10^{-64}.

(b) The random internet network BIM results using $N = 10,000$ sample size. The $s_{107}^{(\text{bim}_j)}$s for $1 \leq j \leq 3$ appear to be larger than $s_{107}^{(\text{bim}_j)}$ for $4 \leq j \leq 108$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding $p-$ value is 2.137×10^{-125}.

Figure 7: The random internet network: BIM results obtained using the $N = 1,000$ and the $N = 10,000$ sample sizes, and $N_{\text{sim}} = 100$.

We proceed with the bench-marking of the PMC algorithm for different failure regimes and extensions (see Remark 1).

4.3.1. The random internet network with SRLG capacity regime

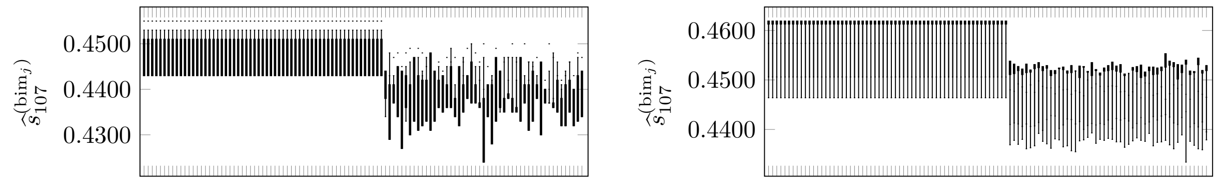
Here, we use the same generated graph with 200 vertices, 288 edges, and 108 SRLGs. However, in addition, for each SRLG, we set a *capacity* number, which is chosen uniformly at random between 1 and 10. It is convenient to view the capacity as the number of fibers in a specific duct. Now, for the network to be reliable, we also require that it will be both fully connected and utilize 99% of the available capacity, where the latter is defined as the sum of capacities in all SRLGs that are in the *up* state. We run the PMC algorithm with $N = 1,000$, $N = 10,000$, and

$N_{\text{sim}} = 100$. Table 9 summarizes the unreliability results for this graph. The $\widehat{\text{RE}}$ is stable. It is also interesting to note that the unreliability increased as compared to the random internet graph case study without the capacity constraint. This growth in the unreliability is expected, because we have an additional constraint on the required capacity.

Figure 8 shows all (108) BIM components. This time, we identified 59 components of major importance. The list of indices of important SRLGs is as follows: 1, 2, 3, 4, 5, 6, 9, 11, 12, 14, 15, 23, 26, 27, 28, 30, 31, 32, 33, 34, 35, 38, 39, 43, 45, 46, 47, 56, 58, 59, 60, 64, 67, 68, 69, 70, 72, 73, 74, 75, 76, 78, 79, 80, 81, 82, 83, 84, 86, 89, 91, 95, 96, 99, 101, 103, 104, 105, 106. One can see that under the capacity constraint setting, about half of SRLG components are important. We conjecture that this is due to the fact that the capacity requirement was set to 99%, thus allowing a sufficient freedom in the sense of components importance.

Table 9 The random internet network with SRLG capacity regime: average unreliability estimators and the corresponding $\widehat{\text{RE}}$s for various values of SRLG failure probability q. The average CPU time ($N_{\text{sim}} = 100$) for a single PMC run with $N = 1,000$ and $N = 10,000$ sample size is 0.1661 and 1.5061 seconds, respectively.

	$N = 1,000$		$N = 10,000$	
q	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$
5×10^{-2}	9.793×10^{-1}	1.025×10^{-4}	9.794×10^{-1}	2.923×10^{-5}
1×10^{-2}	4.750×10^{-1}	1.397×10^{-3}	4.763×10^{-1}	4.397×10^{-4}
5×10^{-4}	2.905×10^{-2}	2.640×10^{-3}	2.922×10^{-2}	8.655×10^{-4}
1×10^{-4}	5.852×10^{-3}	2.719×10^{-3}	5.886×10^{-3}	8.929×10^{-4}
5×10^{-6}	2.931×10^{-4}	2.739×10^{-3}	2.948×10^{-4}	8.996×10^{-4}
1×10^{-6}	5.863×10^{-5}	2.739×10^{-3}	5.897×10^{-5}	8.999×10^{-4}
5×10^{-8}	2.931×10^{-6}	2.740×10^{-3}	2.948×10^{-6}	8.999×10^{-4}
1×10^{-8}	5.863×10^{-7}	2.740×10^{-3}	5.897×10^{-7}	8.999×10^{-4}
5×10^{-10}	2.931×10^{-8}	2.740×10^{-3}	2.948×10^{-8}	8.999×10^{-4}
1×10^{-10}	5.863×10^{-9}	2.740×10^{-3}	5.897×10^{-9}	8.999×10^{-4}
5×10^{-12}	2.931×10^{-10}	2.740×10^{-3}	2.948×10^{-10}	8.999×10^{-4}
1×10^{-12}	5.863×10^{-11}	2.740×10^{-3}	5.897×10^{-11}	8.999×10^{-4}
5×10^{-14}	2.931×10^{-12}	2.740×10^{-3}	2.948×10^{-12}	8.999×10^{-4}
1×10^{-14}	5.863×10^{-13}	2.740×10^{-3}	5.897×10^{-13}	8.999×10^{-4}
5×10^{-16}	2.931×10^{-14}	2.740×10^{-3}	2.948×10^{-14}	8.999×10^{-4}
1×10^{-16}	5.863×10^{-15}	2.740×10^{-3}	5.897×10^{-15}	8.999×10^{-4}
5×10^{-18}	2.931×10^{-16}	2.740×10^{-3}	2.948×10^{-16}	8.999×10^{-4}
1×10^{-18}	5.863×10^{-17}	2.740×10^{-3}	5.897×10^{-17}	8.999×10^{-4}
5×10^{-20}	2.931×10^{-18}	2.740×10^{-3}	2.948×10^{-18}	8.999×10^{-4}
1×10^{-20}	5.863×10^{-19}	2.740×10^{-3}	5.897×10^{-19}	8.999×10^{-4}



(a) The random internet network with SRLG capacity regime BIM results using $N = 1,000$ sample size. About half of $s_{107}^{(\text{bim}_j)}$s appear to be more important than the others. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding $p-$ value is 2.459×10^{-137}.

(b) The random internet network with SRLG capacity regime BIM results using $N = 10,000$ sample size. About half of $s_{107}^{(\text{bim}_j)}$s appear to be more important than the others. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding $p-$ value is 0.

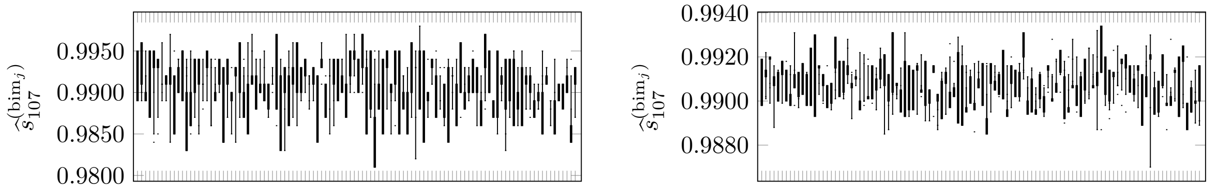
Figure 8: The random internet network with SRLG capacity regime: BIM results obtained using the $N = 1,000$ and the $N = 10,000$ sample sizes, and $N_{\text{sim}} = 100$.

4.3.2. The random internet network with SRLG capacities and partial connectivity regime

Again, we utilize the same randomly generated internet network with 200 vertices, 288 edges, and 108 SRLGs. As in the previous example, for each SRLG, the *capacity* number is chosen uniformly at random from the $\{1, \dots, 10\}$ set. In addition, we would like to benchmark the partial connectivity regime. Under this regime, for the network to be in the *up* state, we require that 1) the network utilizes 80% of the available capacity, and 2) at least 92.5% of vertices are connected. In other words, we allow partial connectivity. Table 10 summarizes the unreliability results for this graph. The $\widehat{\text{RE}}$ shows stability and due to the relaxed requirement of the partial connectivity, the system unreliability experience a dramatic decrease. Figure 9 shows all BIM components. However, for both the $N = 1,000$ and the $N = 10,000$ sample size, we fail to identify the set of most significant SRLGs.

Table 10 The random internet network with SRLG capacities and partial connectivity regime: average unreliability estimators and the corresponding $\widehat{\text{RE}}$s for various values of SRLG failure probability q. The average CPU time ($N_{\text{sim}} = 100$) for a single PMC run with $N = 1,000$ and $N = 10,000$ sample size is 0.1304 and 1.3182 seconds, respectively.

	$N = 1,000$		$N = 10,000$	
q	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$
5×10^{-2}	2.296×10^{-2}	1.449×10^{-2}	2.337×10^{-2}	4.675×10^{-3}
1×10^{-2}	9.782×10^{-4}	4.591×10^{-2}	9.763×10^{-4}	1.240×10^{-2}
5×10^{-4}	2.562×10^{-6}	7.225×10^{-2}	2.453×10^{-6}	2.296×10^{-2}
1×10^{-4}	1.028×10^{-7}	7.392×10^{-2}	9.811×10^{-8}	2.371×10^{-2}
5×10^{-6}	2.571×10^{-10}	7.433×10^{-2}	2.453×10^{-10}	2.390×10^{-2}
1×10^{-6}	1.028×10^{-11}	7.435×10^{-2}	9.811×10^{-12}	2.391×10^{-2}
5×10^{-8}	2.571×10^{-14}	7.435×10^{-2}	2.453×10^{-14}	2.391×10^{-2}
1×10^{-8}	1.028×10^{-15}	7.435×10^{-2}	9.811×10^{-16}	2.391×10^{-2}
5×10^{-10}	2.571×10^{-18}	7.435×10^{-2}	2.453×10^{-18}	2.391×10^{-2}
1×10^{-10}	1.028×10^{-19}	7.435×10^{-2}	9.811×10^{-20}	2.391×10^{-2}
5×10^{-12}	2.571×10^{-22}	7.435×10^{-2}	2.453×10^{-22}	2.391×10^{-2}
1×10^{-12}	1.028×10^{-23}	7.435×10^{-2}	9.811×10^{-24}	2.391×10^{-2}
5×10^{-14}	2.571×10^{-26}	7.435×10^{-2}	2.453×10^{-26}	2.391×10^{-2}
1×10^{-14}	1.028×10^{-27}	7.435×10^{-2}	9.811×10^{-28}	2.391×10^{-2}
5×10^{-16}	2.571×10^{-30}	7.435×10^{-2}	2.453×10^{-30}	2.391×10^{-2}
1×10^{-16}	1.028×10^{-31}	7.435×10^{-2}	9.811×10^{-32}	2.391×10^{-2}
5×10^{-18}	2.571×10^{-34}	7.435×10^{-2}	2.453×10^{-34}	2.391×10^{-2}
1×10^{-18}	1.028×10^{-35}	7.435×10^{-2}	9.811×10^{-36}	2.391×10^{-2}
5×10^{-20}	2.571×10^{-38}	7.435×10^{-2}	2.453×10^{-38}	2.391×10^{-2}
1×10^{-20}	1.028×10^{-39}	7.435×10^{-2}	9.811×10^{-40}	2.391×10^{-2}

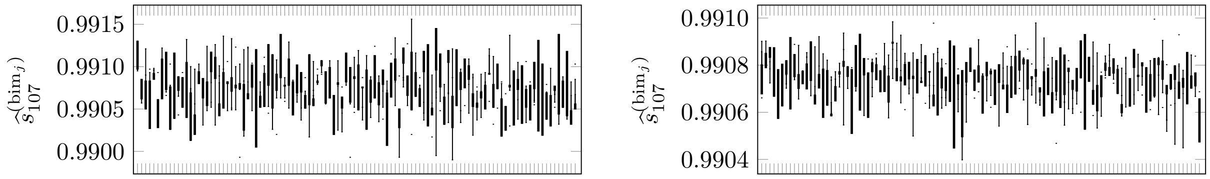


(a) The random internet network with SRLG capacities and partial connectivity regime BIM results using $N = 1,000$ sample size. It is not clear if there exist more important components. Specifically, we fail to reject the *null* hypothesis of the Kruskal–Wallis test; the corresponding $p-$ value is 0.148.

(b) The random internet network with SRLG capacities and partial connectivity regime BIM results using $N = 10,000$ sample size. It is not clear if there exist more important components. Specifically, we fail to reject the *null* hypothesis of the Kruskal–Wallis test; the corresponding $p-$ value is 0.4099.

Figure 9: The random internet network with SRLG capacities and partial connectivity regime: BIM results obtained using the $N = 1,000$ and the $N = 10,000$ sample sizes, and $N_{\text{sim}} = 100$.

In order to verify the result in Figure 9, we executed the PMC algorithm with $N = 100,000$ and $N = 1,000,000$ sample sizes, but failed to identify the most important components, too. We believe that in this case, it happens because there is a significant amount of freedom due to the partial connectivity constraint. The average CPU time for a single PMC run with $N = 100,000$ and $N = 1,000,000$ sample size is 12.818 and 123.34 seconds, respectively. Figure 10 depicts the corresponding results.



(a) The random internet graph (with capacities and partial connectivity) BIM results using $N = 100,000$ sample size. It is not clear if there exist more important components. Specifically, we fail to reject the *null* hypothesis of the Kruskal–Wallis test; the corresponding $p-$value is 0.839.

(b) The random internet graph (with capacities and partial connectivity) BIM results using $N = 1,000,000$ sample size. It is not clear if there exist more important components. Specifically, we fail to reject the *null* hypothesis of the Kruskal–Wallis test; the corresponding $p-$ value is 0.1378.

Figure 10: The random internet network with SRLG capacities and partial connectivity regime: BIM results obtained using the $N = 100,000$ and the $N = 1,000,000$, sample sizes, and $N_{\text{sim}} = 100$.

We proceed with the numerical evaluation of the PMC algorithm on the large real-life network.

4.4. The InterTubes network

In this case study, we benchmark the method performance on the 20 US-based service providers network from the *Internet Atlas Data Repository* `http://internetatlas.org` [45, 46]. This is a large real-life instance of US long-haul fiber-optic network with 273 vertices, 542 edges, and 542 SRLGs. Unfortunately, the sensitivity of this data does not allow us to share it. However, the data can be obtained by contacting the Internet Atlas Data Repository. We execute the PMC algorithm with the $N = 1,000$, and the $N = 10,000$ sample sizes, and $N_{\text{sim}} = 100$. The $N = 1,000$, and the $N = 10,000$ sample sizes are still sufficient for the $\widehat{\text{RE}}$ stability as indicated in Table 11.

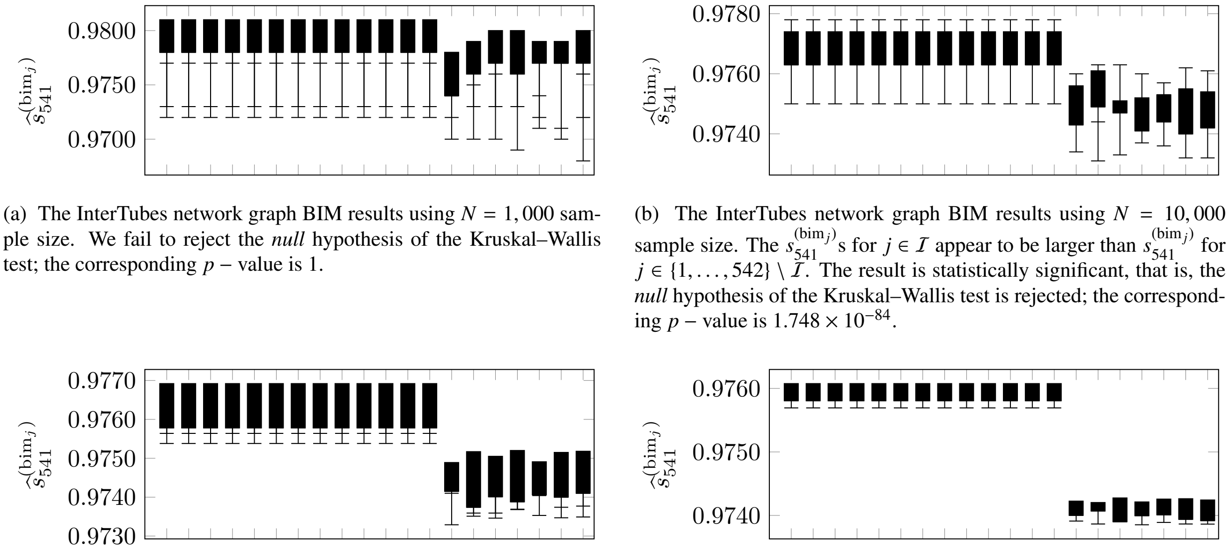
Table 11 The InterTubes network: average unreliability estimators and the corresponding $\widehat{\text{RE}}$s for various values of SRLG failure probability q. The average CPU time ($N_{\text{sim}} = 100$) for a single PMC run with $N = 1,000$ and $N = 10,000$ sample size is 0.5685 and 5.5304 seconds, respectively.

	$N = 1,000$		$N = 10,000$	
q	$\widehat{r}(G,\mathcal{R},q)$	$\widehat{\text{RE}}$	$\widehat{r}(G,\mathcal{R},q)$	$\widehat{\text{RE}}$
5×10^{-2}	6.333×10^{-1}	2.076×10^{-3}	6.306×10^{-1}	6.506×10^{-4}
1×10^{-2}	1.356×10^{-1}	6.418×10^{-3}	1.344×10^{-1}	2.213×10^{-3}
5×10^{-4}	6.608×10^{-3}	1.795×10^{-2}	6.513×10^{-3}	5.338×10^{-3}
1×10^{-4}	1.318×10^{-3}	2.021×10^{-2}	1.301×10^{-3}	5.981×10^{-3}
5×10^{-6}	6.588×10^{-5}	2.083×10^{-2}	6.502×10^{-5}	6.162×10^{-3}
1×10^{-6}	1.318×10^{-5}	2.086×10^{-2}	1.300×10^{-5}	6.170×10^{-3}
5×10^{-8}	6.588×10^{-7}	2.087×10^{-2}	6.502×10^{-7}	6.172×10^{-3}
1×10^{-8}	1.318×10^{-7}	2.087×10^{-2}	1.300×10^{-7}	6.172×10^{-3}
5×10^{-10}	6.588×10^{-9}	2.087×10^{-2}	6.502×10^{-9}	6.172×10^{-3}
1×10^{-10}	1.318×10^{-9}	2.087×10^{-2}	1.300×10^{-9}	6.172×10^{-3}
5×10^{-12}	6.588×10^{-11}	2.087×10^{-2}	6.502×10^{-11}	6.172×10^{-3}
1×10^{-12}	1.318×10^{-11}	2.087×10^{-2}	1.300×10^{-11}	6.172×10^{-3}
5×10^{-14}	6.588×10^{-13}	2.087×10^{-2}	6.502×10^{-13}	6.172×10^{-3}
1×10^{-14}	1.318×10^{-13}	2.087×10^{-2}	1.300×10^{-13}	6.172×10^{-3}
5×10^{-16}	6.588×10^{-15}	2.087×10^{-2}	6.502×10^{-15}	6.172×10^{-3}
1×10^{-16}	1.318×10^{-15}	2.087×10^{-2}	1.300×10^{-15}	6.172×10^{-3}
5×10^{-18}	6.588×10^{-17}	2.087×10^{-2}	6.502×10^{-17}	6.172×10^{-3}
1×10^{-18}	1.318×10^{-17}	2.087×10^{-2}	1.300×10^{-17}	6.172×10^{-3}
5×10^{-20}	6.588×10^{-19}	2.087×10^{-2}	6.502×10^{-19}	6.172×10^{-3}
1×10^{-20}	1.318×10^{-19}	2.087×10^{-2}	1.300×10^{-19}	6.172×10^{-3}

The algorithm manages to reveal the set of important SRLG components. The list of these SRLG indices is:

$$\mathcal{I} = \{55, 75, 99, 103, 141, 225, 289, 413, 419, 434, 469, 499, 529\}.$$

When trying to identify these components, we failed to obtain a statistically significant result with $N = 1,000$ sample size. Nevertheless, the $N = 10,000$ sample size was sufficient for this purpose. In order to verify the importance results, we also run the PMC algorithm with $N = 100,000$ and $N = 1,000,000$ sample sizes, too. The average CPU time for a single PMC run with $N = 100,000$ and $N = 1,000,000$ sample size is 54.192 and 528.63 seconds, respectively. Figure 11 depicts the corresponding results of the first twenty components with the largest $\widehat{s}_{541}^{(\mathrm{bim}_j)}$ values.



(a) The InterTubes network graph BIM results using $N = 1,000$ sample size. We fail to reject the *null* hypothesis of the Kruskal–Wallis test; the corresponding p – value is 1.

(b) The InterTubes network graph BIM results using $N = 10,000$ sample size. The $s_{541}^{(\mathrm{bim}_j)}$s for $j \in \mathcal{I}$ appear to be larger than $s_{541}^{(\mathrm{bim}_j)}$ for $j \in \{1, \ldots, 542\} \setminus \mathcal{I}$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p – value is 1.748×10^{-84}.

(c) The InterTubes network graph BIM results using $N = 100,000$ sample size. The $s_{541}^{(\mathrm{bim}_j)}$s for $j \in \mathcal{I}$ appear to be larger than $s_{541}^{(\mathrm{bim}_j)}$ for $j \in \{1, \ldots, 542\} \setminus \mathcal{I}$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p – value is about 0 (less than 1.748×10^{-84}).

(d) The InterTubes network graph BIM results using $N = 1,000,000$ sample size. The $s_{541}^{(\mathrm{bim}_j)}$s for $j \in \mathcal{I}$ appear to be larger than $s_{541}^{(\mathrm{bim}_j)}$ for $j \in \{1, \ldots, 542\} \setminus \mathcal{I}$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p – value is about 0 (less than 1.748×10^{-84}).

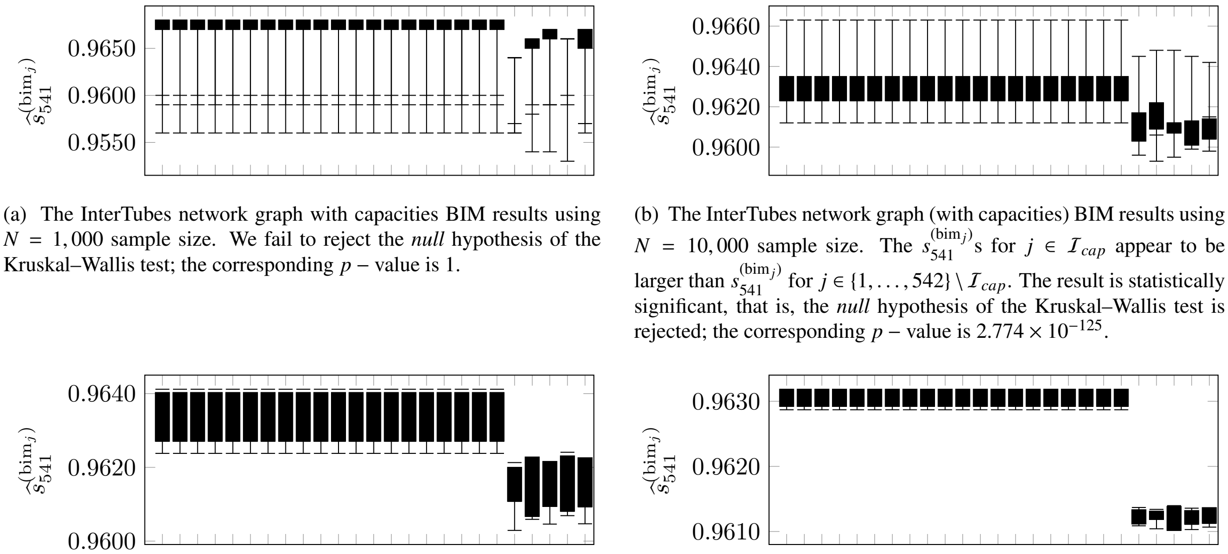
Figure 11: The InterTubes network: BIM results obtained using the $N = 1,000$, the $N = 10,000$, the $N = 100,000$ and the $N = 1,000,000$, sample sizes, and $N_{\mathrm{sim}} = 100$.

4.4.1. The InterTubes network with SRLG capacity regime

In order to test different failure regimes, we consider the InterTubes network again. The original data contains information about the number of fibers in each SRLG. Similar to the example in Section 4.3.1, we set the minimal capacity requirement to be 99%, and oblige the full connectivity. We expect that the system reliability will decrease, since there is a constraint on the minimal capacity of available fibers. We execute the PMC algorithm with the $N = 1,000$, and the $N = 10,000$ sample sizes, and $N_{\mathrm{sim}} = 100$. As indicated in Table 12, we still enjoy the $\widehat{\mathrm{RE}}$ stability. It is interesting to note that the number of important components grows in this case. Specifically, the set of indices of important BIMs is now: $\mathcal{I}_{cap} = \{6, 55, 59, 75, 82, 86, 99, 103, 117, 128, 141, 225, 289, 404, 413, 419, 434, 469, 499, 529\}$. As before, we managed to get a statistically significant result with $N = 10,000$, but not with $N = 1,000$ sample size. In order to verify this, we also run the experiment with $N = 100,000$ and $N = 1,000,000$ sample sizes. The average CPU time for a single PMC run with $N = 100,000$ and $N = 1,000,000$ sample size is 66.082 and 676.96 seconds, respectively. For all experiments with different sample sizes, we get the same set of the most important BIM components. See Figure 12, which depicts the corresponding BIM results for the first twenty five components with the largest $\widehat{s}_{541}^{(\mathrm{bim}_j)}$ values.

Table 12 The InterTubes network with SRLG capacity regime: average unreliability estimators and the corresponding $\widehat{\text{RE}}$s for various values of SRLG failure probability q. The average CPU time ($N_{\text{sim}} = 100$) for a single PMC run with $N = 1,000$ and $N = 10,000$ sample size is 0.6106 and 6.1561 seconds, respectively.

	$N = 1,000$		$N = 10,000$	
q	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$	$\widehat{r}(G, \mathcal{R}, q)$	$\widehat{\text{RE}}$
5×10^{-2}	9.994×10^{-1}	2.029×10^{-5}	9.994×10^{-1}	7.216×10^{-6}
1×10^{-2}	4.981×10^{-1}	2.052×10^{-3}	4.982×10^{-1}	7.010×10^{-4}
5×10^{-4}	1.186×10^{-2}	1.218×10^{-2}	1.175×10^{-2}	3.856×10^{-3}
1×10^{-4}	2.104×10^{-3}	1.499×10^{-2}	2.075×10^{-3}	4.738×10^{-3}
5×10^{-6}	1.019×10^{-4}	1.587×10^{-2}	1.003×10^{-4}	5.015×10^{-3}
1×10^{-6}	2.034×10^{-5}	1.591×10^{-2}	2.004×10^{-5}	5.027×10^{-3}
5×10^{-8}	1.017×10^{-6}	1.592×10^{-2}	1.002×10^{-6}	5.030×10^{-3}
1×10^{-8}	2.034×10^{-7}	1.592×10^{-2}	2.003×10^{-7}	5.030×10^{-3}
5×10^{-10}	1.017×10^{-8}	1.592×10^{-2}	1.002×10^{-8}	5.030×10^{-3}
1×10^{-10}	2.034×10^{-9}	1.592×10^{-2}	2.003×10^{-9}	5.030×10^{-3}
5×10^{-12}	1.017×10^{-10}	1.592×10^{-2}	1.002×10^{-10}	5.030×10^{-3}
1×10^{-12}	2.034×10^{-11}	1.592×10^{-2}	2.003×10^{-11}	5.030×10^{-3}
5×10^{-14}	1.017×10^{-12}	1.592×10^{-2}	1.002×10^{-12}	5.030×10^{-3}
1×10^{-14}	2.034×10^{-13}	1.592×10^{-2}	2.003×10^{-13}	5.030×10^{-3}
5×10^{-16}	1.017×10^{-14}	1.592×10^{-2}	1.002×10^{-14}	5.030×10^{-3}
1×10^{-16}	2.034×10^{-15}	1.592×10^{-2}	2.003×10^{-15}	5.030×10^{-3}
5×10^{-18}	1.017×10^{-16}	1.592×10^{-2}	1.002×10^{-16}	5.030×10^{-3}
1×10^{-18}	2.034×10^{-17}	1.592×10^{-2}	2.003×10^{-17}	5.030×10^{-3}
5×10^{-20}	1.017×10^{-18}	1.592×10^{-2}	1.002×10^{-18}	5.030×10^{-3}
1×10^{-20}	2.034×10^{-19}	1.592×10^{-2}	2.003×10^{-19}	5.030×10^{-3}



(a) The InterTubes network graph with capacities BIM results using $N = 1,000$ sample size. We fail to reject the *null* hypothesis of the Kruskal–Wallis test; the corresponding p – value is 1.

(b) The InterTubes network graph (with capacities) BIM results using $N = 10,000$ sample size. The $s_{541}^{(\text{bim}_j)}$s for $j \in \mathcal{I}_{cap}$ appear to be larger than $s_{541}^{(\text{bim}_j)}$ for $j \in \{1, \ldots, 542\} \setminus \mathcal{I}_{cap}$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p – value is 2.774×10^{-125}.

(c) The InterTubes network graph (with capacities) BIM results using $N = 100,000$ sample size. The $s_{541}^{(\text{bim}_j)}$s for $j \in \mathcal{I}_{cap}$ appear to be larger than $s_{541}^{(\text{bim}_j)}$ for $j \in \{1, \ldots, 542\} \setminus \mathcal{I}_{cap}$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p – value is about 0 (less then 2.774×10^{-125}).

(d) The InterTubes network graph (with capacities) BIM results using $N = 1,000,000$ sample size. The $s_{541}^{(\text{bim}_j)}$s for $j \in \mathcal{I}_{cap}$ appear to be larger than $s_{541}^{(\text{bim}_j)}$ for $j \in \{1, \ldots, 542\} \setminus \mathcal{I}_{cap}$. The result is statistically significant, that is, the *null* hypothesis of the Kruskal–Wallis test is rejected; the corresponding p – value is about 0 (less then 2.774×10^{-125}).

Figure 12: The InterTubes network with SRLG capacity regime: BIM results obtained using the $N = 1,000$, the $N = 10,000$, the $N = 100,000$, and the $N = 1,000,000$ sample sizes, and $N_{\text{sim}} = 100$.

Finally, we consider an example, for which, the PMC algorithm fails.

4.5. The InterTubes network with SRLG capacities and partial connectivity regime

Again, we utilize the InterTubes network with capacity constraint set to 80%, and similar to Section 4.3.2, we allow operating within partial connectivity of 92.5%. We run the PMC algorithm with $N = 1,000$, $N = 10,000$, $N = 100,000$, and $N = 1,000,000$ sample sizes, where the average CPU time for a single PMC run is 0.6292, 6.2366, 62.571, and 630.41 seconds, respectively. However, for this case study, the $\widehat{\text{RE}}$ of the reliability estimator is not stable. Figure 13 shows a typical behavior of the reliability estimator and the corresponding $\widehat{\text{RE}}$. The problem is that there exist rare events in spectra components, and these cannot be adequately estimated with PMC (see Section 3.2). These components should be estimated via advanced variance minimization techniques, namely, using importance sampling or multilevel splitting approach. For a detailed discussion, we refer to Vaisman et. al. [35]. Naturally, enabling the PMC algorithm to handle small spectra components is an important direction of the future research.

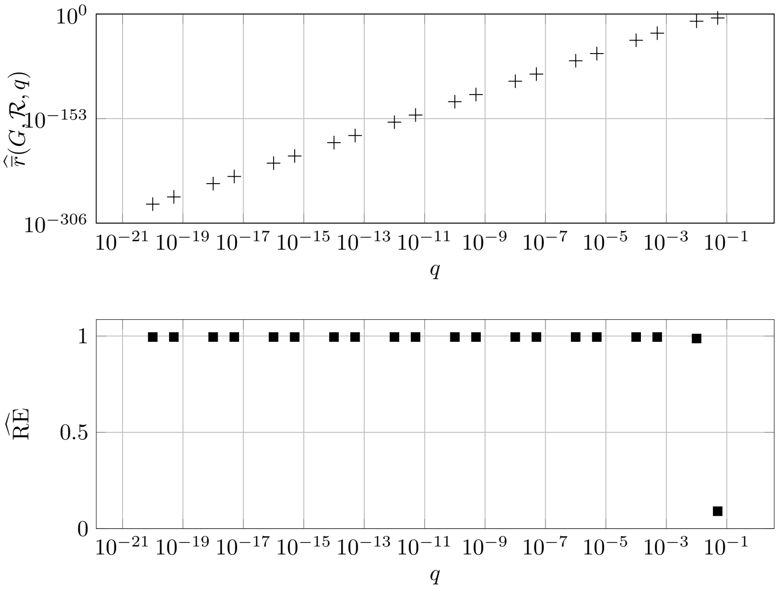


Figure 13: The InterTubes network with SRLG capacities and partial connectivity regime: a typical behavior of unreliability estimators and the corresponding $\widehat{\text{RE}}$s for various values of SRLG failure probability q with the $N = 100,000$ sample size.

5. Conclusion

In this paper, we introduced a computationally efficient method which is suitable for the analyses of reliability and importance measure of complex real-life networks under the shared risk link group failure scenario. The proposed approach is both theoretically and computationally sound. In particular, we showed that the Permutation Monte Carlo method is easy to implement and that it is very useful for assessing system reliability and importance measure in the sense that it can handle real-life problem instances. We also showed that a statistical approach for handling network importance measure can resolve the problem of the corresponding estimator instability. One of the most pivotal limitations of the proposed method is that it is not suitable for handling a scenario in which the failure probability of each shared risk link group component is not the same. While several procedures for this setting are available, they do not scale well for large problems. In addition, the Permutation Monte Carlo algorithm is not suitable for an adequate estimation of spectra that involve extremely small components. In this case, one needs to resort to different variance minimization methods such as importance sampling and multilevel splitting. However, when applying such methods, we need to be aware of the corresponding scale issues since these algorithms are generally more computationally intensive. As for the future research, we believe that the following directions are of interest.

1. It is important from both the theoretical and practical points of view, to identify classes of networks for which the spectra approach is a Fully Polynomial Randomized Approximation Scheme. Namely, given specific network topology, the task is to present an analysis in the spirit of Theorem 1, for both the reliability and the importance measure estimators.
2. In this paper, we used an ANOVA-like test to identify the set of important components. While this approach proves to be useful in practice, it has the limitation that it only considers the last spectra component (which is not equal to 1). We believe that it would be important to examine a more refined statistical framework for identifying important components. Such framework will involve a multiple hypothesis testing and consider all spectra components.
3. While in this paper we only handle shared risk link groups, it is straightforward to extend the work to the setting of shared risk vertices in optical mesh networks and shared risk equipment with multi-port network within vertex groups. The spectra invariant and the corresponding Permutation Monte Carlo machinery remains valid, and these extensions are of great practical importance.
4. We showed several computational experiments that involve different network failure regimes. However, it is of interest to perform an extensive test of the robustness of the proposed algorithmic method under additional important failure regime settings.
5. This is of great interest to extend the spectra ideas beyond the reliability context. In this paper, we considered a graph connectivity function, which is just a special case of a general Boolean function. The application of the proposed method to general Boolean functions in different domains is a very promising research direction. However, for a general Boolean function, one might require a redefinition of the corresponding spectra object and in particular the $\mathcal{A}_i$s and the $\mathcal{B}_i$s sets (please see the beginning of Section 3).
6. Finally, while we provide a single-threaded implementation of the Permutation Monte Carlo algorithm, from the practical point of view, it would be of great interest to develop a parallel software implementation that is capable of running on multiple CPUs or a GPU. Since the Permutation Monte Carlo algorithm is easy to parallelize, the corresponding task is both feasible and beneficial for the analysis of various networks (such as optical communication networks, transportation networks, power grids, etc.), with many shared risk link groups and a large number of vertices and links. Such software will allow practitioners to handle even larger real-life network instances with several thousands of vertices, edges, and shred risk groups.

Acknowledgment

We are thoroughly grateful to the Editor and anonymous Reviewers for their valuable and constructive remarks and suggestions. We would like to thank Professor Matthew Roughan for many helpful suggestions regarding shared risk group setting and the corresponding real-life applications in complex networks, and to Professor Paul Barford for sharing the InterTubes long haul infrastructure data set. This work was supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers, under CE140100049 grant number.

References

[1] T. Koch, I. Kaminow, Optical fiber telecommunications III, Academic Press, San Diego, Calif., 1997.

[2] U. D. of Transportation, Rffects of catastrophic events on transportation system management and operations howard street tunnel fire baltimore city, maryland july 18, 2001 final report: findings, Tech. rep., U.S. Department of Transportation, Washington, D.C. (July 2002).

[3] J. Strand, A. Chiu, R. Tkach, Issues for routing in the optical layer, IEEE Communications Magazine 39 (2) (2001) 81–87.

[4] S. Si, G. Levitin, H. Dui, S. Sun, Importance analysis for reconfigurable systems, Reliability engineering and system safety 126 (2014) 72–80.

[5] S. Si, G. Levitin, H. Dui, S. Sun, Component state-based integrated importance measure for multi-state systems, Reliability engineering and system safety 116 (2013) 75–83.

[6] Z. Zhili, L. Tachun, K. Thulasiraman, Survivable cloud network design against multiple failures through protecting spanning trees, Journal of Lightwave Technology 35 (2) (2017) 288–298.

[7] L. Liu, B. Li, B. Qi, X. Ye, Y. Sun, S. Tian, C. Zhu, P. Xi, Optimization of communication capacity for load control considering shared risk link group in source-grid-load system, International journal of electrical power & energy systems 122 (2020) 106166.

[8] J. Tapolcai, L. Ronyai, B. Vass, L. Gyimothi, Fast enumeration of regional link failures caused by disasters with limited size, IEEE/ACM transactions on networking (2020) 1–14.

[9] M. Garey, D. Johnson, Computers and intractability : a guide to the theory of NP-completeness, Series of books in the mathematical sciences, W.H. Freeman, San Francisco, 1979.

[10] Q. Jian, Diverse routing in optical mesh networks, IEEE Transactions on Communications 51 (3) (2003) 489–494.

[11] S. Lu, Y. Xi, B. Ramamurthy, Shared risk link group (srlg)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks, IEEE/ACM Transactions on Networking 13 (4) (2005) 918–931.

[12] Z. Qingfu, S. Jianyong, X. Gaoxi, E. Tsang, Evolutionary algorithms refining a heuristic: A hybrid method for shared-path protections in WDM networks under srlg constraints, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37 (1) (2007) 51–61.

[13] J. Zheng, H. Okamura, T. Pang, T. Dohi, Availability importance measures of components in smart electric power grid systems, Reliability engineering and system safety 205 (2021) 107164.

[14] X. Lu, P. Baraldi, E. Zio, A data-driven framework for identifying important components in complex systems, Reliability engineering and system safety 204 (2020) 107197.

[15] A. Lisnianski, I. Frenkel, L. Khvatskin, On Birnbaum importance assessment for aging multi-state system under minimal repair by using the L_Z-transform method, Reliability engineering and system safety 142 (2015) 258–266.

[16] A. Lisnianski, E. Levit, L. Teper, Short-term availability and performability analysis for a large-scale multi-state system based on robotic sensors, Reliability engineering and system safety 205 (2021) 107206.

[17] H. Dui, S. Si, R. C. Yam, Importance measures for optimal structure in linear consecutive-k-out-of-n systems, Reliability engineering and system safety 169 (2018) 339–350.

[18] P. Do, C. Bérenguer, Conditional reliability-based importance measures, Reliability engineering and system safety 193 (2020) 106633.

[19] Y. Shi, W. Zhu, Y. Xiang, Q. Feng, Condition-based maintenance optimization for multi-component systems subject to a system reliability requirement, Reliability engineering and system safety 202 (2020) 107042.

[20] F. Bistouni, M. Jahanshahi, Reliability-aware ring protection link selection in Ethernet ring mesh networks, Reliability engineering and system safety 191 (2019) 106575.

[21] P.-C. Chang, Y.-K. Lin, Y.-M. Chiang, System reliability estimation and sensitivity analysis for multi-state manufacturing network with joint buffers — a simulation approach, Reliability engineering and system safety 188 (2019) 103–109.

[22] I. Gertsbakh, Reliability Theory: With Applications to Preventive Maintenance, 1st Edition, Springer-Verlag, Berlin Heidelberg, 2000.

[23] I. Gertsbakh, Y. Shpungin, Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo, CRC Press, Inc., Boca Raton, FL, USA, 2009.

[24] L. Valiant, The complexity of enumeration and reliability problems, SIAM Journal on Computing 8 (3) (1979) 410–421.

[25] R. Vaisman, D. Kroese, I. Gertsbakh, Splitting sequential Monte Carlo for efficient unreliability estimation of highly reliable networks, Structural Safety 63 (C) (2016) 1–10.

[26] R. Karp, M. Luby, Monte-Carlo algorithms for the planar multiterminal network reliability problem, Journal of Complexity 1 (1) (1985) 45–64.

[27] D. Karger, A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem, SIAM Review 43 (3) (2001) 499–522.

[28] A. Satyanarayana, R. Wood, A linear-time algorithm for computing k-terminal reliability in series-parallel networks, SIAM Journal on Computing 14 (4) (1985) 818–832.

[29] R. Zenklusen, M. Laumanns, High-confidence estimation of small s-t reliabilities in directed acyclic networks, Networks 57 (4) (2011) 376–388.

[30] I. Gertsbakh, Y. Shpungin, Combinatorial approach to computing component importance indexes in coherent systems, Probability in the Engineering and Informational Sciences 26 (2012) 117–128.

[31] M. Finkelstein, I. Gertsbakh, R. Vaisman, On a single discrete scale for preventive maintenance with two shock processes affecting a complex system, Applied Stochastic Models in Business and Industry 33 (1) (2017) 54–62.

[32] M. Finkelstein, I. Gertsbakh, 'time-free' preventive maintenance of systems with structures described by signatures, Applied Stochastic Models in Business and Industry 31 (6) (2015) 836–845.

[33] Z. Birnbaum, On the importance of different components in a multicomponent system, in: P. Krishnaiah (Ed.), Multivariate Analysis II, Academic Press, New York, 1969, pp. 581–592.

[34] P. Miziula, J. Navarro, Birnbaum importance measure for reliability systems with dependent components, IEEE Transactions on Reliability 68 (2) (2019) 439–450.

[35] R. Vaisman, D. Kroese, I. Gertsbakh, Improved sampling plans for combinatorial invariants of coherent systems, IEEE Transactions on Reliability 65 (1) (2016) 410–424.

[36] F. Coolen, T. Coolen-Maturi, The structure function for system reliability as predictive (imprecise) probability, Reliability engineering and system safety 154 (C) (2016) 180–187.

[37] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms (3. ed.), MIT Press, Cambridge, Mass., 2009.

[38] R. Rubinstein, A. Ridder, R. Vaisman, Fast Sequential Monte Carlo Methods for Counting and Optimization, John Wiley and Sons, New York, 2013.

[39] F. Samaniego, System Signatures and Their Applications in Engineering Reliability, International Series in Operations Research & Management Science, Springer, 2007.

[40] F. P. A. Coolen, T. Coolen-Maturi, Generalizing the signature to systems with multiple types of components, in: Complex Systems and Dependability, Advances in Intelligent and Soft Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 115–130.

[41] R. Tarjan, Efficiency of a good but not linear set union algorithm, Journal of the ACM (JACM) 22 (2) (1975) 215–225.

[42] R. Tarjan, A class of algorithms which require nonlinear time to maintain disjoint sets, Journal of Computer and System Sciences 18 (2) (1979) 110–127.

[43] M. Mitzenmacher, E. Upfal, Probability and computing: randomized algorithms and probabilistic analysis, Cambridge University Press, New York, 2005.

[44] A. Elmokashfi, A. Kvalbein, C. Dovrolis, On the scalability of bgp: The role of topology growth, IEEE Journal on Selected Areas in Communications 28 (8) (2010) 1250–1261.

[45] R. Durairajan, S. Ghosh, X. Tang, P. Barford, B. Eriksson, Internet atlas: A geographic database of the internet, in: Proceedings of the 5th

ACM Workshop on HotPlanet, HotPlanet 13, Association for Computing Machinery, New York, NY, USA, 2013, p. 15–20.

[46] R. Durairajan, P. Barford, J. Sommers, W. Willinger, Intertubes: A study of the us long-haul fiber-optic infrastructure, SIGCOMM Comput. Commun. Rev. 45 (4) (2015) 565–578.

[47] R. Rubinstein, D. Kroese, Simulation and the Monte Carlo Method, 3rd Edition, John Wiley & Sons, New York, 2017.

[48] W. Kruskal, W. Wallis, Use of ranks in one-criterion variance analysis, Journal of the American Statistical Association 47 (260) (1952) 583–621.