

New Sampling Plans for Estimating Residual Connectedness Reliability

Rohan Shah

School of Mathematics and Physics
The University of Queensland,
Brisbane, Australia
rohan.shah@uqconnect.edu.au

Radislav Vaisman

School of Mathematics and Physics
The University of Queensland,
Brisbane, Australia
r.vaisman@uq.edu.au

Abstract—Residual connectedness reliability plays an important role in the analysis of Radio Broadcast and Mobile Ad hoc networks. In this paper we focus on binary systems with identical component reliabilities. The reliability of such systems depends solely on an invariant called the spectra. Calculation of the spectra is considered to be computationally hard, since it is equivalent to the problem of counting the number of connected subgraphs of every size. As a consequence one has to rely on approximation techniques. However for a general network the efficiency of such methods depends on their ability to handle rare-event probability estimation. To cope with this challenge, we study a combination of two Monte Carlo methods. Namely, a well-known Permutation Monte Carlo and the recently pioneered Stochastic Enumeration algorithm. In particular, we show that the former is able to solve easier problems while providing provable theoretical performance guarantees to the delivered estimator, and the latter can successfully handle hard problems under the rare-event setting. Finally, our numerical results imply that these algorithms can provide fast and reliable estimators to the residual connectivity problem.

Index Terms—Residual connectedness reliability, System structure function, Permutation Monte Carlo, Stochastic Enumeration, Rare Events.

I. INTRODUCTION

Residual connectedness reliability (RCR) is an important problem from both theoretical and practical points of view. In particular, this problem is NP-hard [1] and it has important applications in Radio Broadcast and Mobile Ad hoc networks [2], [3].

The general RCR problem can be defined as follows. Suppose we are given an undirected graph $G = G(V, E)$, where V and E are the vertex and edge sets respectively. We assume that the edges are absolutely reliable but that the vertices are subject to failure with probability $q_v = 1 - p_v$, for each $v \in V$. We further assume that the vertices fail independently at random. If a vertex fails then all incident edges become non-operational. The network is defined to be in an operational state if the surviving vertices induce a connected graph. The residual node connectedness reliability

$r(G)$ is defined as the probability that the network G is in the operational state [1].

Network reliability research is mostly concerned with the so-called K -terminal network reliability (KTR). See for example [4], [5], [6], [7], [8], [9], [10], [11], [12]. However there is a relatively small number of works dedicated to the RCR problem. These works mostly consider specific graph topologies for which rigorous results can be obtained [13], [14], [15]. There has been very limited work done on the RCR problem for general graphs [16], [17].

In this paper we consider the special but important case of the RCR problem where all the failure probabilities are equal, namely $q_v = q$ for all $v \in V$. In this case the RCR is given by $r(G, q)$. A similar setting in the context of KTR was considered in [11], where the authors used a well-known combinatorial system invariant called the spectra [4], [5], [6], [18], to deliver fast and reliable estimators of the KTR. We consider this combinatorial object in detail in Section II, but it is important to bear in mind that the spectra only depends on the graph topology and is hence a system invariant [19]. As soon as the spectra is available, one can immediately obtain the corresponding network reliability in linear time (in the number of vertices), for any vertex failure probability $q \in (0, 1)$. An additional merit of having the spectra in hand is the possibility to perform network sensitivity analysis. See [11] and [20] for details.

Unfortunately, the exact spectra is generally not available analytically, since its calculation problem belongs to the #P-complete complexity class [1], [21]. Hence, it is usually estimated via Permutation Monte Carlo (PMC) [4]. However, the PMC method was shown to be inefficient under the rare-event setting [10], [11]. To overcome this problem, we follow the ideas of [11] and adapt the recently pioneered tree counting method called Stochastic Enumeration (SE) [22], [23] for the estimation of the RCR spectra. Although the RCR spectra does not enjoy the monotonicity properties of the KTR [4], [18], we show that the spectra approach is useful for RCR estimation.

The rest of this paper is organized as follows. In Section II we define the system structure function (the spectra) and demonstrate its importance for estimating the RCR. In Section III we rigorously analyze the efficiency of the PMC estimator and discuss the rare-event phenomena. In Section IV we show

This work was supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers, under grant number CE140100049.

that the spectra estimation problem can be reduced to the tree counting problem and give a brief overview of the SE algorithm for counting trees. Section V provides numerical evidence for the accuracy of the proposed method. Finally, in Section VI we summarize our findings and discuss possible directions for future research.

II. THE SYSTEM STRUCTURE FUNCTION

To start with, consider an undirected graph $G = G(V, E)$, $|V| = n$, and for $k = 0, \dots, n$ let \mathcal{A}_k be the set of all subgraphs of G induced by a vertex subset of size k . Let

$$\mathcal{B}_k = \left\{ G' = G'(V', E') \mid G' \in \mathcal{A}_k, \forall v, u \in G' : u \overset{\sim}{\underset{(G')}{\rightsquigarrow}} v \right\},$$

be the subset of \mathcal{A}_k , such that each graph in \mathcal{B}_k forms a single connected component. The RCR is given by

$$r(G, q) = \sum_{k=0}^n |\mathcal{B}_k| p^k q^{n-k}, \quad (1)$$

where $q = 1 - p$ is the vertex failure probability. Similar to [18], we define the RCR spectra as follows.

Definition 2.1 (RCR spectra): Given an undirected graph $G = G(V, E)$, $|V| = n$, the RCR spectra is defined by

$$\mathbf{s}(G) = \{s_0, \dots, s_n\},$$

where $s_k = |\mathcal{B}_k|/|\mathcal{A}_k|$ for $k = 0, \dots, n$. \square

Note that since $|\mathcal{A}_k| = \binom{n}{k}$ for $k = 0, \dots, n$, (1) can be written as:

$$r(G, q) = \sum_{k=0}^n \binom{n}{k} s_k p^k q^{n-k}. \quad (2)$$

Moreover, since $|\mathcal{A}_k|$ and $|\mathcal{B}_k|$ are clearly independent of the vertex failure probability q for $k = 0, \dots, n$, $\mathbf{s}(G)$ is a system invariant. That is, as soon as the spectra is available, one can calculate the RCR via (2), in $\mathcal{O}(n)$ time for any $q \in (0, 1)$.

In order to understand the potential benefit of using Definition 2.1 and the corresponding equation (2), we consider a simple but illustrative example, that demonstrates the surprising power of the spectra.

Example 2.1 (Simple example): Consider the simple network in Fig. 1. It is not very hard to see that $|\mathcal{B}_0| = 1, |\mathcal{B}_1| = 3, |\mathcal{B}_2| = 1$ and $|\mathcal{B}_3| = 0$. Hence, from Definition 2.1, the spectra of this simple network G is given by:

$$\mathbf{s}(G) = \{s_0, s_1, s_2, s_3\} = \{1, 1, 1/3, 0\}. \quad (3)$$

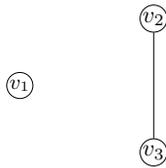


Fig. 1. Simple network with 3 vertices.

Combining the obtained spectra with (2), we arrive at the analytical expression of the network reliability:

$$\begin{aligned} r(G, q) &= \sum_{k=0}^3 \binom{3}{k} s_k p^k q^{3-k} = \binom{3}{0} 1 q^3 + \binom{3}{1} 1 p q^2 \\ &\quad + \binom{3}{2} \frac{1}{3} p^2 q + \binom{3}{3} 0 p^3 = q^3 + 3p q^2 + p^2 q. \end{aligned}$$

Fig. 2 demonstrates the exact reliability of this simple network as a function of q . Note that when q approaches 1, all

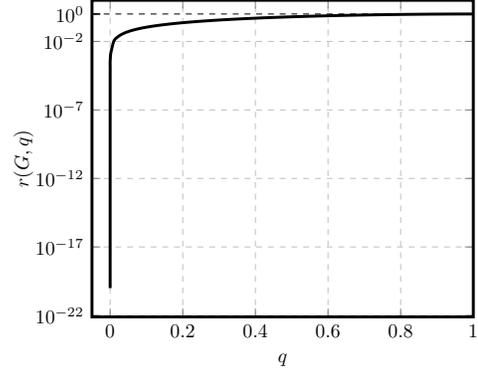


Fig. 2. The Simple Network RCR as a function of q .

the vertices are failed with high probability, so the RCR approaches 1. On the other hand, when q approaches 0, all the graph vertices are present with high probability and the network becomes disconnected since v_1 is not connected to v_2 and v_3 .

Suppose now that we would like to estimate the RCR using a Crude Monte Carlo (CMC) procedure. To do so, we toss a biased coin for each vertex, such that $\mathbb{P}(\text{vertex fail}) = q$, delete the failed vertices from G , and check the resulting subgraph connectivity. This trivial experiment results in an unbiased estimator of $r(G, q)$, but unfortunately for small values of $r(G, q)$ the CMC is known to fail, due to the rare-event setting [24]. As a rule of thumb one should not consider the CMC algorithm for estimation of probabilities that are smaller than 10^{-7} .

Now, let us turn our attention to the spectra components in (3). Note that these components are not small. This, and the above discussion, imply that one can design an efficient CMC experiment to estimate the spectra (and use (2)), instead of estimating $r(G, q)$ directly. Indeed, to estimate the component s_k for $k = 0, 1, 2, 3$, we can choose k vertices uniformly at random from G and check if they are connected. Performing this experiment with $N = 10^5$ sample size, we estimated the empirical spectra to be $\hat{\mathbf{S}}(G) = \{1, 1, 0.3347, 0\}$. The corresponding reliability estimator $\hat{r}(G, q)$ is derived via (2) and we arrive at

$$\begin{aligned} \hat{r}(G, q) &= \sum_{k=0}^3 \binom{3}{k} \hat{s}_k p^k q^{3-k} \\ &= q^3 + p q^2 + 3p q^2 + 1.0041 p^2 q. \end{aligned}$$

We conclude the example by noting that in this case, the absolute error (AER) [24], of the empirical reliability $\widehat{r}(G, q)$ is at most 0.41% for any q , since it holds that:

$$\begin{aligned} \text{AER} &= \frac{|\widehat{r}(G, q) - r(G, q)|}{|r(G, q)|} = \frac{|1 - 1.0041|p^2q}{q^3 + pq^2 + 3pq^2 + p^2q} \\ &\leq \frac{0.0041p^2q}{p^2q} = 0.0041. \end{aligned}$$

The CMC estimator presented in Example 2.1 can be generalized to the well-known PMC algorithm [4]. In the next section, we present a detailed analysis of the PMC method and consider its potential pitfalls.

III. THE PMC ALGORITHM

The basic idea of the PMC algorithm for RCR spectra estimation is a straightforward generalization of the KTR procedure [4]. The RCR PMC method is summarized in Algorithm 3.1.

Algorithm 3.1 (PMC Algorithm): Given $G = G(V, E)$, $V = \{v_1, \dots, v_n\}$, and a sample size N , execute the following steps.

- 1) **(Initialization):** Set $t \leftarrow 0$ and $\widehat{s}_k \leftarrow 0$, $k = 0, \dots, n$.
- 2) **(Permutation generation):** Set $t \leftarrow t + 1$ and generate a permutation $\mathbf{\Pi} = (\Pi_1, \dots, \Pi_n)$ of $\{1, \dots, n\}$, uniformly at random. Set $k \leftarrow 0$.
- 3) **(Connectivity check):** Let the graph $G' = G(V', E')$, $V' = \{v_{\Pi_1}, \dots, v_{\Pi_k}\}$, $E' = \{(u, v) \in E \mid u, v \in V'\}$ be the subgraph of G induced by V' . If G' is connected, that is, for all $u, v \in V'$,

$$u \overset{(G')}{\rightsquigarrow} v,$$

holds, set $\widehat{s}_k \leftarrow \widehat{s}_k + 1/N$. Repeat this step for $k = 1, \dots, n$.

- 4) **(Terminal position?):** If $t = N$, go to Step 5, else, go to Step 2.
- 5) **(Estimator):** Output $\widehat{\mathbf{S}} = \{\widehat{s}_0, \widehat{s}_1, \dots, \widehat{s}_n\}$.

Example 3.1 (PMC for the simple graph): Consider again the simple graph example from Fig. 1. Table I summarizes all possible permutations over the graph's vertices. Ones and zeros stand for the corresponding subgraph connectivity and disconnectedness, respectively. For example, consider the S_2 entry for $\pi = (2, 3, 1)$. Since v_2 and v_3 induce a connected subgraph, the entry is set to one. On the other hand, the S_2 entry of $\pi = (3, 1, 2)$ is set to zero, because v_3 and v_1 induce a disconnected subgraph. Note that there are 6 possible

TABLE I
SIMPLE GRAPH'S VERTICES PERMUTATIONS

π	vertex permutation	S_0	S_1	S_2	S_3
$\pi = (1, 2, 3)$	(v_1, v_2, v_3)	1	1	0	0
$\pi = (1, 3, 2)$	(v_1, v_3, v_2)	1	1	0	0
$\pi = (2, 1, 3)$	(v_2, v_1, v_3)	1	1	0	0
$\pi = (2, 3, 1)$	(v_2, v_3, v_1)	1	1	1	0
$\pi = (3, 1, 2)$	(v_3, v_1, v_2)	1	1	0	0
$\pi = (3, 2, 1)$	(v_3, v_2, v_1)	1	1	1	0

permutations, and thus, it is not very hard to derive the spectra. In particular, we arrive at: $\widehat{\mathbf{S}} = \{6/6, 6/6, 2/6, 0/6\} = \{1, 1, 1/3, 0\}$.

We next show that Algorithm 3.1 provides an unbiased estimator. Consider a random variable $S_k(\mathbf{\Pi})$ for any $k = 0, \dots, n$ satisfying:

$$S_k(\mathbf{\Pi}) = \begin{cases} 1 & \text{if } \{v_{\Pi_1}, \dots, v_{\Pi_k}\} \text{ induces a connected subgraph,} \\ 0 & \text{otherwise.} \end{cases}$$

Note that for each connected subgraph $G' \in \mathcal{B}_k$, there exist $k!(n-k)!$ corresponding vertex permutations. Combining this with the fact that there are $n!$ possible permutations over the $\{1, \dots, n\}$ set, we arrive at:

$$\begin{aligned} \mathbb{E}(S_k) &= \sum_{\pi} S_k(\pi) \mathbb{P}(\pi) = \frac{1}{n!} \sum_{\pi} S_k(\pi) \\ &= \frac{1}{n!} |\mathcal{B}_k| k!(n-k)! = |\mathcal{B}_k| \binom{n}{k}^{-1} = s_k. \end{aligned} \quad (4)$$

Although unbiasedness is certainly a desired property, we should also consider the efficiency of Algorithm 3.1. Theorem 3.1 shows that under some conditions, the PMC algorithm is a fully polynomial randomized approximation scheme (FPRAS) for the RCR problem. The FPRAS is considered to be a very efficient algorithm and it is essential the best result one can hope to achieve for this #P-complete problem. To prove that a randomized algorithm is an FPRAS, it is sufficient to show that its coefficient of variation (CV) is bounded by a polynomial in the algorithm input size.

Theorem 3.1 (PMC efficiency): Given a graph $G = (V, E)$, $|V| = n$, the spectra estimator $\widehat{\mathbf{S}} = \{\widehat{s}_0, \dots, \widehat{s}_n\}$ obtained via PMC Algorithm 3.1, and the RCR estimator $\widehat{r}(G, q)$ from (2), it holds that

$$\text{CV} = \mathbb{E}(\widehat{r}(G, q)^2) / \mathbb{E}(\widehat{r}(G, q))^2 \leq n\mathcal{P}(n)^2,$$

provided that the minimal non-zero component of $\mathbf{s}(G)$ is at least equal to $1/\mathcal{P}(n)$, where $\mathcal{P}(n)$ is a polynomial in n .

Proof: The proof is by a straightforward bounding of the $\widehat{r}(G, q)$ moments. In particular, from (2), we know that:

$$\widehat{r}(G, q) = \sum_{k=0}^n \binom{n}{k} \widehat{s}_k p^k q^{n-k}. \quad (5)$$

Combining the above equation and the unbiasedness result (4), we arrive at:

$$\begin{aligned} \mathbb{E}(\widehat{r}(G, q)) &= \mathbb{E}\left(\sum_{k=0}^n \binom{n}{k} \widehat{s}_k p^k q^{n-k}\right) \\ &= \sum_{k=0}^n \binom{n}{k} \mathbb{E}(S_k(\mathbf{\Pi})) p^k q^{n-k} \\ &\stackrel{(4)}{=} \sum_{k=0}^n \binom{n}{k} s_k p^k q^{n-k} = r(G, q). \end{aligned} \quad (6)$$

In order to upper bound the second moment of $\widehat{r}(G, q)$, let \underline{s} and \bar{s} be the minimal non zero and the maximal components

of $s(G)$, respectively. With these definitions in hand, it holds that:

$$\begin{aligned}
\mathbb{E}(\widehat{r}(G, q)^2) &= \mathbb{E}\left(\left(\sum_{k=0}^n \binom{n}{k} \widehat{s}_k p^k q^{n-k}\right)^2\right) \quad (7) \\
&\leq \mathbb{E}\left(n \cdot \sum_{k=0}^n \binom{n}{k}^2 \widehat{s}_k^2 p^{2k} q^{2(n-k)}\right) \\
&\stackrel{\text{(Jensen's inequality [25])}}{\leq} \mathbb{E}\left(n \cdot \sum_{k=0}^n \binom{n}{k}^2 \mathbb{E}(S_k(\mathbf{\Pi})^2) p^{2k} q^{2(n-k)}\right) \\
&= n \sum_{k=0}^n \binom{n}{k}^2 \mathbb{E}(S_k(\mathbf{\Pi})^2) p^{2k} q^{2(n-k)} \\
&\leq n \sum_{k=0}^n \binom{n}{k}^2 \bar{s} p^{2k} \cdot q^{2(n-k)} \\
&\stackrel{(*)}{\leq} n \sum_{k=0}^n \binom{n}{k}^2 \bar{s} p^{2k} \cdot q^{2(n-k)} \\
&= n \bar{s} \left(\sum_{k=0}^n \binom{n}{k}^2 p^{2k} q^{2(n-k)}\right),
\end{aligned}$$

where $(*)$ holds since $S_k \sim \text{Bernoulli}(s_k)$, so, $\mathbb{E}(S_k(\mathbf{\Pi})^2) = s_k$, thus $\mathbb{E}(S_k(\mathbf{\Pi})^2) \leq \bar{s} \leq 1$. Combining (6) and (7) yields

$$\begin{aligned}
\text{CV} &= \frac{\mathbb{E}(\widehat{r}(G, q)^2)}{\mathbb{E}(\widehat{r}(G, q))^2} \leq \frac{n \bar{s} \left(\sum_{k=0}^n \binom{n}{k}^2 p^{2k} q^{2(n-k)}\right)}{\left(\sum_{k=0}^n \binom{n}{k} s_k p^k q^{n-k}\right)^2} \\
&\leq \frac{n \bar{s} \left(\sum_{k=0}^n \binom{n}{k}^2 p^{2k} q^{2(n-k)}\right)}{\left(\sum_{k=0}^n \binom{n}{k} \underline{s} p^k q^{n-k}\right)^2} \\
&\leq \frac{n \bar{s} \left(\sum_{k=0}^n \binom{n}{k}^2 p^{2k} q^{2(n-k)}\right)}{\underline{s}^2 \left(\sum_{k=0}^n \binom{n}{k} p^k q^{n-k}\right)^2} \\
&\leq \frac{n \bar{s} \left(\sum_{k=0}^n \binom{n}{k}^2 p^{2k} q^{2(n-k)}\right)}{\underline{s}^2 \left(\sum_{k=0}^n \binom{n}{k} p^{2k} q^{2(n-k)}\right)} \\
&\leq \frac{n \bar{s}}{\underline{s}^2} \leq \frac{n}{\underline{s}^2} \leq \frac{n}{(\underline{s} \geq 1/\mathcal{P}(n))} \leq n \mathcal{P}(n)^2,
\end{aligned}$$

thus, completing the proof. \square

Theorem 3.1 opens a way to identify easy problems, in the sense of their reliability estimation. Below we give one such example called the dodecahedron network. This example is widely used as a benchmark in reliability studies.

Example 3.2 (The dodecahedron network): Consider the dodecahedron network in Fig. 3.

To estimate the spectra, we applied the PMC Algorithm 3.1 with $N = 10^5$ sample size. This modest sample size was sufficient to estimate the spectra within 4% relative error (RE) [24]. Moreover, since we are dealing with a relatively small network, we were able to find the exact analytical spectra by exhaustive calculation. The minimal non-zero component is $s_6 = 690/38760 \approx 0.0178$, so our simulation results are in-line with Theorem 3.1. Fig. 4 shows the relative and absolute errors obtained in our simulation.

Unsurprisingly, we also get a good estimator for the RCR via (5). The analytical and the estimated reliability values are summarized in Fig. 5.

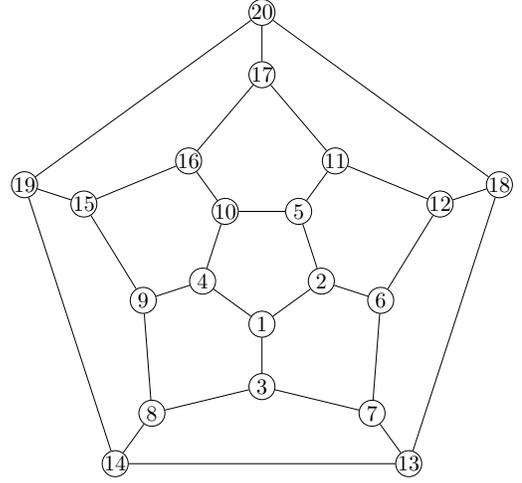


Fig. 3. The dodecahedron network with 20 vertices.

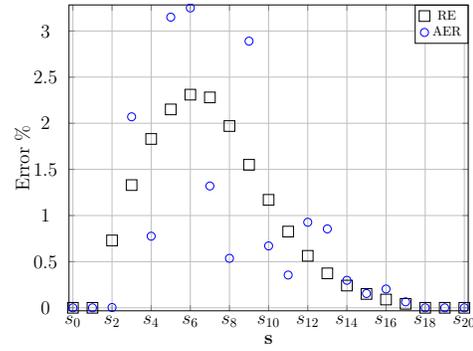


Fig. 4. The RE and AER obtained using the PMC algorithm with $N = 10^5$ sample size.

Remark 3.1 (Residual connectedness unreliability (RCU)): Although handling network unreliability is outside the scope of the current paper, it is important to note that the RCU $\bar{r}(G, q)$, can be estimated using the spectra from Definition 2.1 via

$$\widehat{r}(G, q) = \sum_{k=0}^n \binom{n}{k} (1 - \widehat{s}_k) p^k q^{n-k},$$

since $(1 - s_k)$ is the probability that k randomly selected vertices do not form a single connected component. We reused the empirical spectra obtained for the dodecahedron network to measure this network unreliability. The results are presented in Fig. 6.

Although the PMC Algorithm 3.1 can be very efficient for handling many problems, it will fail under the rare-event setting. In particular, PMC will fail to estimate small spectra components [11]. This failure can have a crucial effect on the overall RCR estimation. However, not all the spectra components are hard to estimate. That is, a reasonable approach for the spectra estimation can be summarized as follows.

- 1) Run the PMC Algorithm 3.1 to calculate the estimator $\widehat{\mathbf{S}} = \{\widehat{s}_0, \dots, \widehat{s}_n\}$.

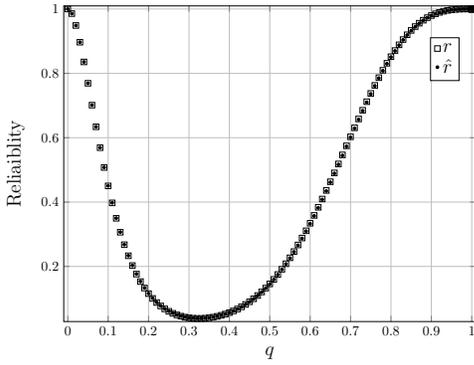


Fig. 5. The analytical and estimated reliability for the dodecahedron network.

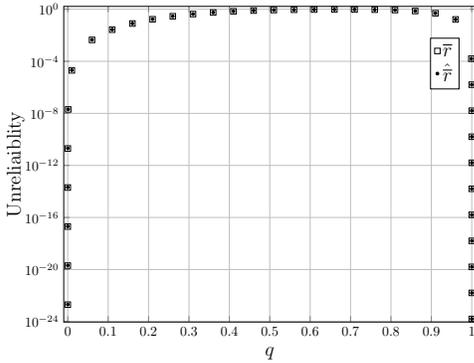


Fig. 6. The analytical and estimated unreliability for the dodecahedron network.

- 2) Define some threshold value T , say $T \in [10^{-2}, 10^{-4}]$, and apply some other approximation procedure to accurately estimate the spectra values that are smaller than T , that is, design an efficient approximation scheme for $\{\hat{s}_k \in \mathbf{S} \mid \hat{s}_k < T\}$.

The next section is solely dedicated to the second task.

IV. THE STOCHASTIC ENUMERATION METHOD

For the rest of this section, we consider the estimation of $s_k \in \mathbf{s}$ for some fixed $k = 0, \dots, n$. By Definition 2.1, it is sufficient to estimate the cardinality of \mathcal{B}_k , since $s_k = |\mathcal{B}_k| |\mathcal{A}_k|^{-1}$. To start with, we define a tree counting problem.

Definition 4.1 (The tree counting problem [23]): Consider a rooted tree $T = (\mathcal{V}, \mathcal{E})$ with node set \mathcal{V} and edge set \mathcal{E} (so that $|\mathcal{E}| = |\mathcal{V}| - 1$). We denote the root of the tree by v_0 , and for any $v \in \mathcal{V}$ the subtree rooted at v is denoted by T_v . With each node v is associated a cost $c(v) \in \mathbb{R}$. The tree counting problem is to calculate the total cost of the tree,

$$\text{Cost}(T) = \sum_{v \in \mathcal{V}} c(v),$$

or more generally, the total cost of a subtree T_v denoted by $\text{Cost}(T_v)$.

We next show that the task of calculating $|\mathcal{B}_k|$ reduces to the tree counting problem from Definition 4.1. To see this,

consider the following process of a tree construction. Given a graph $G = G(V, E)$, $|V| = n$, let (v_1, \dots, v_n) be an arbitrary vertex ordering. Using this ordering, define a binary tree T of height $n + 1$. For each tree node v , except for the leaves, we define the right and the left successors to indicate that v was used or not used to form a subgraph (of cardinality k), respectively.

Note that each path from the tree root corresponds to a specific induced subgraph of G (of cardinality k). We next partition the vertices of T into *regular* and *special* nodes. The node is called *special* if it is the first node on a path from the root, such that this path forms a connected subgraph with exactly k vertices. All other nodes are called *regular*. We complete the reduction by assigning appropriate costs to the *regular* and *special* nodes. Namely, each *regular* and *special* node v is assigned with $c(v) = 0$ and $c(v) = 1$, respectively. Finally, regardless of the initial vertex ordering, this construction yields

$$\text{Cost}(T) = \sum_{v \in \mathcal{V}} c(v) = \sum_{v \in \mathcal{V}, \text{ and } v \text{ special}} 1 = |\mathcal{B}_k|,$$

since for each path from the root to a leaf there is at most one *special* node. To get a better understanding of the above tree construction process, consider the reduction in Example 4.1.

Example 4.1 (Reduction to tree counting problem): We consider the network with 4 vertices in Fig. 7. For the remainder

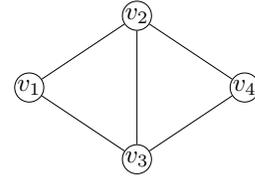


Fig. 7. Network with 4 vertices.

of this example we will be interested in the second spectra component, namely s_2 . Recall the tree reduction discussed in the beginning of this section and suppose, without loss of generality, that the vertex ordering is given by (v_1, v_2, v_3, v_4) . The corresponding tree is shown in Fig. 8. For convenience, for each tree node $v \in \mathcal{V}$, we label the left child (v is not used to form the subgraph) with “F”, and the right child (v is used to form the subgraph) with “T”. The *special* nodes are colored gray and numbered from 1 to 5. Indeed, note that each *special* node corresponds to a unique connected subgraph with exactly two vertices. For example, the leftmost *special* node corresponds to the subgraph induced by $\{v_3, v_4\}$. This can be verified by following the path from the root to this node.

To achieve a substantial variance reduction using the SE algorithm [22, Chapter 4], it is desirable to work with a partial tree in which we truncate all the subtrees that do not contain a *special* node. The truncated tree of the full binary in Fig. 8 is presented in Fig. 9.

Remark 4.1 (Tree truncation complexity): In order to truncate the reduction tree at some node $v \in \mathcal{V}$, we need to consider two cases.

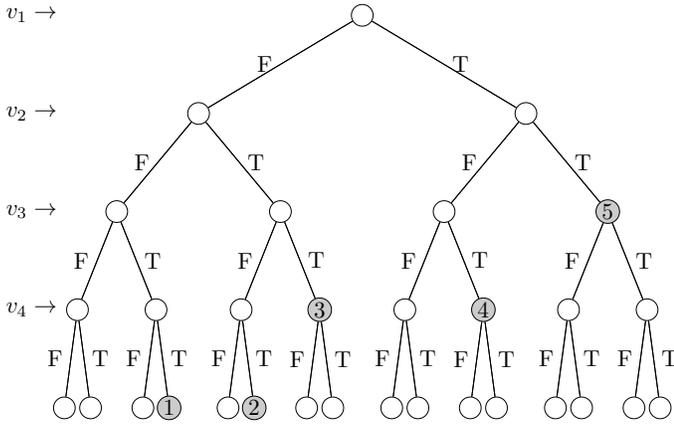


Fig. 8. The full binary tree.

- 1) If v is *special*, we just cut the entire tree rooted at v .
- 2) If v is not a *special* node, we run a graph traversal algorithm such as Breadth First Search (BFS) [26] from v , in order to identify a *special* node within the v -rooted subtree.

That is, for each node $v \in \mathcal{V}$, the time required for truncation is linear in the network size.

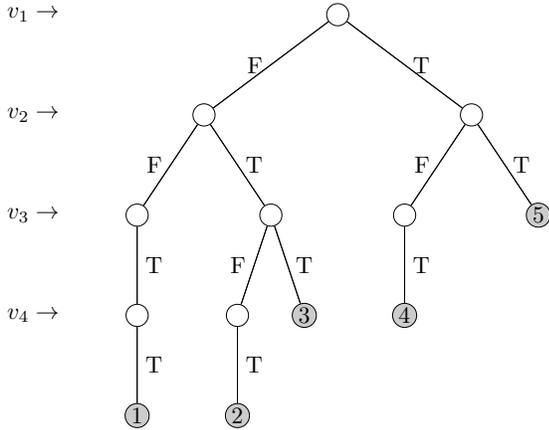


Fig. 9. The truncated binary tree.

We complete this Section by a brief overview of the SE algorithm for counting trees [23]. This method is an efficient generalization of Knuth's estimator [27]. To state the algorithm, we will need the following definition.

Definition 4.2 (Hyper nodes and forests [23]): Let $\{v_1, \dots, v_r\} \in \mathcal{V}$ be tree nodes.

- We call a collection $\mathbf{v} = \{v_1, \dots, v_r\}$ of distinct nodes in the same level of the tree a *hyper node* of cardinality $|\mathbf{v}| = r$.
- Let \mathbf{v} be a hyper node. Generalizing the tree node cost, we define the cost of the hyper node as $c(\mathbf{v}) = \sum_{v \in \mathbf{v}} c(v)$.
- Let \mathbf{v} be a hyper node. Define the set of successors of \mathbf{v} as $S(\mathbf{v}) = \bigcup_{v \in \mathbf{v}} S(v)$.

- Let \mathbf{v} be a hyper node and let $B \in \mathbb{N}$, $B \geq 1$. Define

$$H(\mathbf{v}) = \begin{cases} \{S(\mathbf{v})\} & \text{if } |S(\mathbf{v})| \leq B \\ \{\mathbf{w} \mid \mathbf{w} \subseteq S(\mathbf{v}), |\mathbf{w}| = B\} & \text{if } |S(\mathbf{v})| > B, \end{cases}$$

to be the set of all possible hyper nodes having cardinality $\min\{B, |S(\mathbf{v})|\}$ that can be formed from the set of \mathbf{v} 's successors. Note that if $|S(\mathbf{v})| \leq B$, we get a single hyper node with cardinality $|S(\mathbf{v})|$.

- For each hyper node \mathbf{v} let $T_{\mathbf{v}} = \bigcup_{v \in \mathbf{v}} T_v$ be the forest of trees rooted at \mathbf{v} . See Fig. 10 for an example of hyper node $\mathbf{v} = \{v_1, v_2, v_3, v_4\}$ and its corresponding forest $T_{\mathbf{v}} = \{T_{v_1}, T_{v_2}, T_{v_3}, T_{v_4}\}$.
- For each forest rooted at hyper node \mathbf{v} , define its total cost as $\text{Cost}(T_{\mathbf{v}}) = \sum_{v \in \mathbf{v}} \text{Cost}(T_v)$.

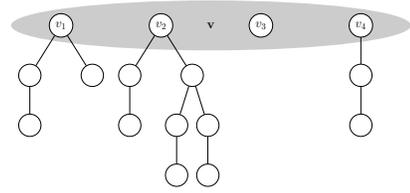


Fig. 10. Hyper node \mathbf{v} that contains regular tree nodes v_1, v_2, v_3 and v_4 with their corresponding subtrees.

The strength of the SE algorithm is due to its built in splitting mechanism [22, Chapter 4] and the fact that it operates on the tree hyper nodes. Definition 4.2 is sufficient to state the main SE algorithm.

Algorithm 4.1 (Stochastic Enumeration Algorithm [23]): Given a forest $T_{\mathbf{v}}$ of height h rooted at a hyper node \mathbf{v} , and a budget $B \geq 1$, execute the following steps.

- 1) **(Initialization):** Set $k \leftarrow 0$, $D \leftarrow 1$, $\mathbf{X}_0 = \mathbf{v}$ and $C_{\text{SE}} \leftarrow c(\mathbf{X}_0)/|\mathbf{X}_0|$.
- 2) **(Compute the successors):** Let $S(\mathbf{X}_k)$ be the set of all children of \mathbf{X}_k .
- 3) **(Terminal position?):** If $|S(\mathbf{X}_k)| = 0$, the algorithm stops, returning $|\mathbf{v}| C_{\text{SE}}$ as an estimator of $\text{Cost}(T_{\mathbf{v}})$.
- 4) **(Advance):** Choose hyper node $\mathbf{X}_{k+1} \in H(\mathbf{X}_k)$ at random, each choice being equally likely. (Thus, each choice occurs with probability $1/|H(\mathbf{X}_k)|$.) Set $D_k = |S(\mathbf{X}_k)|/|\mathbf{X}_k|$ and $D \leftarrow D_k D$, then set

$$C_{\text{SE}} \leftarrow C_{\text{SE}} + \left(\frac{c(\mathbf{X}_{k+1})}{|\mathbf{X}_{k+1}|} \right) D.$$

Increase k by 1 and return to Step 2.

It can be shown [23] that Algorithm 4.1 outputs a random variable

$$C_{\text{SE}} = \frac{c(\mathbf{X}_0)}{|\mathbf{X}_0|} + \frac{|S(\mathbf{X}_0)|}{|\mathbf{X}_0|} \frac{c(\mathbf{X}_1)}{|\mathbf{X}_1|} + \frac{|S(\mathbf{X}_0)|}{|\mathbf{X}_0|} \frac{|S(\mathbf{X}_1)|}{|\mathbf{X}_1|} \frac{c(\mathbf{X}_2)}{|\mathbf{X}_2|} + \dots + \left(\prod_{0 \leq j \leq \tau-1} \frac{|S(\mathbf{X}_j)|}{|\mathbf{X}_j|} \right) \frac{c(\mathbf{X}_\tau)}{|\mathbf{X}_\tau|}, \quad (8)$$

where $\tau \leq h$ is the random variable that represents the length of the random walk. The term

$$\left(\prod_{0 \leq j \leq k-1} \frac{|S(\mathbf{X}_j)|}{|\mathbf{X}_j|} \right) \frac{c(\mathbf{X}_k)}{|\mathbf{X}_k|} = D \frac{c(\mathbf{X}_k)}{|\mathbf{X}_k|},$$

which is calculated at Step 4 of Algorithm 4.1 is the estimator for the total cost of the vertices at tree level k . Moreover, this estimator is unbiased; that is, for a tree T rooted at v_0 , and for $\mathbf{v}_0 = \{v_0\}$, $\mathbb{E}(C_{SE}(T_{\mathbf{v}_0})) = \text{Cost}(T)$. For the proof of unbiasedness, efficiency considerations, and examples, we refer to [11] and [23].

V. NUMERICAL EXPERIMENTS

In our numerical experiments we considered three different estimators of the spectra. The sample sizes and budgets used were the same for the first three examples, but different for the multi-dimensional cube example. For the first three examples we used PMC with $N = 10^7$. The stochastic enumeration method was used with $B = 10^5$ and a fixed vertex ordering. We refer to this estimator subsequently as SE, to distinguish it from the third estimator which we refer to as SE+PERM. For SE+PERM we took our estimator to be an average of ten different applications of SE with $B = 10^4$. In the first of the ten applications the vertex ordering was fixed, and in the remaining nine the vertex ordering was randomly chosen. Note that different vertex orderings lead to different trees (although the cost of the tree is unchanged). In the last example the budgets used for the stochastic enumeration methods were smaller. For SE we used $B = 2000$, and for SE+PERM we used 10 repetitions with $B = 1000$.

Each of these methods was run 500 times for each example. We compare the different estimators using the *work normalized variance*. For any estimator $\hat{\ell}$ this is $T \times \widehat{\text{Var}}(\hat{\ell})$, where T is the average time taken to compute the estimate.

Example 5.1 (Dodecahedron): We consider again the dodecahedron graph (Fig. 3). Under the SE method the budget was large enough that every spectra component was computed exactly. For the SE+PERM method the smaller budget for each of the ten applications meant that the spectra was not always computed exactly (Fig. 11). However on a work-normalized basis the SE+PERM method still outperforms the PMC method at estimating the spectra components by a factor of almost 100. When these spectra estimates are used to compute the RCR, the SE+PERM algorithm generally outperforms PMC, especially for $p < 0.25$. However for p close to 0.5 the PMC algorithm performs better with the specified sample sizes.

Example 5.2 (12×12 grid): Take G to be the 12×12 two-dimensional grid graph. In general a graph of this size is far too large for the spectra to be computed exactly. For regular grid graphs the *transfer matrix method* [28] allows the exact computation of the spectra much more quickly. Even so, the 12×12 grid graph is towards the limit of this technique. The average estimates of the spectra components are shown in Fig. 13, along with the exact values. Note that PMC gives spectra

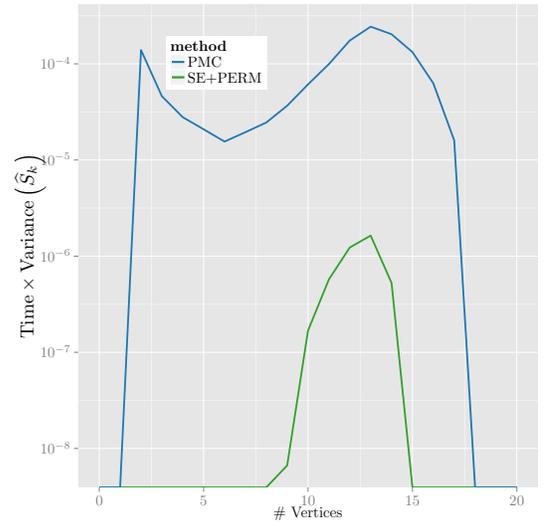


Fig. 11. Work normalized variances of estimates of the spectra components for the dodecahedron graph.

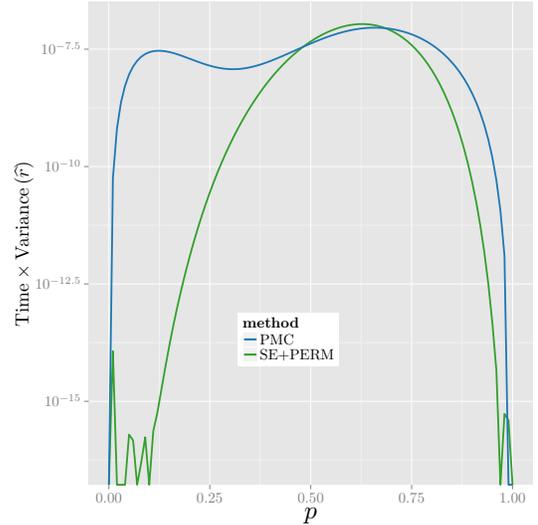


Fig. 12. Work normalized variances of estimates of the RCR for the dodecahedron graph.

component estimates of 0 for a range of subgraph sizes. Also, the average estimate for some components is bigger than 1 for SE and SE+PERM. This illustrates a possible weakness of the SE algorithm; it may return estimates bigger than 1, and may do so often. As all the estimators tested are unbiased, significant deviation from the exact value suggests that some of these estimates have a large variance.

Work normalized variances of the estimates of the spectra components are given in Fig. 14. Note that PMC outperforms SE and SE+PERM significantly for spectra components for sizes bigger than 100, but SE+PERM performs best for all other components. In general SE+PERM appears to outperform SE by a factor of 10 - 100. For the RCR, the work normalized variance estimates suggest that PMC performs well

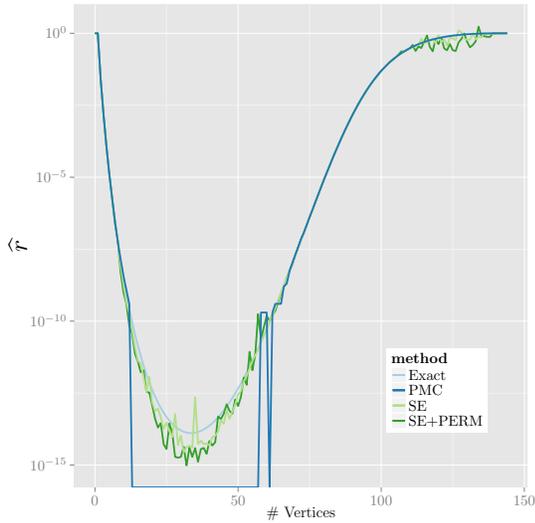


Fig. 13. Average estimates of spectra components for the 12×12 grid graph.

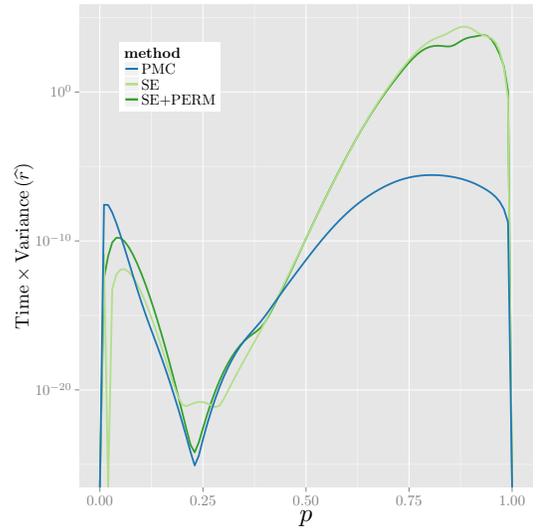


Fig. 15. Work normalized variances of estimates of the RCR for the 12×12 grid graph.

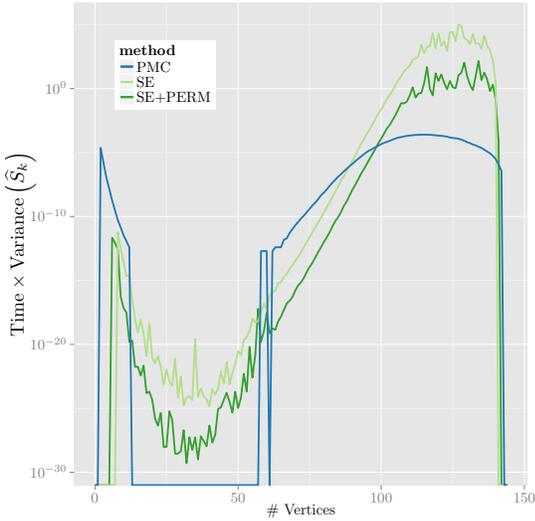


Fig. 14. Work normalized variances of estimates of spectra components for the 12×12 grid graph.

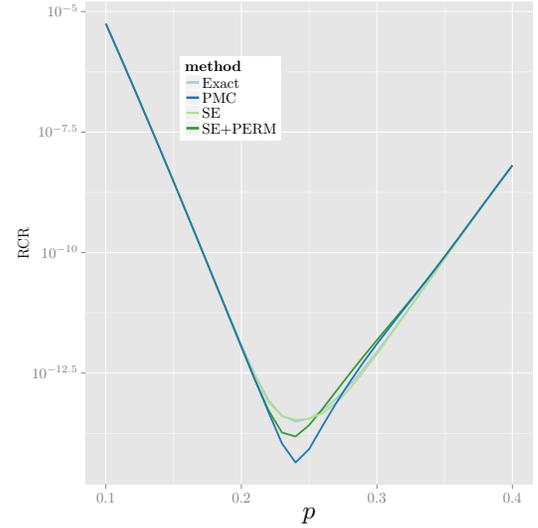


Fig. 16. Average estimates of the RCR for the 12×12 grid graph.

(Fig. 15). However it is likely that the variance of the PMC method is poorly estimated for some values of p , as demonstrated by its mean estimated value being significantly different from the true value, when averaged over 500 replications (Fig. 16). This confirms that PMC does not handle the rare-event probability problem well. Due to the difficulty of interpreting Fig. 15, it is simplest to evaluate the methods based on how well the sample mean approximates the true RCR value (Fig. 16). Based on this, the SE method appears to be the best performer for $p < 0.35$. For p larger the PMC method appears to perform best.

Example 5.3 (7 × 7 hexagonal grid): For the hexagonal grid, the PMC method gives an estimate of zero for a range of spectra components (Fig. 17). In general the SE+PERM

method performs best, except for components corresponding to large subgraph sizes, in which case PMC performs best (Fig. 18). For the RCR, there is a significant difference between the average estimated values for the SE methods and the PMC method (Fig. 20). It appears that apart for p close to 0.3, for $p < 0.5$ both of the SE methods significantly outperform the PMC method (Fig. 19). The PMC method appears to outperform both SE methods around $p = 0.3$, but this corresponds to parameters where the difference between the methods is largest (Fig. 20). Therefore the variance of the PMC method is believed to be poorly estimated for this parameter value.

Example 5.4 (5 dimensional cube): Let G be a regular grid graph of side length 3 and dimension 5. The spectra component estimates are significantly different for the PMC

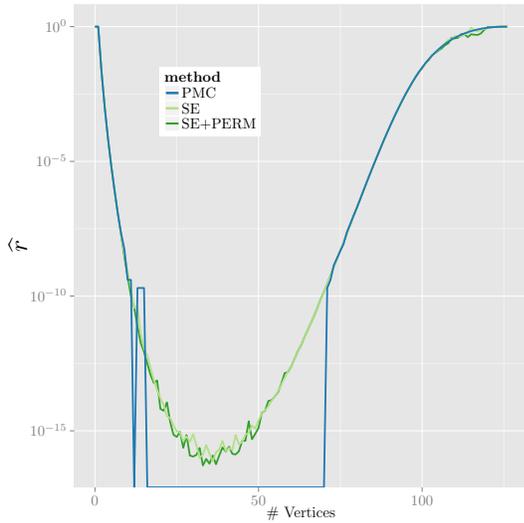


Fig. 17. Average estimates of spectra components for the 7×7 hexagonal grid graph.

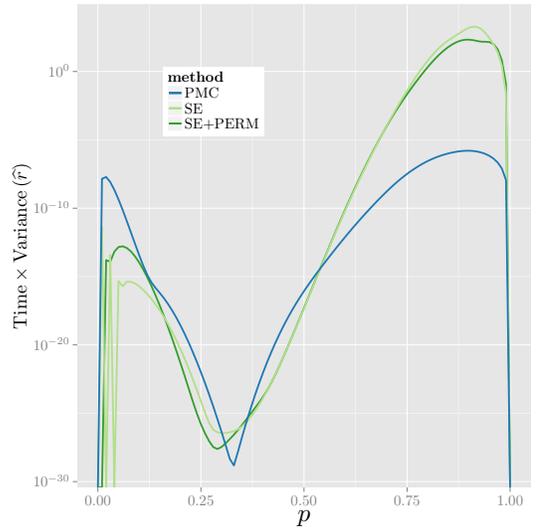


Fig. 19. Work normalized variances of estimates of the RCR for the 7×7 hexagonal grid graph.

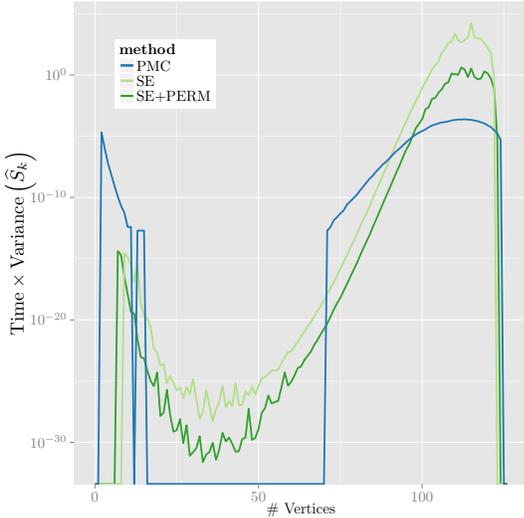


Fig. 18. Work normalized variances of estimates of spectra components for the 7×7 hexagonal grid graph.

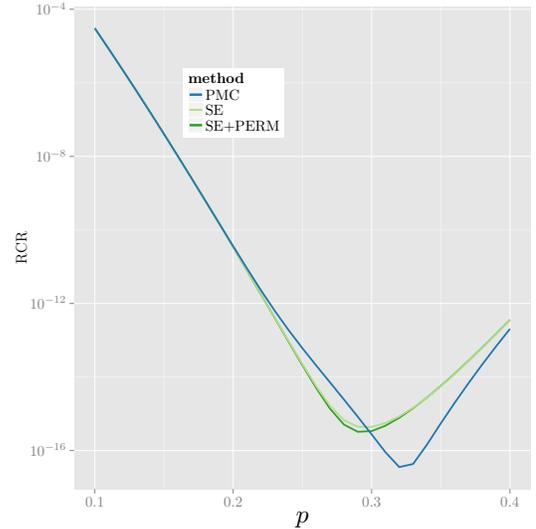


Fig. 20. Average estimates of the RCR for the 7×7 hexagonal grid graph.

and the SE methods (Fig. 21). For components corresponding to subgraph sizes 10 to 100, the average PMC estimate is either 0 or significantly larger than the SE estimates. In this case we believe the SE estimates are more accurate. For components corresponding to subgraph sizes 100 to 150, both methods agree and the estimates are believed to be accurate, although the PMC method has a significantly lower work normalized variance. However for components 150 and larger the SE methods give clearly inaccurate estimates. The work normalized variance of the spectra component estimates is not shown as some values are poorly estimated, making the plot difficult to interpret. It is not easy to say which of SE and SE+PERM performs better at estimating the spectral components.

The work normalized variances of the estimates of the RCR are shown in Fig. 22. For p between approximately 0.1 and 0.2 the SE and SE+PERM methods outperform the PMC method by a factor of up to 10^5 , and this is likely an underestimate. For $p > 0.25$ the PMC method is believed to outperform the SE and SE+PERM methods.

VI. CONCLUSIONS AND FUTURE RESEARCH

In this paper we propose to use the widely employed spectra approach for the problem of RCR estimation. We showed that the PMC algorithm is suitable for solving easy RCR problems while providing rigorous performance guarantees. In order to handle harder RCR problems, we introduced a new sampling scheme that adapts the SE algorithm to the task of estimating the RCR spectra. Our numerical results indicate

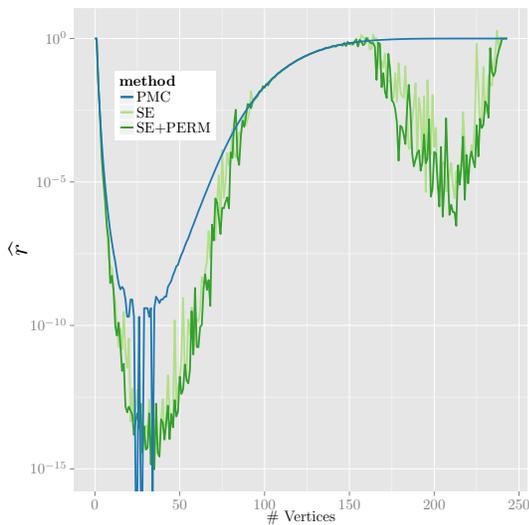


Fig. 21. Average estimates of spectra components for the 5 dimensional grid graph.

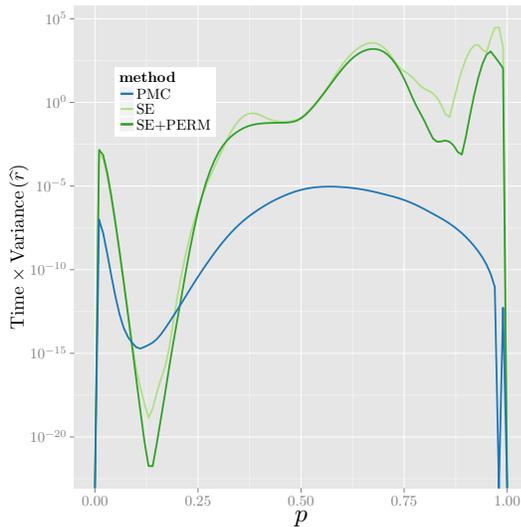


Fig. 22. Work normalized variances of estimates of the RCR for the 5 dimensional grid graph.

that the SE algorithm is able to successfully handle the hard task of estimating small spectra components while using a relatively modest sample size. For future theoretical work, it will be interesting to find specific classes of graphs for which the SE algorithm has a proven performance guarantee. From a practical point of view, this research opens up an important opportunity to handle network sensitivity using the obtained spectra.

REFERENCES

[1] K. Sutner, A. Satyanarayana, and C. L. Suffel, "The complexity of the residual node connectedness reliability problem," *SIAM J. Comput.*, vol. 20, no. 1, pp. 149–155, 1991.
 [2] A. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall Professional Technical Reference, 2002.

[3] E. M. Royer and C. K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, pp. 46–55, 1999.
 [4] I. B. Gertsbakh and Y. Shpungin, *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*. Boca Raton, FL, USA: CRC Press, Inc., 2009.
 [5] T. Elperin, I. B. Gertsbakh, and M. Lomonosov, "Estimation of network reliability using graph evolution models," *IEEE Transactions on Reliability*, vol. 40, no. 5, pp. 572–581, 1991.
 [6] T. Elperin, I. Gertsbakh, and M. Lomonosov, "An evolution model for Monte Carlo estimation of equilibrium network renewal parameters," *Probability in the Engineering and Informational Sciences*, vol. 6, pp. 457–469, 1992.
 [7] P. L'Ecuyer, G. Rubino, S. Saggadi, and B. Tuffin, "Networks – approximate zero-variance importance sampling for static network reliability estimation," *IEEE transactions on reliability*, vol. 8, pp. 590–604, 2011.
 [8] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of Monte Carlo methods*. New York: John Wiley & Sons, 2011.
 [9] H. Cancela, M. E. Khadiri, and G. Rubino, "Rare event analysis by monte carlo techniques in static models," in *Rare Event Simulation Using Monte Carlo Methods*, 1st ed. New York: Wiley, 2009, pp. 145–170.
 [10] Z. I. Botev, P. L'Ecuyer, G. Rubino, R. Simard, and B. Tuffin, "Static network reliability estimation via generalized splitting," *INFORMS J. on Computing*, vol. 25, no. 1, pp. 56–71, Jan. 2013.
 [11] R. Vaisman, D. P. Kroese, and I. B. Gertsbakh, "Improved sampling plans for combinatorial invariants of coherent systems," *IEEE transactions on reliability*, pp. 1–15, 2015, early access. [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7161416>
 [12] I. B. Gertsbakh, Y. Shpungin, and R. Vaisman, *Ternary Networks: Reliability and Monte Carlo*, ser. Springer Briefs in Electrical and Computer Engineering. Springer, 2014.
 [13] M.-S. Lin and C.-C. Ting, "A polynomial-time algorithm for computing K-terminal residual reliability of d-trapezoid graphs," *Information Processing Letters*, vol. 115, pp. 371–376, 2015.
 [14] C. J. Colbourn, A. Satyanarayana, C. Suffel, and K. Sutner, "Computing residual connectedness reliability for restricted networks," *Discrete Applied Mathematics*, vol. 44, no. 1, pp. 221–232, 1993.
 [15] A. A. Chernyak, "Residual reliability of p-threshold graphs," *Discrete Applied Mathematics*, vol. 135, no. 1-3, pp. 83–95, 2004.
 [16] H. Cancela and M. Urquhart, "Adapting RVR simulation techniques for residual connectedness network reliability models," *IEEE Transactions on Computers*, vol. 51, no. 4, pp. 439–443, Apr 2002.
 [17] R. Shah and D. P. Kroese, "Estimating residual connectivity for random graphs," 2015, submitted.
 [18] F. J. Samaniego, *System Signatures and Their Applications in Engineering Reliability*, ser. International Series in Operations Research & Management Science. Springer, 2007.
 [19] E. S. de Cabezán and H. P. Wynn, "Hilbert functions in design for reliability," *IEEE transactions on reliability*, no. 99, pp. 1–11, 2014.
 [20] I. B. Gertsbakh and Y. Shpungin, "Combinatorial approach to computing component importance indexes in coherent systems," *Probability in the Engineering and Informational Sciences*, vol. 26, pp. 117–128, 1 2012.
 [21] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 410–421, 1979.
 [22] R. Y. Rubinstein, A. Ridder, and R. Vaisman, *Fast Sequential Monte Carlo Methods for Counting and Optimization*. New York: John Wiley and Sons, 2013.
 [23] R. Vaisman and D. P. Kroese, "Stochastic enumeration algorithm for counting trees," *Methodology and Computing in Applied Probability*, in press, 2015.
 [24] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method, Second Edition*. New York: John Wiley and Sons, 2008.
 [25] W. Feller, *An Introduction to Probability Theory and Its Applications*, ser. Wiley mathematical statistics series. New York: John Wiley & Sons, 1968, vol. 1.
 [26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.
 [27] D. E. Knuth, "Estimating the efficiency of backtrack programs," *Math. Comp.*, vol. 29, 1975.
 [28] D. J. Klein, G. E. Hite, and T. G. Schmalz, "Transfer-matrix method for subgraph enumeration: Applications to polypyrène fusenes," *Journal of Computational Chemistry*, vol. 7, no. 4, pp. 443–456, 1986.