Reliability estimation in networks with link and node failures

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

and

Antonio Mauttone
Universidad de la República, F. de Ingeniería, Instituto de Computación
Montevideo, Uruguay
mauttone@fing.edu.uy

Abstract

The RVR (recursive variance reduction) simulation technique has been used with success for the evaluation of the $\mathcal{K}$-*terminal reliability* measure of networks where only links can fail.

In this paper, we show how this technique can be adapted for computing the $\mathcal{K}$-*terminal reliability* measure in the case of networks where both links and nodes can fail.

Experimental results show the interest of this technique, which can improve the precision of reliability estimation for highly reliable networks.

Key words — Network reliability, Monte Carlo methods, variance reduction techniques, recursive variance reduction.

Resumen

El método RVR (reducción recursiva de la varianza - recursive variance reduction) es una técnica de simulación que ha sido empleada con éxito para la evaluación de la confiabilidad $\mathcal{K}$-*terminal reliability* en el caso de redes con fallas en las líneas y nodos perfectos.

En este trabajo, se muestra como esta técnica puede ser empleada para calcular la confiabilidad $\mathcal{K}$-*terminal* en el caso de redes con fallas en nodos y aristas.

Resultados experimentales muestran el interés de esta técnica, que puede aumentar drásticamente la precisión de la estimación de la confiabilidad para redes altamente confiables.

Palabras clave — Confiabilidad de redes, métodos Monte Carlo, técnicas de reducción de la varianza, redes de comunicaciones.

1 Introduction

The system under study is an undirected connected communication network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{K})$ consisting of a set of nodes $\mathcal{V}$, a set of connecting links $\mathcal{E}$ and a set of terminals $\mathcal{K}$ (a fixed subset of the node-set). Both nodes and links can fail, and are assigned an s-independent probability of failure q_l (called node/link unreliability).

The success of communication between nodes in $\mathcal{K}$ is a random event, which has probability $R(\mathcal{G})$. The problem of evaluating $R(\mathcal{G})$ or its complement, $Q(\mathcal{G}) = 1 - R(\mathcal{G})$, is usually called the *$\mathcal{K}$-terminal reliability* problem.

In graph terms, $R(\mathcal{G})$ is the probability that there is at least a path between any pair of nodes in $\mathcal{K}$ with all its nodes and links working. The particular case $\mathcal{K} = \{s, t\}$ is called the *source-to-terminal* network reliability problem. This approach is used not only for the evaluation of communication networks but also in the case of other binary stochastic systems.

The exact evaluation of $R(\mathcal{G})$ was shown to be *NP*-hard [1], even in the special cases where $\mathcal{K} = \{s, t\}$ or $\mathcal{K} = V$, and where only nodes or only links can fail. Then exact evaluation techniques, like the ones introduced in [10, 11], have worst case exponential computation time.

In the particular case of nodes not subject to failure, a number of works [2, 4, 3] have shown that a simulation technique known as RVR (Recursive Variance Reduction) has the potential for efficient evaluation of the measure. In this paper we study the application of this technique to the more general case where both nodes and links can fail.

The paper is organized as follows. The rest of this section presents some general notation and preliminary definitions. Section 2 recalls the RVR method in the context of perfect-node networks, and Section 3 presents the RVR method in the general case (when both links and nodes can fail). Section 4 is devoted to present the results of some numerical experiments. Finally, Section 5 corresponds to the conclusions of this work.

Notation (General)

$\mathcal{G}$	$(\mathcal{V}, \mathcal{E}, \mathcal{K})$: an undirected network topology
$\mathcal{V}$	$\{u_1, \ldots, u_n\}$: the node-set of $\mathcal{G}$
$\mathcal{E}$	$\{l_1, \ldots, l_m\}$: the link-set of $\mathcal{G}$
$\mathcal{K}$	*target set* of $\mathcal{G}$, $\mathcal{K} \subseteq \mathcal{V}$; the nodes of $\mathcal{K}$ are the *terminals* of $\mathcal{G}$
m,n,k	the number of [links, nodes,terminals] of $\mathcal{G}$
$\lvert A \rvert$	cardinality of the set A
$\mathbf{1}_{(.)}$	indicator function: $\mathbf{1}_{True} = 1$, $\mathbf{1}_{False} = 0$
$\overline{a}$	real $1 - a$
x_l	$\mathbf{1}_{(\text{link } l \text{ is up})}$: state of link l
r_l	$\Pr\{x_l = 1\}$: operating probability of link l
q_l	$\Pr\{x_l = 0\} = 1 - r_l$: failure probability of link l
x_u	$\mathbf{1}_{(\text{node } u \text{ is up})}$: state of node u
r_u	$\Pr\{x_u = 1\}$: operating probability of node u
q_u	$\Pr\{x_u = 0\} = 1 - r_u$: failure probability of node u
$X_{\mathcal{G}}$	$(x_{l_1}, \ldots, x_{l_m}, x_{u_1}, \ldots, x_{u_n})$: random network-state vector
$\mathcal{G}_X$	network with node-set derived from $\mathcal{V}$ and link-set derived from $\mathcal{E}$ by removing respectively all failed nodes and links in $X_{\mathcal{G}}$
$\Phi_{\mathcal{G}}$	structure function associated with the $\mathcal{K}-$*terminal reliability* measure : $\Phi_{\mathcal{G}}(X) = \mathbf{1}_{(\mathcal{G}_X \text{ is } \mathcal{K}\text{-connected})}$ for $X \in \{0,1\}^{m+n}$
$Y_{\mathcal{G}}$	$1 - \Phi_{\mathcal{G}}(X_{\mathcal{G}})$: random state of the network $\mathcal{G}$
$R(\mathcal{G})$	$\Pr\{\mathcal{G} \text{ is } \mathcal{K}\text{-connected}\} = \mathrm{E}\{\Phi_{\mathcal{G}}(X_{\mathcal{G}})\}$: $\mathcal{K}-$*terminal reliability* of $\mathcal{G}$
$Q(\mathcal{G})$	$\Pr\{\mathcal{G} \text{ is not } \mathcal{K}\text{-connected}\} = \mathrm{E}\{Y_{\mathcal{G}}\} = 1 - R(\mathcal{G})$: $\mathcal{K}-$*terminal unreliability* of $\mathcal{G}$

Definitions & Nomenclature

- A network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{K})$ is $\mathcal{K}-$connected if there is at least one path in $\mathcal{G}$ between every pair of nodes in $\mathcal{K}$ (all the nodes in $\mathcal{K}$ are connected in $\mathcal{G}$).

- A subset C of $\mathcal{E}$ is a $\mathcal{K}$-cutset (or $\mathcal{K}$-link cutset) of $\mathcal{G}$ if the subnetwork $\mathcal{G}' = (\mathcal{V}, \mathcal{E} - C, \mathcal{K})$ of $\mathcal{G}$ is not $\mathcal{K}$-connected.
- A subset C of $\mathcal{V}$ is a $\mathcal{K}$-node cutset of $\mathcal{G}$ if the subnetwork $\mathcal{G}' = (\mathcal{V} - C, \mathcal{E}, \mathcal{K})$ of $\mathcal{G}$ is not $\mathcal{K}$-connected.
- A set $D \subset \mathcal{V} \cup \mathcal{E}$ is a $\mathcal{K}$-extended cutset of $\mathcal{G}$ if the subnetwork $\mathcal{G}' = (\mathcal{V} - D, \mathcal{E} - D, \mathcal{K})$ of $\mathcal{G}$ is not $\mathcal{K}$-connected.
- A $\mathcal{K}$-tree of $\mathcal{G}$ is a $\mathcal{K}$-connected tree of $\mathcal{G}$ whose leaves belong to $\mathcal{K}$ [8].
- For a given link $l \in \mathcal{E}$ in $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{K})$, $\mathcal{G} - l$ denotes the network with node-set $\mathcal{V}$ and link-set derived from $\mathcal{E}$ by removing link $l \in \mathcal{E}$. The target set of $\mathcal{G} - l$ is equal to $\mathcal{K}$.
- For a given node $u \in \mathcal{V}$ in $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{K})$, $\mathcal{G} - u$ denotes the network with node-set $\mathcal{V} - \{u\}$ and link-set derived from $\mathcal{E}$ by removing all links incident to u. The target set of $\mathcal{G} - u$ is equal to $\mathcal{K} - \{u\}$.
- For a given link l in $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{K})$, $\mathcal{G}|l$ denotes the network derived from $\mathcal{G}$ by setting the operating probability of link l, p_l to 1.
- For a given node u in $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{K})$, $\mathcal{G}|u$ denotes the network derived from $\mathcal{G}$ by setting the operating probability of node u, $p_u = 1$.

2 Recursive Variance Reduction Monte Carlo method for perfect-node networks

The aim of variance-reduction methods is to construct random variables having same s-expectation as $Y_{\mathcal{G}} = 1 - \Phi_{\mathcal{G}}(X_{\mathcal{G}})$ and smaller variance, since a sample mean based on such a r.v. is a more accurate estimator than the crude one based on $Y_{\mathcal{G}}$.

We give here a brief summary of a previous paper [2], where we have shown that r.v. $Z_{\mathcal{G}}$ recalled below in Proposition 2.1 can be used as the basis of a variance reduction method for perfect-node network reliability estimation (i.e., it is more accurate than the one based on $Y_{\mathcal{G}}$). This random variable is expressed in terms of $|C|$ network-state random variables $Y_{\mathcal{G}_i}$, $1 \leq i \leq |C|$, where C is a $\mathcal{K}$-(link) cutset of the network $\mathcal{G}$. Each $\mathcal{G}_i$ is deduced from $\mathcal{G}$ by $(i-1)$ link deletions and one link contraction.

Proposition 2.1 *For any network $\mathcal{G}$, let*

C	$\{l_1, l_2, \ldots, l_{\|C\|}\}$*: a fixed $\mathcal{K}$-cutset in $\mathcal{G}$*
A_C	*the event "all links in C are in failed state"*
Q_C	$\Pr\{A_C\} = \prod_{i=1}^{\|C\|} q_{l_i}$
B_i	*the event "all links in $\{l_1, \ldots, l_{i-1}\}$ are failed and l_i is up"*
$\mathcal{G}_i$	$(\mathcal{G} - l_1 - \ldots - l_{i-1})\,\|l_i$
V	*discrete r.v. s-independent from r.v. $Y_{\mathcal{G}_i}$'s, with*

$$pmf\{v\} = \Pr\{V = v\} = \Pr\{B_v\}/\overline{Q_C} = ((1 - q_{l_v}) \prod_{j=1}^{v-1} q_{l_j})/\overline{Q_C},\ 1 \leq v \leq |C|,$$

then the random variable

$$Z_{\mathcal{G}} = Q_C + \overline{Q_C} \sum_{i=1}^{|C|} \mathbf{1}_{(V=i)} Y_{\mathcal{G}_i} \tag{1}$$

verifies

$$\mathrm{E}\{Z_{\mathcal{G}}\} = Q(\mathcal{G}) \tag{2}$$

$$\mathrm{Var}\{Z_{\mathcal{G}}\} = (Q(\mathcal{G}) - Q_C)R(\mathcal{G}) \leq Q(\mathcal{G})R(\mathcal{G}) = \mathrm{Var}\{Y_{\mathcal{G}}\}. \tag{3}$$

In [2], we have shown that the $Z_{\mathcal{G}}$ definition can be recursively used to construct a random variable $F(\mathcal{G})$ giving a more accurate estimator than the one based on $Z_{\mathcal{G}}$. Using the same notation as in Proposition 2.1, $F(\mathcal{G})$ can been expressed as follows:

$$F(\mathcal{G}) = \begin{cases} 1 & \text{if } \mathcal{G} \text{ is not } \mathcal{K}\text{-connected;} \\ 0 & \text{if } \mathcal{K} \text{ is a singleton set;} \\ Q_C + \overline{Q_C} \sum_{i=1}^{|C|} \mathbf{1}_{(V=i)} F(\mathcal{G}_i) & \text{otherwise.} \end{cases} \tag{4}$$

The algorithm based on F(G) is called RVR simulation algorithm.

A procedure to obtain a trial of $F(\mathcal{G})$ can be summarized as follows:

Procedure RVR($\mathcal{G}$)

Input: network $\mathcal{G}$
Output: a random sample of r.v. $F(\mathcal{G})$

1. Check end recursion condition:
   1.1. Check if $\mathcal{G}$ is always $\mathcal{K}$-connected: If $|\mathcal{K}| = 1$ return(0).
   1.2. Check if the network is not $\mathcal{K}$-connected: If $\mathcal{G}$ is not $\mathcal{K}$-connected return(1).
2. Find a K-cutset C of $\mathcal{G}$: $C = \{l_1, \ldots, l_{|C|}\}$ the set of all links adjacent to $s \in K$ arbitrary.
3. Compute the probability that all components in C are failed: $Q_C = \prod_{i=1}^{|C|} q_{l_i}$.
4. Generate a trial v of V (with distribution $\Pr\{V = i\} = \Pr\{B_i\} / \overline{Q_C}$, $1 \leq i \leq |C|$).
5. Construct the corresponding network : $\tilde{\mathcal{G}}_v = (\mathcal{G} - l_1 - l_2 - \ldots - l_{v-1}) | l_v$.
6. Recursive step: return($Q_C + \overline{Q_C} \times$ RVR($\tilde{\mathcal{G}}_v$)).

When computing a trial for $F(\mathcal{G})$, we generate a trial v of the r.v. V (task 4). Since $\mathbf{1}_{(V=i)}$ is equal to 0 for all $i \neq v$, then only the term $\mathbf{1}_{(V=v)}$ survives. It results that the computations involved in the recursion process can be represented by a linear computational structure. Its root corresponds to the network $\mathcal{G}$ under study, each internal node corresponds to a recursive call (task 6) and the last node presents a network that can be exactly evaluated (conditions 1.1 and 1.2 of the above procedure). Because at each recursive step the number of links of the network resulting from task 5 is diminished by at least 1, the size of the related linear structure is bounded by $|\mathcal{E}|$.

In [2], we have shown that each recursive step of the RVR procedure has linear time complexity (in the number of links $|\mathcal{E}|$). The total complexity to generate a trial of $F(\mathcal{G})$ is then quadratic in the number of links $|\mathcal{E}|$.

3 Recursive Variance Reduction Monte Carlo method for general networks

We study in this section the case where both nodes and links can fail. We present some remarks which will be used to simplify our work.

Remark 3.1 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{K})$ be any network, $\mathcal{K} = \{u_1, \ldots, u_{|\mathcal{K}|}\}$. Then*

$$R = \left(\prod_{u_i \in \mathcal{K}} p_{u_i} \right) R(\mathcal{G} | u_1 | u_2 | \ldots | u_{|K|}) \tag{5}$$

and

$$Q(\mathcal{G}) = (1 - \prod_{u_i \in \mathcal{K}} p_{u_i}) + \left(\prod_{u_i \in \mathcal{K}} p_{u_i} \right) Q(\mathcal{G} | u_1 | u_2 | \ldots | u_{|K|}). \tag{6}$$

Proof . The proof is immediate from the definitions of $R(\mathcal{G})$ and $Q(\mathcal{G})$ (if a terminal node does not work, it can't communicate with the other terminals). □

¿From now on we will assume that we are working with networks whose terminals are perfect (but other nodes may fail), as the reliability of any network can be reduced to the reliability of a network with these characteristics.

Remark 3.2 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{K})$ be any network, and let $l = (u_1, u_2)$ be a link of $\mathcal{G}$ such that $p_l = p_{u_1} = p_{u_2} = 1$. We denote $\mathcal{G} * l$ the network derived from $\mathcal{G}$ by contracting l (eliminating l; merging its extremities u_1 and u_2 into a new node w; and setting the terminal set to $\mathcal{K} - \{u_1, u_2\} \cup \{w\}$ if u_1 or u_2 belongs to $\mathcal{K}$, and $\mathcal{K}$ otherwise).*

Then

$$R(\mathcal{G}) = R(\mathcal{G} * l). \tag{7}$$

A proof of this property may be easily obtained by establishing a bijection between the states of network $\mathcal{G}$ and those of network $\mathcal{G} * l$, remarking that corresponding states have the same probability, and proving that whenever a state of $\mathcal{G}$ is an operating (respectively failed) state of the network, the corresponding state of $\mathcal{G} * l$ is also an operating (respectively failed) state.

As in the case of perfect-node networks, we now start by constructing a random variable $Z_{\mathcal{G}}$ having same s-expectation as $Y_{\mathcal{G}} = 1 - \Phi_{\mathcal{G}}(X_{\mathcal{G}})$ and smaller variance. This random variable is expressed in terms of $|D|$ network-state random variables $Y_{\mathcal{G}_i}$, $1 \leq i \leq |D|$, where D is a $\mathcal{K}$-extended cutset of the network $\mathcal{G}$. For any network $\mathcal{G}$ such that $p_u = 1\ \forall u \in \mathcal{K}$, we define the random variable

$$Z_{\mathcal{G}} = Q_D + \overline{Q_D} \sum_{i=1}^{|D|} \mathbf{1}_{(V=i)} Y_{\mathcal{G}_i}, \tag{8}$$

where:

D	$\{e_1, e_2, \ldots, e_{\lvert D\rvert}\}$: a fixed $\mathcal{K}$-extended cutset in $\mathcal{G}$
A_D	the event "all components in D are in failed state"
Q_D	$\Pr\{A_D\} = \prod_{i=1}^{\lvert D\rvert} q_{e_i}$
B_i	the event "all components in $\{e_1, \ldots, e_{i-1}\}$ are failed and e_i is up"
$\mathcal{G}_i$	$(\mathcal{G} - e_1 - \ldots - e_{i-1})\,\lvert e_i$
V	discrete r.v. s-independent from r.v. $Y_{\mathcal{G}_i}$'s, with

$$pmf\{v\} = \Pr\{V = v\} = \Pr\{B_v\}/\overline{Q_D} = ((1 - q_{e_v}) \prod_{j=1}^{v-1} q_{e_j})/\overline{Q_D},\ 1 \leq v \leq |D|$$

Proposition 3.3 *For any network $\mathcal{G}$ such that $p_u = 1\ \forall u \in \mathcal{K}$, the random variable $Z_{\mathcal{G}}$ defined in Equation 8 verifies*

$$\mathrm{E}\{Z_{\mathcal{G}}\} = Q(\mathcal{G}) \tag{9}$$

$$\mathrm{Var}\{Z_{\mathcal{G}}\} = (Q(\mathcal{G}) - Q_D)R(\mathcal{G}) \leq Q(\mathcal{G})R(\mathcal{G}) = \mathrm{Var}\{Y_{\mathcal{G}}\}. \tag{10}$$

This property is also easy to prove, applying the total probability theorem with the partition $\{A_D\} \cup \{B_i, 1 \leq i \leq |D|\}$ of the probability space of network $\mathcal{G}$.

We now proceed as in [2] to apply recursively the construction process of $Z_{\mathcal{G}}$ to find random variable $F(\mathcal{G})$ giving a more accurate estimator than the one based on $Z_{\mathcal{G}}$. Using the same notation as before, $F(\mathcal{G})$ has been expressed as follows:

$$F(\mathcal{G}) = \begin{cases} 1 & \text{if } \mathcal{G} \text{ is not } \mathcal{K}\text{-connected;} \\ 0 & \text{if } \mathcal{K} \text{ is a singleton set;} \\ Q_D + \overline{Q_D} \sum_{i=1}^{|D|} \mathbf{1}_{(V=i)} F(\mathcal{G}_i) & \text{otherwise.} \end{cases} \tag{11}$$

The algorithm based on $F(\mathcal{G})$ is called RVR simulation algorithm.

A procedure to obtain a trial of $F(\mathcal{G})$ can be summarized as follows:

Procedure RVR($\mathcal{G}$)

Input: network $\mathcal{G}$
Output: a random sample of r.v. $F(\mathcal{G})$

1. Check end recursion condition:
   1.1. Check if $\mathcal{G}$ is always $\mathcal{K}$-connected: If $|\mathcal{K}| = 1$ return(0).
   1.2. Check if the network is not $\mathcal{K}$-connected: If $\mathcal{G}$ is not $\mathcal{K}$-connected return(1).
2. Find a $\mathcal{K}$-extended cutset D of $\mathcal{G}$: $D = \{e_1, \ldots, e_{|D|}\}$.
3. Compute the probability that all components in D are failed: $Q_D = \prod_{i=1}^{|D|} q_{e_i}$.
4. Generate a trial v of V (with distribution $\Pr\{V = i\} = \Pr\{B_i\}/\overline{Q_D}$, $1 \leq i \leq |D|$).
5. Compute the corresponding network.
   5.1. Construct $\tilde{\mathcal{G}}_v = (\mathcal{G} - e_1 - e_2 - \ldots - e_{v-1})|e_v$.
   5.2. If in $\tilde{\mathcal{G}}_v$ there appears a link $l = (u_1, u_2)$ such that $p_l = p_{u_1} = p_{u_2} = 1$, contract l.
6. Recursive step: return($Q_D + \overline{Q_D} \times$ RVR $(\tilde{\mathcal{G}}_v)$).

To find a $\mathcal{K}$-extended cutset D of $\mathcal{G}$ efficiently, we propose the following procedure. Choose any terminal node $k \in \mathcal{K}$. Consider all the links incident to k, $l_i = (k, u_i)$, $1 \leq i \leq$degree(k). Then we can assume that either $q_{l_i} > 0$, or $q_{u_i} > 0$, or both (if this was not the case, as $q_k = 0$, the link l_i would have been contracted previously). We define $D = \{l_i = (k, u_i) | q_{l_i} > 0\} \cup \{u_i | \exists l_i = (k, u_i), q_{u_i} > 0\}$.

The order in which the elements are considered in steps 4 and 5 can be important from the viewpoint of ease of implementation, and also affect the efficiency of the procedure. In particular, it can happen that we mark as working a link such that one of its extremities was previously marked as failed; this doesn't affect the correctness of the algorithm, but can be seen as a waste of computational resources, as it does not contribute to effectively reducing the network. It may be interesting then to number first consecutively all links in D, and afterwards all nodes (in this way, links are always considered before nodes, and the above mentioned situation can't arise).

When computing a trial for $F(\mathcal{G})$, we generate a trial v of the r.v. V (task 4). Since $\mathbf{1}_{(V=i)}$ is equal to 0 for all $i \neq v$, then only the term $\mathbf{1}_{(V=v)}$ survives. It results that the computations involved in the recursion process can be represented by a linear computational structure. Its root corresponds to the network $\mathcal{G}$ under study, each internal node corresponds to a recursive call (task 6) and the last node presents a network that can be exactly evaluated (conditions 1.1 and 1.2 of the above procedure). Because at each recursive step the number of components of the network resulting from task 5 is diminished by at least 1, the size of the related linear structure is bounded by $|\mathcal{E}| + |\mathcal{V}|$.

4 Numerical experiments

In this section we present numerical experiments based on the dodecahedron network topology, shown in Figure 1. This network has been much used in the perfect-node literature (see for example [6, 5, 9, 7]). The number of nodes is 20, and the number of links 25. We will be interested in the source-to-terminal reliability measure (the source and terminal nodes are shown in Figure 1), and the relative efficiency of RVR with respect to crude Monte Carlo as a function of the node and link reliabilities. We have twenty five experiments, corresponding to all the combinations of taking independently node and link reliabilities in the set $\{0.9, 0.95, 0.98, 0.99, 0.999\}$.

We have run crude Monte Carlo and RVR algorithms with the same sample size. All the algorithms were run in a Sun Sparc Enterprise 250 workstation, with 256 Mb RAM memory and 250 Mhz processor, with operating system SunOS 5.7. The algorithms were implemented in C++; the compiler used was the GNU one (gcc 2.95) with code optimization for 64 bit processor. We denote by T_Y and T_Z the execution

times respectively of the crude Monte Carlo and of the RVR algorithms (measured in seconds). Their ratio, T_Y/T_Z, corresponds to the relative speed of Z with respect to Y (i.e, if T_Y/T_Z is less than 1, then Z is slower than Y).

Another important measure is the quotient $\mathrm{Var}\{Y\}/\mathrm{Var}\{Z\}$, which shows the accuracy improvement of RVR respect to crude Monte Carlo with the same number of iterations (if $\mathrm{Var}\{Y\}/\mathrm{Var}\{Z\} = v > 1$, then the variance of RVR is v times smaller than the variance of crude Monte Carlo).

Finally, the combined measure $W = \mathrm{Var}\{Y\}/\mathrm{Var}\{Z\} \times T_Y/T_Z$ (usually called the speedup of method Z with respect to method Y) as a single efficiency measure for comparing the algorithms.

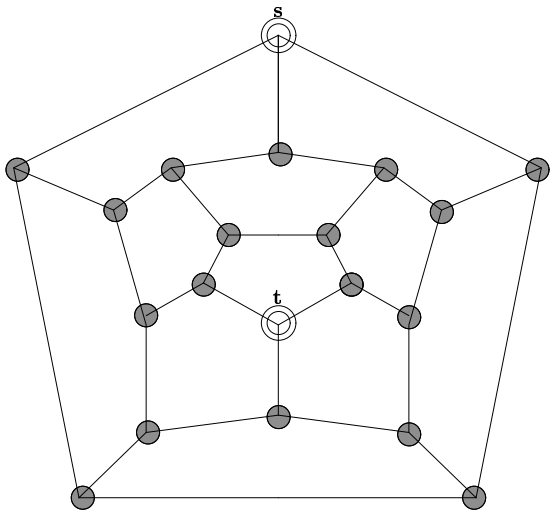

Figure 1: The "Dodecahedron" Network Topology

Figures 2 and 3 show the relative error for the crude Monte Carlo and the RVR algorithms. The relative error of crude Monte Carlo grows very rapidly with the reliability of the nodes of the network, and more slowyly with the link reliability. It has the highest error in networks with highly reliable nodes and links. On the other hand, the RVR algorithm has small relative error for these same networks, and higher error for networks with highly reliable nodes and not so reliable links. When the nodes are not very reliable, the relative error of RVR is relatively insensible to the link reliability.

Figure 4 shows the combined speedup measure W, which takes into account the variances and the execution times of both methods. We can see that the RVR method is in general more efficient than crude Monte Carlo, except for the case of highly reliable nodes and not so reliable links, where it has roughly the same or slightly worse performance than crude Monte Carlo.

Table 1 shows the detailed results for crude Monte Carlo and RVR for the dodecahedron network. We can see that the variances of RVR are always smaller than the variances of the crude Monte Carlo method, for the same number of iterations. On the other hand, the execution time of RVR is much bigger. The execution times for both methods do not change much for the different combinations of link and node reliabilities. The last column shows the combined speedup measure W.

5 Conclusions and future work

In this report, we have adapted the RVR method for reliability evaluation in networks with failures in nodes and links. We have shown that RVR has always smaller variance than crude Monte Carlo; this result is confirmed experimentally, showing that this difference is more important for highly reliable networks

r_u	r_l	$R(\mathcal{G})$	Var $\{Y\}$	T_Y	Var $\{Z\}$	T_Z	$\frac{\text{Var}\{Y\}}{\text{Var}\{Z\}}$	T_Y/T_Z	$S = \frac{\text{Var}\{Y\}}{\text{Var}\{Z\}} \times \frac{T_Y}{T_Z}$
0.9	0.9	0.7860	1.7E-05	4.7	7.2E-07	134.4	2.3E+01	3.5E-02	8.2E-01
0.95	0.9	0.8938	9.5E-06	4.9	3.5E-07	139.3	2.7E+01	3.5E-02	9.5E-01
0.98	0.9	0.9556	4.2E-06	5.0	2.4E-07	142.5	1.7E+01	3.5E-02	6.1E-01
0.99	0.9	0.9763	2.3E-06	5.0	1.7E-07	143.5	1.3E+01	3.5E-02	4.7E-01
0.999	0.9	0.9954	4.6E-07	5.1	1.0E-07	144.4	4.6E+00	3.5E-02	1.6E-01
0.9	0.95	0.8019	1.6E-05	4.8	1.9E-07	135.6	8.4E+01	3.5E-02	3.0E+00
0.95	0.95	0.9010	8.9E-06	5.0	7.4E-08	140.4	1.2E+02	3.5E-02	4.3E+00
0.98	0.95	0.9599	3.8E-06	5.1	7.1E-10	143.1	5.4E+03	3.5E-02	1.9E+02
0.99	0.95	0.9797	2.0E-06	5.1	9.8E-09	144.0	2.0E+02	3.5E-02	7.1E+00
0.999	0.95	0.9978	2.2E-07	5.1	2.0E-08	145.4	1.1E+01	3.5E-02	3.9E-01
0.9	0.98	0.8064	1.6E-05	4.7	7.2E-08	136.4	2.2E+02	3.5E-02	7.5E+00
0.95	0.98	0.9020	8.8E-06	5.0	1.1E-08	140.7	8.3E+02	3.5E-02	2.9E+01
0.98	0.98	0.9603	3.8E-06	5.1	1.4E-10	142.9	2.7E+04	3.5E-02	9.7E+02
0.99	0.98	0.9800	2.0E-06	5.1	2.1E-11	143.6	9.4E+04	3.5E-02	3.3E+03
0.999	0.98	0.9980	2.0E-07	5.1	7.6E-14	144.6	2.6E+06	3.5E-02	9.3E+04
0.9	0.99	0.8071	1.6E-05	4.8	4.2E-08	136.6	3.7E+02	3.5E-02	1.3E+01
0.95	0.99	0.9022	8.8E-06	4.9	2.1E-09	140.3	4.3E+03	3.5E-02	1.5E+02
0.98	0.99	0.9604	3.8E-06	5.1	9.3E-11	143.0	4.1E+04	3.5E-02	1.5E+03
0.99	0.99	0.9801	2.0E-06	5.1	6.8E-12	144.0	2.9E+05	3.5E-02	1.0E+04
0.999	0.99	0.9980	2.0E-07	5.1	2.0E-15	144.6	1.0E+08	3.5E-02	3.6E+06
0.9	0.999	0.8079	1.6E-05	4.8	2.5E-08	136.4	6.2E+02	3.5E-02	2.2E+01
0.95	0.999	0.9023	8.8E-06	5.0	1.8E-09	140.4	4.9E+03	3.5E-02	1.7E+02
0.98	0.999	0.9604	3.8E-06	5.1	3.1E-11	142.9	1.2E+05	3.5E-02	4.4E+03
0.99	0.999	0.9801	2.0E-06	5.1	3.5E-12	143.6	5.6E+05	3.5E-02	2.0E+04
0.999	0.999	0.9980	2.0E-07	5.1	3.3E-17	144.4	6.0E+09	3.5E-02	2.1E+08

Table 1: Results for the Dodecahedron network

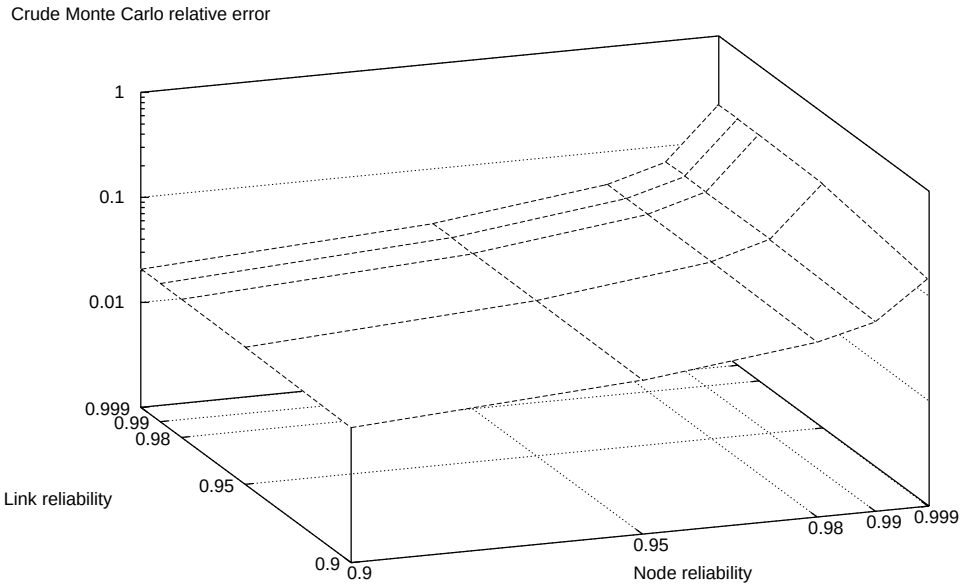

Figure 2: Relative error of Crude Monte Carlo mehod

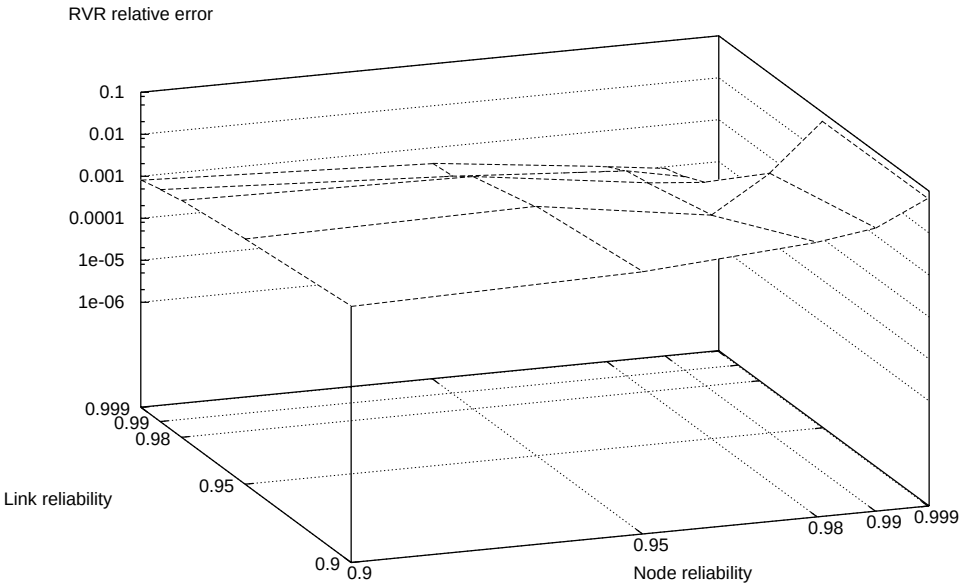

Figure 3: Relative error of RVR method

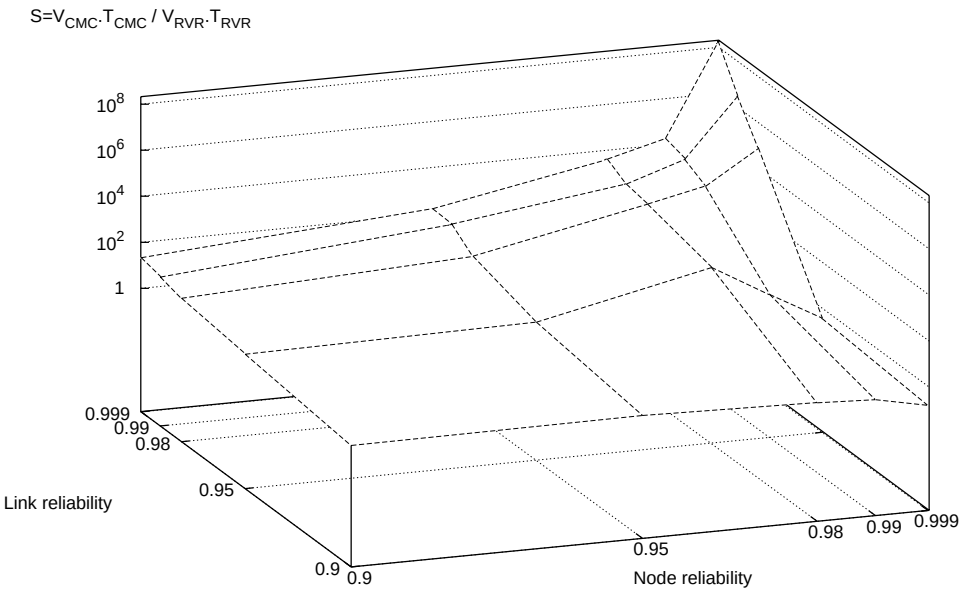

Figure 4: Performance index S

(specially in the case of highly reliable nodes). The accuracy improvement is obtained at the cost of higher execution times.

The tradeoff between these two different effects is reflected in the combined speedup measure; in the numerical example studied, the speedup is favorable to RVR method, except for the case of networks with highly reliable nodes and not so reliable links, where the difference in precision is smaller, and the overall speedup results in the CMC method being the more efficient one.

The performance of the RVR method depends on how the extended cutset D is chosen at each iterative step; future work is necessary to clarify the nature of this dependency, and to eventually find a rule for optimally determining D, in order to maximize the variance reduction attained by the method.

References

[1] M.O. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Trans. Reliab.*, R-35(3):230–239, August 1986.

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

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

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

[5] M. El Khadiri and G. Rubino. Reliability evaluation of communication networks. In *SAFECOMP'92, International Conference on Safety, Security and Reliability of Computers*, pages 279–283, Zurich, Suisse, October 1992.

[6] G.S. Fishman. A comparison of four Monte-Carlo methods for estimating the probability of s-t connectedness. *IEEE Trans. Reliab.*, R-35(2):145–155, June 1986.

[7] Sy-Yen Kuo, Shyue-Kung Lu, and Fu-Min Yeh. Determining terminal-pair reliability based on edge expansion diagrams using OBDD. *IEEE Transactions on Reliability*, 48(3):234–246, September 1999.

[8] A. Satyarayana and M.K. Chang. Network reliability and the factoring theorem. *Networks*, 13:107–120, 1983.

[9] S. Soh and S. Rai. Experimental results on preprocessing of path/cut term in the sum of disjoint products technique. *IEEE Transactions on Reliability*, 42, 1993.

[10] O.R. Theologou and J.G. Garlier. Factoring and reductions for networks with imperfect vertices. *IEEE Trans. on Reliab.*, 40(2):210–217, June 1991.

[11] D. Torrieri. Calculation of node-pair reliability in large networks with unreliable nodes. *IEEE Trans. on Reliab.*, 43(3):375–377, September 1994.