Comparison of Five Monte Carlo Methods to Estimate the Network Diameter Constrained Reliability

Leslie Murray
Universidad Nacional de Rosario, FCEIA,
Rosario, Argentina, 2000
leslie@eie.fceia.unr.edu.ar

and

Héctor Cancela
Universidad de la República, InCo, Facultad de Ingeniería
Montevideo, Uruguay, 11300
cancela@fing.edu.uy

Abstract
The exact calculation of network reliability by means of most of the classical operating criteria is an NP-hard problem. Monte Carlo methods are a simple, commonly accepted and some times very effective way to obtain network reliability estimations. This article reviews five Monte Carlo methods: Crude Monte Carlo, Dagger, Permutation Monte Carlo, and Cross-Entropy over Crude and Permutation, and applies them to the determination of the Diameter Constrained Reliability which is a generalization of the classical $K$–terminal function of structure. All these Monte Carlo methods were implemented for the estimation and analysis of three simple networks unreliability. The numerical performance of all the implementations was assessed, leading to some conclusions regarding their robustness and efficiency.

Keywords: Monte Carlo, Network Reliability, Diameter, Path Length, Variance Reduction.
1 Introduction

A communication network is said to be reliable if it can achieve the purpose for which it is committed. To measure how well these purpose is achieved, it is necessary to define an operating criterium and to say how near or far the network is to be operative under this criterium. In a mathematical sense the reliability is defined as the probability that the network is not operative or failed under the same criterium.

Let the undirected graph $G = (V, E)$ be the model of a communication network, being $V$ a set of $n$ absolutely reliable nodes and $E$ a set of $m$ unreliable independent links. Let the vector $X = (X_1, X_2, \ldots, X_m)$ denote the state of the links such that $X_i = 1$ means that the $i^{th}$ link is operative while $X_i = 0$ means that the $i^{th}$ link is failed. Let $f_X(x)$ be the probability mass function of vector $X$. According to $f_X(x)$ (formed by independent components $f_{X_i}(x))$, $r_i = P[X_i = 1]$ is the single link reliability, whereas $q_i = 1 - r_i = P[X_i = 0]$ is the single link unreliability.

A function $\Phi(X)$, called structure function, denotes the state of the whole network if it reflects some operative criteria. The behavior of $\Phi$ is therefore much like the behavior of every $X_i$ in the sense that $\Phi = 1$ means that the network is operative, and $\Phi = 0$ means that the network is failed. The definitions of network reliability $R$ and network unreliability $Q$ fall then naturally as: $R = P[\Phi(X) = 1]$ and $Q = P[\Phi(X) = 0]$. It is clear also that when $X$ follows $f_X(x)$, $R = E\{\Phi(X)\}$ and $Q = E\{1 - \Phi(X)\} = 1 - R$.

According to the network purpose, there are different ways to define $\Phi(X)$, i.e. to define the meaning of operative and failed for the whole network. One way is to say that $\Phi = 1$ if two selected nodes $s$ and $t$ in $V$ are connected by operative links, and $\Phi = 0$ otherwise. Another way is to say that $\Phi = 1$ if all nodes in $K$, being $K \subseteq V$, are connected, and $\Phi = 0$ otherwise (from now on, unless stated specifically, “connected” means connected by operative links). The determination of network reliability by means of these two approaches is known respectively as the source-terminal reliability and the terminal reliability problem (is clear that the source-terminal problem is the $K$–terminal problem, with $|K| = 2$). None of these definitions consider the length of the path between the pairs of connected nodes. The Diameter Constrained Reliability [12] is a generalization of the $K$–terminal reliability concept in which $\Phi = 1$ if every pair of nodes in $K$ ($K \subseteq V$) is connected through a path of length $D$ or less, and $\Phi = 0$ otherwise$^1$. $D$ is a constant, and the definition holds either for $|K| = 2$ or $|K| > 2$. Diameter Constrained Reliability is particularly useful when the messages that travel through the network ought to reach the target node with a delay not greater than $DT$, being $T$ the single link delay.

The exact evaluation of a network reliability as $R = E\{\Phi(X)\}$ (unreliability as $Q = E\{1 - \Phi(X)\} = 1 - R$) shown below is an NP–hard problem, even under exact knowledge of $f_X(x)$ and no matter whether the structure function corresponds to $K$–terminal or Diameter Constrained [7]:

$$\begin{align*}
R &= E\{\Phi(X)\} = \sum_{i=1}^{2^m} P[X^i] \Phi(X^i) \\
Q &= E\{1 - \Phi(X)\} = \sum_{i=1}^{2^m} P[X^i] (1 - \Phi(X^i))
\end{align*}$$

(1)

where $X^i$ is every possible value of $X$. For networks with a high number of nodes and links it is usually more attainable to estimate the values of $R$ or $Q$. Monte Carlo Methods are a simple, commonly accepted and some times very effective way to make such estimations.

In Section 2 we present the concept of Diameter Constrained Reliability [1]. The following five sections review five Monte Carlo methods that were proposed in the literature for the estimation of the classical $K$–terminal reliability measure but can be easily be employed to estimate the Diameter Constrained Reliability: Crude Monte Carlo [6], Dagger [4, 10], Permutation Monte Carlo [5, 8, 9], and Cross-Entropy over Crude and Permutation Monte Carlo [2, 3, 9, 8]. Section 8 shows some experimental results obtained when applying the methods to the estimation of the unreliability of three elementary networks. Some conclusions are issued at the end of the article.

2 Diameter Constrained Reliability

According to the network purpose, there are different ways to define the structure function $\Phi(X)$. The classical approach is defined by the $K$–terminal reliability problem. In this case, the network is considered

\footnote{The length of a path is measured in number of links, i.e. the links have unitary lengths.}
operative ($\Phi(X) = 1$) if all nodes in $K \subseteq V$ are connected, whereas if not all nodes in $K$ are connected, the network is considered not operative or failed ($\Phi(X) = 0$).

The Diameter Constrained Reliability [12] is a generalization of the $K$-terminal problem in which the concept of connected is replaced by $D$-connected with the meaning that “two nodes are $D$-connected if there is a path made by $D$ or less operative links between them”. Upon this definition, $\Phi(X) = 1$ if all nodes in $K \subseteq V$ are $D$-connected while, if not all nodes in $K$ are $D$-connected, $\Phi(X) = 0$. If any message takes a time (delay) $T$ to go through a single link, the Diameter Constrained Reliability is the probability that any message can travel between any two pair of nodes in $K$ in time not longer than $DT$.

The following algorithm implements the structure function for the Diameter Constrained Reliability problem. It performs $D$ iterations of a Breadth First Search from every node in $K$. If after the $D^{th}$ iteration all the other $K-1$ nodes are visited, $\Phi(X) = 1$, otherwise $\Phi(X) = 0$.

```
for(all nodes s of network){
    enqueue_Q(s)
    num=1
    do D times{
        while(queue_Q not empty){
            d=dequeue_Q()
            mark d as visited
            for(all neighbors n of d)
                if(n not visited){
                    mark n as visited
                    enqueue_B(n)
                    num=num+1
                }
        }
        while(queue_B not empty)
            enqueue_Q(dequeue_B())
    }
    if(num < number of nodes of network){
        "network is not D-connected"
        exit
    }
} "network is D-connected"
```

Up to our knowledge, the literature does not contain studies of Monte Carlo methods for estimating the Diameter Constrained Reliability measure; only exact (exponential time in the general case) algorithms and also some bounds have been studied.

3 Crude Monte Carlo

The simplest Monte Carlo algorithm (called Standard or Crude Monte Carlo) consists in sampling $N$ independent values $X^{(i)}$ from $f_X(x)$. The estimations for $R$ and $Q$ are then:

$$\hat{R} = \frac{1}{N} \sum_{i=1}^{N} \Phi(X^{(i)}) \quad \hat{Q} = \frac{1}{N} \sum_{i=1}^{N} (1 - \Phi(X^{(i)}))$$

Both $\hat{R}$ and $\hat{Q}$, are unbiased estimators of $R$ and $Q$ respectively. The main weakness of these estimators is that their relative precision is low, especially when the measure to estimate is very small (near 0). In the case of $\hat{Q}$, the exact variance is $\mathbb{V}(\hat{Q}) = Q(1-Q)/N$, and the relative error, as expressed by the coefficient of variation, is $\sqrt{\mathbb{V}(\hat{Q})/\mathbb{E}(\hat{Q})} = \sqrt{(1-\hat{Q})/((N-1)\hat{Q})}$ (an unbiased estimator of $\mathbb{V}(\hat{Q})$ is $\hat{Q}(1-\hat{Q})/(N-1)$). Hence, for highly reliable networks, where $\hat{Q}$ is supossed to be small, an accurate estimation requires a large sample size $N$.

The following methods attempt to improve Crude Monte Carlo, mostly by attaining a variance reduction. In this article all the methods are developed for the estimation of the network unreliability $\hat{Q}$.
4 Dagger

The variance of the Crude Monte Carlo estimation is \( Q(1 - Q)/N \), if and only if all samples \( X^{(i)} \) are independent. If the samples are correlated, the resulting estimator \( \hat{Q}' \) has the same mean than \( Q \), but a different variance:

\[
\mathbb{V} \{ \hat{Q}' \} = Q(1 - Q)/N + \frac{2}{N^2} \sum_{1 \leq i < j \leq N} \mathbb{Cov}(\Phi(X^{(i)}), \Phi(X^{(j)})�).
\]

A smart sampling plan can make the sum of covariances to be negative and, consequently, \( \mathbb{V} \{ \hat{Q}' \} < \mathbb{V} \{ \hat{Q} \} \). To find such a sampling plan is not an easy task. It is well known that two variables sampled from \( U \) and \( 1 - U \) respectively, being \( U \) a Uniform random number generator in \([0, 1]\), are negatively correlated. If every set of \( m \) samples \((X_1, X_2, \ldots, X_m)\) is alternatively sampled from \( U \) and \( 1 - U \), compared to the Crude Monte Carlo scheme, one half of the accesses to the random number generator is done. It can be shown also that for any link \( k \) with single reliability \( r_k \), if two consecutive samples \((X^{(i)}_k, X^{(j)}_k)\) are taken from \( U \) and \( 1 - U \) by means of the following algorithm:

\[
\text{if } U < r_k \quad X^{(i)}_k = 1 \quad \text{else } X^{(i)}_k = 0, \\
\text{if } (1 - U) < r_k \quad X^{(j)}_k = 1 \quad \text{else } X^{(j)}_k = 0;
\]

then \( \mathbb{Cov}(X^{(i)}_k, X^{(j)}_k) = -q^2_k \), where:

\[
q_k = \begin{cases} 
\quad r_k & \text{if } r < 0.5 \\
1 - r_k & \text{if } r \geq 0.5
\end{cases}
\]

Since \( \Phi(X) \) is monotone, a negative covariance on \((X^{(i)}_k, X^{(j)}_k)\) causes a negative covariance on the pair \((\Phi(X^{(i)}_k), \Phi(X^{(j)}_k))\), which is the condition required by (3) to achieve a variance reduction.

These are the basis of the Antithetic Variables Method: to create a pair of negatively correlated samples out of each single sample \( U \). The Dagger algorithm \([4, 10]\), attempts to create more than two samples out of each single one, being therefore a generalization of the Antithetic Variables Method. To do this, the interval \([0, 1]\) is divided into as many subintervals as \( q_k \) fits into \([0, 1]\), say \( L_k \), leaving at most one interval shorter than \( q_k \). Calling \( I^1, i = 1, 2, \ldots, L_k \), every subinterval defined as \( I^1 = [0, q], I^2 = [q, 2q], \ldots \), and defining \( X_{k-\text{default}} \) as:

\[
X_{k-\text{default}} = \begin{cases} 
\quad 0 & \text{if } r_k < 0.5 \\
\quad 1 & \text{if } r_k \geq 0.5
\end{cases}
\]

the following algorithm creates \( L_k \) correlated samples of \( X_k \) out of only one value of \( U \):

\[
X^{(i)}_k = (1 - X_{k-\text{default}}) \cdot 1_{(U \in I^i)} + X_{k-\text{default}} \cdot 1_{(U \notin I^i)} \quad i = 1, 2, \ldots, L_k
\]

The single reliability may not be the same for all the links. In the extreme case, the reliability of all the links can be different. The number of samples \( L_k \), \( k = 1, 2, \ldots, m \), that can be created out of every single value \( U \) will be different then. The way to proceed in the Dagger algorithm is to determine \( L \) as the least common multiple of the \( m \) \( L_k \) values, and to see how many times every \( L_k \) fits into \( L \). This relation, \( n_k = L/L_k \) indicates the number of \( U \) samples necessary to generate \( L \) samples for every link.

The variance reduction does not go further than \( q^2_k \) for each link (being the whole reduction \( q^2 \) for the case where \( q \) is the single unreliability for all links). To choose high values of \( L_k \) (even the highest possible value) does not yield a variance reduction increase. However as the values of \( L_k \) grow, the number of accesses to the random numbers generator fall (improving computational efficiency).

5 Permutation Monte Carlo

In Crude Monte Carlo the links are checked without a time consideration. The parameters \( r_i = P[X_i = 1] \) or \( q_i = 1 - r_i = P[X_i = 0] \) do not mean anything about an observation instant, they are rather a statistical concept. In Permutation Monte Carlo \([8, 9]\), the state of the links is determined assuming that they are all failed at time \( t = 0 \) and they become operative at time \( \tau(i) \), \( i = 1, 2, \ldots, m \), exponentially distributed with parameter \( \lambda(i) \), that is: \( P[\tau(i) \leq t] = 1 - e^{-\lambda(i)t} \). Vector \( X = (X_1, X_2, \ldots, X_m) \) becomes then
\( \mathbf{X}(t) = (X_1(t), X_2(t), \cdots, X_m(t)) \), a Markov Process called \( \text{ConstructionProcess} \) \cite{5}. Since \( (1 - e^{-\lambda(i)t}) \) is the probability that the \( i^{th} \) link becomes operative at time \( t \) or earlier, if \( \lambda(i) = -\log(q_i) \), with \( q_i = 1 - r_i \), the probability that the \( i^{th} \) link is operative at \( t = 1 \) is exactly the single link reliability, \( r_i \). As a consequence, the probability that the whole network is found to be \emph{operative (failed)} at \( t = 1 \) is exactly \( R \) (\( Q \)).

\[
R = \mathbb{E}\{\Phi(\mathbf{X}(1))\} \quad Q = \mathbb{E}\{1 - \Phi(\mathbf{X}(1))\} \tag{5}
\]

However, an experiment where the links state goes from \emph{failed} to \emph{operative} at times \( \tau(i), i = 1, 2, \cdots, m \), and the operating conditions of the network are checked by means of \( \Phi(\mathbf{X}(1)) \), is nothing but a different implementation of \emph{Crude Monte Carlo}. The real profit of \emph{Permutation Monte Carlo} comes from sampling feasible \emph{permutations}, assessing the probability that each sampled \emph{permutation} puts the network operative at \( t = 1 \) and then averaging all the assessed probabilities. The meaning of \emph{permutation} is the following: at every realization of the \emph{ConstructionProcess}, a specific order in which the links become operable is obtained; this order is a random variable \( \Omega \), taking values in the space of permutation of \( \mathcal{E} \). In every permutation links are arranged like: \( (e_1^1, e_2^1, e_3^1, \cdots, e_m^1) \), being \( \tau(i) < \tau(j) < \tau(k) < \cdots < \tau(l) \). Upon this notation, \( e_y^x \) means that the link \( x \) occupies the \( y^{th} \) position within the permutation \( (\lambda^y \text{ and } \tau^y \text{ can also be used to refer } \lambda(x) \text{ and } \tau(x) \text{ respectively}).

From the account of links that are still \emph{failed} in every subinterval, it is possible to define:

\[
\mathcal{E}_0 = \mathcal{E} \quad 0 \leq t < \tau^1
\]
\[
\mathcal{E}_i = \mathcal{E}_{i-1} - e^i \quad \tau^i \leq t < \tau^{i+1} \quad i = 1, 2, \cdots, m - 1
\]
\[
\mathcal{E}_m = \emptyset \quad \tau^m \leq t
\]

Then:

\[
\mathbb{P}[\Omega = \omega] = \prod_{i=1}^{m} \frac{\lambda^i}{\lambda(\mathcal{E}_{i-1})} \tag{6}
\]

where \( \lambda(\mathcal{E}_i) = \sum_{e^i \in \mathcal{E}_i} \lambda^j = \sum_{j=i+1}^{m} \lambda^j \).

In every \emph{permutation} \( \omega \) there is an element \( b(\omega) \in \omega \), so-called \emph{critical number}, that references the link due to which the network reaches the \emph{operative} condition:

\[
b(\omega) = \min\{i : \Phi(\mathbf{X}(\tau(i))) = 1\} \tag{7}
\]

Conditioning (5) to the \emph{permutation} \( \Omega \), the value of \( Q \) results:

\[
Q = \mathbb{E}\{1 - \Phi(\mathbf{X}(1))\} = \sum_{\omega} \mathbb{P}[\Phi(\mathbf{X}(1)) = 0 \mid \Omega = \omega] \mathbb{P}[\Omega = \omega] \tag{8}
\]

Considering that the times \( A_i = \tau(i) + 1 - \tau(i), i = 0, 1, \cdots, m - 1 \) (accepting \( \tau(0) = 0 \)) are exponentially distributed with parameter \( \lambda(\mathcal{E}_i) \), then:

\[
\mathbb{P}[\Phi(\mathbf{X}(t)) = 0 \mid \Omega = \omega] = \mathbb{P} \left[ A_0 + A_1 + \cdots + A_{b(\omega) - 1} > t \mid \Omega = \omega \right] \tag{9}
\]

\[
= \mathbb{P} \left[ \text{Conv} \{e^{\lambda(\mathcal{E}_{b(\omega)-1})t}\} > t \right] \tag{10}
\]

The conditional on \( \Omega \) of (9) is not necessary in (10) because the values \( \mathcal{E}_i \) actually depend on the \emph{permutation}. Setting \( t = 1 \), the left term in (9) defines a function of \( \Omega \), \( G(\Omega) \):

\[
G(\omega) = \mathbb{P}[\Phi(\mathbf{X}(1)) = 0 \mid \Omega = \omega] \tag{11}
\]

then:

\[
Q = \mathbb{E}\{G(\Omega)\} = \sum_{\omega} G(\omega) \mathbb{P}[\Omega = \omega], \tag{12}
\]

where the mean \( \mathbb{E}\{G(\Omega)\} \) can be estimated by Monte Carlo:

\[
\hat{Q} = \frac{1}{N} \sum_{i=1}^{N} G(\Omega^{(i)}) \tag{13}
\]
The estimation of $Q$ is finally done as the average of a set of probabilities, actually the probabilities that, given a certain permutation, the network is operative not later than $t = 1$. The scenario is quite different from Crude Monte Carlo, where a Bernoulli variable is averaged ($\Phi = 1$ if the network is operative and $\Phi = 0$ if it is failed).

The variance of the estimated variable $Q$ in Crude Monte Carlo is then:

$$\mathbb{V}_C\{Q\} = Q(1 - Q),$$

(14)

whereas the variance of the estimated variable $Q$ in Permutation Monte Carlo is:

$$\mathbb{V}_P\{Q\} = \sum_{\omega} G(\omega)^2 \mathbb{P}[\Omega = \omega] - \left(\sum_{\omega} G(\omega) \mathbb{P}[\Omega = \omega]\right)^2$$

(15)

Replacing (12) in (14) results:

$$\mathbb{V}_C\{Q\} = \mathbb{V}_P\{Q\} + \sum_{\omega} G(\omega) \mathbb{P}[\Omega = \omega] (1 - G(\omega))$$

(16)

This proves that $\mathbb{V}_P\{Q\} \leq \mathbb{V}_C\{Q\}$.

In the rest of this section some guidelines for the Permutation Monte Carlo implementation are introduced. The key is to find the way to sample the values $G(\Omega(i))$. Permutations only refer to the order in which links passes from failed to operative, but do not specify the times in which these commutations take place. Actually, two realizations of the same permutation may put the network operative before and after $t = 1$. The whole mechanism can be summarized as follows:

1. Generate a random permutation $\omega$, sampling the times $\tau(n)$, $n = 1, 2, \ldots, m$ from exponential distributions with parameters $\lambda(n)$ and arrange them by increasing order of time. That is:

$$\left(e_1^1, e_2^1, e_3^1, \ldots, e_m^1\right) \text{ if } \tau(i) < \tau(j) < \tau(k) < \cdots < \tau(l)$$

2. Identify the critical number, applying successively the structure function to a network to which one link at the time is added. If the links are added in the order pointed by the sampled permutation, as soon as the network becomes operative, the order of the last added link is the critical number.

3. Calculate the following distribution:

$$\text{Conv}_{0 \leq t \leq h(\omega)^{-1}} \{e^{\lambda(\omega)^{j} t}\}$$

This distribution is a sum of exponentials. Determine then the corresponding cumulative distribution, replacing every $\lambda(i) e^{\lambda(i)^{j} t}$ by $(1 - e^{\lambda(i)^{j} t})$. Calculate then the resulting function in $t = 1$ and accumulate the complement of this result.

4. Divide the accumulated values by the number of values accumulated and the result is the estimation of the network unreliability, $Q$.

6 Cross-Entropy over Crude Monte Carlo

Importance Sampling [6] helps to deal with the estimation of rare events probability. If the problem can be reduced to the estimation of the mean of some function of a random variable, $g(X)$, where $X$ follows the distribution $f(x)$, then $f(x)$ is the distribution to sample from in the estimation. If for any reason sampling from $f(x)$ is inefficient, the problem can be transformed as follows, by a change of measure (a change in the distribution to sample from, with the consequent change in the estimated function):

$$\mathbb{E}_f\{g(X)\} = \int_{-\infty}^{+\infty} g(x) f(x) \, dx = \int_{-\infty}^{+\infty} g(x) \frac{f(x)}{h(x)} \, h(x) \, dx = \mathbb{E}_h \left\{ \frac{g(X) f(X)}{h(X)} \right\}$$

(17)

where $h(x) \neq 0$ unless where $f(x) = 0$. Sub indexes $(f$ or $h$) in $\mathbb{E}\{}$ points out the distribution to sample the $X$ values from. It is simple to prove that if $h(x)$ is:

$$h(x) = \frac{g(x) f(x)}{\mathbb{E}_f\{g(X)\}}$$

(18)
then:
\[
\forall h \left\{ \frac{g(x)f(x)}{h(x)} \right\} = 0
\]  

(19)

This is of course impossible, because the value of (18) makes use of \( \mathbb{E}_f\{g(x)\} \) which is precisely the value to estimate. \textit{Cross-Entropy}, [2, 3, 9, 8], proposes to find an \( h(x) \) somehow considered the “nearest” to the value in (18), under some measure of distance. The measure of distance proposed is \textit{The Kullback-Leibler Distance} \(^2\), also known as \textit{Cross-Entropy}.

A rare event can be formulated in many ways. One that is suitable for the problem that this paper deals with, is:

\[
\ell = \mathbb{P}_u[S(X) \geq \gamma]
\]

(20)

where \( S \) is a real function and \( \gamma \) a constant. Being \( f(x, v) \) a family of distributions with parameter \( v \), variable \( X \) follows the distribution \( f(x, u) \) where \( u \) is a particular value of the parameter \( v \) (thereby subindex \( u \) in \( \mathbb{P} \)). If \( \ell \) is “very” small, \( S(X) \geq \gamma \) is a rare event. The probability \( \ell \) can be also presented as:

\[
\ell = \mathbb{E}_u\{I_{\{S(X) \geq \gamma\}}\}
\]

(21)

where \( I_{\{S(X) \geq \gamma\}} = 1 \) if \( S(X) \geq \gamma \) and 0 otherwise. An unbiased estimator for \( \ell \) is:

\[
\hat{\ell}_u = \frac{1}{N} \sum_{i=1}^N I_{\{S(X^{(i)}) \geq \gamma\}}
\]

(22)

where \( X^{(i)} \) values are sampled from \( f(x, u) \). Being \( \ell \) a rare event under \( f(x, u) \), an accurate estimation would require a very high value of \( N \). Applying \textit{Importance Sampling}:

\[
\hat{\ell}_{u,g} = \frac{1}{N} \sum_{i=1}^N I_{\{S(X^{(i)}) \geq \gamma\}} \frac{f(X^{(i)}, u)}{g(X^{(i)})}
\]

(23)

where the \( X^{(i)} \) values are sampled from \( g(x) \). Just like in (18), the “ideal” value of \( g(x) \) would be \( g^*(x) \):

\[
g^*(x) = \frac{I_{\{S(X(x)) \geq \gamma\}} f(x, u)}{\ell}
\]

(24)

Instead of \( g^*(x) \), \textit{Cross-Entropy} proposes one of the functions in the family of distributions \( f(x, v) \), say \( f(x, v^*) \). \( v^* \) should then make \( f(x, v^*) \) to be the nearest to \( g^*(x) \), out of all distributions in the family \( f(x, v) \), by means of the \textit{The Kullback-Leibler Distance}. It can be proved that:

\[
v^* = \arg\max_v \mathbb{E}_w \{I_{\{S(X(x)) \geq \gamma\}} W(X, u, w) \ln f(X, v)\}
\]

(25)

where \( W(x, u, w) = f(x, u)/f(x, w) \) comes from a new application of \textit{Importance Sampling}. If \( \mathbb{E}_w \{\} \) is replaced by an estimation:

\[
v^* = \arg\max_v \frac{1}{N} \sum_{i=1}^N I_{\{S(X^{(i)}) \geq \gamma\}} W(X^{(i)}, u, w) \ln f(X^{(i)}, v)
\]

(26)

The \( X^{(i)} \) values are now sampled from \( f(x, w) \). There is also an iterative process that converges to \( v^* \):

\[
v_k = \arg\max_v \sum_{i=1}^N I_{\{S(X^{(i)}) \geq \gamma\}} W(X^{(i)}, u, v_k) \ln f(X^{(i)}, v)
\]

(27)

In the particular case where the distributions of the family \( f(x, v) \) are exponentials, expression (27) can be expressed, by components, as:

\[
v_{k,j} = \frac{\sum_{i=1}^N I_{\{S(X^{(i)}) \geq \gamma\}} W(X^{(i)}; u, v_{k-1}) X^{(i)}_j}{\sum_{i=1}^N I_{\{S(X^{(i)}) \geq \gamma\}} W(X^{(i)}; u, v_{k-1})}
\]

(28)

\(^2\)Being \( d_1(X) \) and \( d_2(X) \) two probability density functions, the \textit{Kullback-Leibler Distance} between them is:

\[
D(d_1, d_2) = \mathbb{E}_d_1 \{\ln(d_1(X)/d_2(X))\} = \mathbb{E}_d_1 \{\ln d_1(X)\} - \mathbb{E}_d_1 \{\ln d_2(X)\} = \int \ln d_1(x) d_1(x) \, dx - \int \ln d_2(x) \, dx
being $v_k = (v_{k,1}, v_{k,2}, \cdots, v_{k,m})$ and:

$$W(x; u, v) = \frac{f(x, u)}{f(x, v)} = \exp \left( -\sum_{j=1}^{m} X_j \left( \frac{1}{u_j} - \frac{1}{v_j} \right) \right) \prod_{j=1}^{m} \frac{v_j}{u_j}$$

Once $v^*$ is finally determined, $\ell$ is estimated as:

$$\hat{\ell} = \frac{1}{M} \sum_{i=1}^{M} I_{\{S(x^{(i)}(1)) \geq 1\}} W(X^{(i)}; u, v^*)$$

where the values of $X^{(i)}$ are sampled from $f(x, v^*)$.

Cross-Entropy can be applied to yield a change of measure in the network unreliability estimation by Crude Monte Carlo. To do this, the Crude Monte Carlo implementation outlined in Section 5 must be considered. Calling $T = (\tau(1), \tau(2), \cdots, \tau(m))$ the set of commutation times for the $m$ links, the time when the whole network becomes operative is a function of $T$, say $S(T)$, then:

$$Q = P[S(T) \geq 1]$$

The distribution to sample the times $\tau(i), i = 1, 2, \cdots, m$ is taken from the family of exponentials $f(t, v)$, with $v = u$, being $u$ the set of means: $u = (1/\lambda(1), 1/\lambda(2), \cdots, 1/\lambda(m))$. The only modification that can be made to the procedure introduced so far is not to set $\gamma = 1$ from the very beginning but to iterate on a $\gamma_k$, until it reaches the value 1, instead. So, for every $v_{k-1}$, a new $\gamma_k$ must be obtained as the $(1 - \rho)$ quantile of $S(T)$ (this means that $P_{v_{k-1}}[S(T) \geq \gamma_k] \geq \rho$ and $P_{v_{k-1}}[S(T) \leq \gamma_k] \geq (1 - \rho)$).

The procedure can be finally summarized as follows:

1. Set $k = 1$ and $v_{k-1} = u$.
2. Sample $N$ values $T^{(1)}, T^{(2)}, \cdots, T^{(N)}$ from $f(t, v_{k-1})$ and, with these samples, determine $\gamma_k$.
3. Calculate $v_k$ by means of expression (28) with $X = T$.
4. If $\gamma_k < 1$ do $k = k + 1$ and go back to 2, otherwise go to 5.
5. Estimate $Q$ as:

$$\hat{Q} = \frac{1}{M} \sum_{i=1}^{M} I_{\{S(T^{(i)}) \geq 1\}} W(T^{(i)}; u, v_K)$$

where $K$ is the number of iterations required, that is the las value of $k$ and $M$ the sample size to make the estimation.

7 Cross-Entropy over Permutation Monte Carlo

While for highly reliable networks Crude Monte Carlo can be transformed into a “rare event estimation problem”, Permutation Monte Carlo must be undertaken as the “estimation of a mean that is close to 0”. Actually, the whole development of Cross-Entropy, [8, 9], is started from the determination of a mean (see 21). The target is then to find, in Permutation Monte Carlo, the function to replace $I_{\{S(X) \geq 1\}}$ or, say, $I_{\{S(T) \geq 1\}}$ in (21).

In (12) it was shown that:

$$Q = E\{G(\Omega)\} = \sum_{\omega} G(\omega) P[\Omega = \omega]$$

where:

$$G(\omega) = P[\Phi(X(1)) = 0 | \Omega = \omega]$$

But in fact $\Omega = \Omega(T)$ because permutations are determined by the set of commutations time $T = (\tau(1), \tau(2), \cdots, \tau(m))$. And, being $f(t, u)$ the set of exponentials to sample from, belonging to the family $f(t, v)$ with $v = u$ and $u = (1/\lambda(1), 1/\lambda(2), \cdots, 1/\lambda(m))$, then:

$$\hat{Q} = E_u\{G(\Omega(T))\} = E_u\{F(T)\}$$

where $F(T) = G(\Omega(T))$. The whole procedure results:

\textsuperscript{3}The value of $\rho < 1$ must be not too small, in the order of $10^{-2}$. 

8
1. Set \( k = 1 \) and \( v_{k-1} = u \).

2. Sample \( N \) values \( T^{(1)}, T^{(2)}, \ldots, T^{(N)} \) from \( f(t, v_{k-1}) \) and, with these samples, determine \( v_k \) by means of the following expression:

\[
v_{k,j} = \frac{\sum_{i=1}^{N} F(T^{(i)}; u, v_{k-1}) \tau(j)^{(i)}}{\sum_{i=1}^{N} F(T^{(i)}; u, v_{k-1})}
\]

where \( v_k = (v_{k,1}, v_{k,2}, \ldots, v_{k,m}) \) and:

\[
W(t; u, v) = f(t, u) / f(t, v) = \exp \left( -\sum_{j=1}^{m} \tau(j) \left( \frac{1}{u_j} - \frac{1}{v_j} \right) \right) \prod_{j=1}^{m} \frac{v_j}{u_j}
\]

3. If the values of \( v_k \) are not “stabilized” do \( k = k + 1 \) and go back to 2, otherwise go to 4.

4. Estimate \( Q \), the mean of \( F(T) \), as:

\[
\hat{Q} = \frac{1}{M} \sum_{i=1}^{M} F(T^{(i)}; u, v_K)
\]

where \( K \) is the number of iterations required, that is the last value of \( k \), and \( M \) is the sample size to make the estimation.

8 Numerical Results

In this section, the Monte Carlo methods introduced so far are applied to estimate and analyze the Diameter Constrained Reliability of the networks shown in Figure 1: a triangular (TRI), a squared grid (GRID) and an hexagonal topology (HEX). In all cases \( K = V \). The implementation follows the ideas discussed in each section, and the structure function employed is the one presented in Section 2.

TRI

GRID

HEX

Figure 1: Topologies of Networks Under Analysis

The first set of experiments proceeds in three stages. A particular value of \( D \) is selected for every network and a single link reliability value is set in every stage (the same for every link in all the networks). In all the cases a sample size of \( 10^5 \) is used to estimate the unreliability \( Q \), together with an estimation of the Standard Deviation. Tables 1, 2 and 3 show the results. As expected, the worst estimation is made by Crude Monte Carlo, in every case. Dagger does not improve accuracy, its error has the same order of magnitude of Crude Monte Carlo. The real improvement comes up with Permutation Monte Carlo. The ratio of Standard Deviations between Permutation and Crude is in the order of 2 for link reliabilities of 0.80 and 20 for link reliabilities of 0.99. Cross-Entropy produces in both, Crude and Permutation, a slight improvement with respect to the “not Cross–Entropy” versions. An additional test, not shown in the tables, points out that (for link reliabilities of 0.90) the same accuracy level is obtained with the following orders of sample sizes: Crude Monte Carlo: \( 10^5 \), Dagger: \( 10^6 \), Permutation Monte Carlo: \( 2 \times 10^3 \), CE–Crude Monte Carlo: \( 10^4 \) and CE–Permutation Monte Carlo: \( 2 \times 10^3 \).

The second set of experiments concerns the value of \( D \). Numerically the scope of this parameter can be: \( 0 < D < \infty \). However from a practical point of view, different bounds should be considered: \( D_d < D < D_u \).
To choose $D_d$, the length of the shortest path between the two nodes that are the most far apart, has to be found. It is clear to see that a path of length 1 can not guarantee connectivity between many pairs of nodes in TRI and and HEX. Actually a minimum of 2 is necessary for these two networks. In the case of GRID, the nodes at the diagonal vertexes are at a distance of 4 from the opposite node, being therefore $D_d = 4$. On the other side, there is a longest path between any pair of nodes, in every network (which will always be of length at most $V - 1$). These paths are shown in Figure 2, corresponding values of 5, 8 and 6 for TRI, GRID and HEX respectively. If $D < D_d$, the value of $Q$, the Diameter Constrained unreliability, will be always 1.00, indicating a permanent fail condition, for whatever set of sampled states for the links. On the other side, for any value of $D > D_d$ the values of $Q$ will be always the same.

![Figure 2: Examples of the Longest Path for Every Network](image)

Tables 5 and 4 show the estimations of $Q$, together with an estimation of the Standard Deviation, for two Monte Carlo methods, Crude and Permutation. The three networks are tested in the corresponding range of $D_d < D < D_u$. The results shows that the values of $Q$ fall as $D$ grows higher. This is predictable for any network because as $D$ grows, the event of finding the network in a failed condition becomes more rare. Estimations of such events are also more accurate. Besides, as the value of $D$ grows higher, the Standard Deviations falls, for both methods. This fall is slightly more pronounced in Permutation than Crude.
The set of experiments shown in Tables 1, 2 and 3 reported execution times that, considered as an average for every method, and measured in units taking the Crude Monte Carlo as the reference are: Crude Monte Carlo: 1.00, Dagger: 1.20, Permutation Monte Carlo: 3.90, Cross-Entropy Crude: 2.70 and Cross-Entropy Permutation: 4.00.

All experiments were performed by programs written in C (gcc compiler), making use of the Mersenne Twister [11] as the random numbers generator. As pointed by [9], to guarantee convergence and avoid oscillations, the value of ρ has to be adapted to the single link reliability value in the Cross-Entropy Crude Montecarlo (ρ = 0.40, 0.20 and 0.10 for r = 0.80, 0.90 and 0.99 respectively). The sample size to estimate the values of γk was 2000.

9 Conclusions

In this work, Five Monte Carlo methods to estimate network unreliability were studied and implemented: Crude Monte Carlo, Dagger, Permutation Monte Carlo, and Cross-Entropy over Crude and Permutation Monte Carlo. The methods were easy to adapt to compute the Diameter Constrained Reliability. To study their performance, the paper shows results over three elementary network topologies showing the accuracy level of all the Monte Carlo methods for every topology, and its relation to the single link reliability and to the parameter D.

Permutation Monte Carlo obtains very good results while Crude and Dagger performed poorly and both almost in the same order of accuracy. The Cross-Entropy models are modifications over Permutation Monte Carlo and Crude Monte Carlo. In both cases Cross-Entropy results in some numerical improvements, but actually the improvement attained is much more considerable in the case of Crude than in Permutation Monte Carlo.

Considering that Cross-Entropy models require a high effort, either in the programming and in the usage feature (selecting and adjusting parameters, sample sizes, etc.), Permutation Monte Carlo appears to be the best option out of all the implemented models (best relation “implementation simplicity–efficiency”).

The influence of D is more pronounced in Permutation than in Crude Monte Carlo, meaning that the most accurate method is more influenced by the value of D.

Since there is a vast literature on the problem of network reliability estimation in the case of classical functions of structure (k-terminal, for example) but (up to our knowledge) none in the Diameter Constrained Reliability approach, the results of this article can be a helpful contribution towards choosing the most

<table>
<thead>
<tr>
<th>D</th>
<th>TRI</th>
<th>GRID</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.83 x 10^-2 (5.65 x 10^-3)</td>
<td>1.93 x 10^-1 (3.65 x 10^-3)</td>
<td>4.70 x 10^-1 (5.58 x 10^-3)</td>
</tr>
<tr>
<td>3</td>
<td>4.31 x 10^-2 (1.74 x 10^-3)</td>
<td>1.10 x 10^-1 (2.78 x 10^-3)</td>
<td>2.93 x 10^-2 (1.44 x 10^-3)</td>
</tr>
<tr>
<td>4</td>
<td>3.08 x 10^-2 (1.47 x 10^-3)</td>
<td>6.58 x 10^-2 (2.15 x 10^-3)</td>
<td>8.36 x 10^-3 (7.70 x 10^-4)</td>
</tr>
<tr>
<td>5</td>
<td>3.02 x 10^-2 (1.46 x 10^-3)</td>
<td>5.28 x 10^-2 (1.93 x 10^-3)</td>
<td>6.56 x 10^-3 (6.82 x 10^-4)</td>
</tr>
</tbody>
</table>

Table 4: Network Unreliability Q (Standard Deviation) – Crude MC

<table>
<thead>
<tr>
<th>D</th>
<th>TRI</th>
<th>GRID</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.84 x 10^-1 (4.08 x 10^-3)</td>
<td>1.94 x 10^-1 (1.87 x 10^-3)</td>
<td>4.69 x 10^-1 (4.30 x 10^-3)</td>
</tr>
<tr>
<td>3</td>
<td>4.42 x 10^-2 (6.03 x 10^-4)</td>
<td>1.12 x 10^-1 (1.30 x 10^-3)</td>
<td>2.88 x 10^-2 (3.87 x 10^-4)</td>
</tr>
<tr>
<td>4</td>
<td>3.13 x 10^-2 (5.69 x 10^-4)</td>
<td>6.64 x 10^-2 (8.62 x 10^-4)</td>
<td>8.28 x 10^-3 (1.73 x 10^-4)</td>
</tr>
<tr>
<td>5</td>
<td>3.06 x 10^-2 (5.69 x 10^-4)</td>
<td>5.40 x 10^-2 (8.13 x 10^-4)</td>
<td>6.52 x 10^-3 (1.67 x 10^-4)</td>
</tr>
<tr>
<td>6</td>
<td>3.05 x 10^-2 (5.69 x 10^-4)</td>
<td>5.31 x 10^-2 (8.13 x 10^-4)</td>
<td>6.49 x 10^-3 (1.67 x 10^-4)</td>
</tr>
<tr>
<td>7</td>
<td>3.04 x 10^-2 (5.69 x 10^-4)</td>
<td>5.30 x 10^-2 (8.13 x 10^-4)</td>
<td>6.49 x 10^-3 (1.67 x 10^-4)</td>
</tr>
<tr>
<td>8</td>
<td>3.03 x 10^-2 (5.69 x 10^-4)</td>
<td>5.29 x 10^-2 (8.13 x 10^-4)</td>
<td>6.48 x 10^-3 (1.67 x 10^-4)</td>
</tr>
</tbody>
</table>

Table 5: Network Unreliability Q (Standard Deviation) – Permutation MC
appropriate methods in this last case.

Further work on these subjects include (i) to try the Diameter Constrained Reliability estimation on larger networks and for a subset of nodes $K \subset V$ rather than $K = V$, (ii) to implement other Monte Carlo alternatives from the literature, and (iii) to improve the implementation of the convergence mechanisms for the Cross-Entropy models.

References


[3] de Mello, T. H. Rare event estimation for static models via cross-entropy and importance sampling.


