Annals of Operations Research 134, 101–118, 2005


# The Cross-Entropy Method for Network Reliability Estimation

K.-P. HUI * Kin-Ping.HUI@dsto.defence.gov.au
*IN Division, Defence Science and Technology Organisation, Edinburgh 5111, Australia*

N. BEAN
*Department of Applied Mathematics, University of Adelaide, Adelaide 5005, Australia*

M. KRAETZL
*ISR Division, Defence Science and Technology Organisation, Edinburgh 5111, Australia*

DIRK P. KROESE
*Department of Mathematics, The University of Queensland, Brisbane 4072, Australia*

*Received March 2003; Revised September 2003; Accepted November 2003*

**Abstract.** Consider a network of unreliable links, modelling for example a communication network. Estimating the reliability of the network – expressed as the probability that certain nodes in the network are connected – is a computationally difficult task. In this paper we study how the Cross-Entropy method can be used to obtain more efficient network reliability estimation procedures. Three techniques of estimation are considered: Crude Monte Carlo and the more sophisticated Permutation Monte Carlo and Merge Process. We show that the Cross-Entropy method yields a speed-up over all three techniques.

**Keywords:** network reliability, cross-entropy, rare events, importance sampling, permutation Monte Carlo, merge process

It is well known that for large networks the exact calculation of network reliability is difficult. Indeed, computing the probability that a graph is connected is a #*P*-complete problem (Colbourn, 1987; Provan and Ball, 1982). Although approximation (Burtin and Pittel, 1972) and bounding (Barlow and Proschan, 1975; Barlow and Marshall, 1964; Esary, Proschan, and Walkup, 1967) methods are available, their accuracy and scope are very much dependent on the properties (such as size and topology) of the network. For large networks estimating the reliability using simulation techniques becomes desirable. In highly reliable networks such as modern communication networks, the probability of network failure is very small. Direct simulation of such rare events is slow and hence very expensive. Various techniques have been developed to produce better estimates. For example, Kumamoto proposed a very simple technique called *Dagger Sampling* to improve the Crude Monte Carlo simulation (Kumamoto et al., 1980). Fishman proposed

* Corresponding author.

*Procedure Q* which can provide reliability estimates as well as bounds (Fishman, 1986). Colbourn and Harms proposed a technique that provides progressive bounds that eventually converge to an exact reliability value (Colbourn and Harms, 1994). Easton and Wong proposed a sequential construction method (Easton and Wong, 1980). Elperin, Gertsbakh, and Lomonosov proposed *Evolution Models* for estimating reliability of highly reliable networks (Elperin, Gertsbakh, and Lomonosov, 1991, 1992). Hui et al. proposed a hybrid scheme that provides bounds and can provide speed-up by several orders of magnitude in a certain class of networks (Hui et al., 2003).

The Cross-Entropy (CE) method originated from Rubinstein (1997) as an adaptive algorithm for estimating probabilities of *rare events* in complex stochastic networks. The method can also be used for solving optimisation problems (Rubinstein, 1999, 2001). The Cross-Entropy method has been successfully applied to a wide range of combinatorial and continuous optimisation problems (Dubin, 2002; Lieber, 1998; Margolin, 2002; Rubinstein, 1999), including problems in reliability theory (Lieber, Rubinstein, and Elmakis, 1997), buffer allocation (Alon et al., 2005), telecommunication systems (de Boer, 2000; de Boer, Kroese, and Rubinstein, 2004; de Boer and Nicola, 2002; de Boer, Nicola, and Rubinstein, 2000), neural computation (Dubin, 2002), control and navigation (Helvik and Wittner, 2001; Wittner and Helvik, 2002), DNA sequence alignment (Keith and Kroese, 2002), scheduling (de Mello and Rubinstein, 2002; Margolin, 2002) and Max-Cut and bipartition problems (Rubinstein, 2002). A short review of the basic ideas behind the Cross-Entropy method is given at the end of this section, but for details we refer to the book on Cross-Entropy (Rubinstein and Kroese, 2004), and the tutorial in de Boer et al. (2005).

In this paper we investigate the benefits of the Cross-Entropy method to the estimation of network reliability or, equivalently, network *un*reliability. Basically, the Cross-Entropy method provides an iterative procedure to adaptively estimate the optimal Importance Sampling parameters for the quantity of interest, in our case the unreliability. We show that the Cross-Entropy technique indeed can lead to a significant speed-up.

The rest of the paper is organised as follows. At the end of this introduction we give a brief review of the most important aspects of the Cross-Entropy method for Monte Carlo simulation. In Section 2 we discuss unreliable networks and give three methods, Crude Monte Carlo (CMC), Permutation Monte Carlo (PMC) and the Merge Process (MP) to estimate network unreliability. Section 3 deals with the Cross-Entropy modification of the three methods. In this section we use a simple bridge system as an example. In Section 4 we give numerical results for larger examples, exemplifying what we found in Section 3 for the simple bridge system.

## 1. A very short introduction to CE

It is not our intention to give a detailed account of the Cross-Entropy method – for this we refer to de Boer et al. (2005) – but in order to keep this paper self-contained, we mention

the main points. Consider the problem of estimating

$$\ell = \mathbb{E}_{\boldsymbol{u}}[H(\boldsymbol{Y})] = \int H(\boldsymbol{y})\, f(\boldsymbol{y};\boldsymbol{u})\, d\boldsymbol{y}, \tag{1}$$

where $H(\boldsymbol{y})$ is some positive function and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ is a random vector with (joint) probability density function (pdf) $f(\cdot;\boldsymbol{u})$ which depends on a *reference parameter* $\boldsymbol{u}$. We can estimate $\ell$ using *Importance Sampling* (IS) as

$$\hat{\ell} = \frac{1}{N_1} \sum_{i=1}^{N_1} H\big(\boldsymbol{Y}_{(i)}\big)\, W\big(\boldsymbol{Y}_{(i)};\boldsymbol{u},\mathbf{v}\big), \tag{2}$$

where $\boldsymbol{Y}_{(1)}, \ldots, \boldsymbol{Y}_{(N_1)}$ is a random sample from $f(\cdot;\mathbf{v})$ – using a *different* reference parameter $\mathbf{v}$ – and

$$W(\boldsymbol{Y};\boldsymbol{u},\mathbf{v}) = \frac{f(\boldsymbol{Y};\boldsymbol{u})}{f(\boldsymbol{Y};\mathbf{v})}, \tag{3}$$

is the *likelihood ratio*. We can choose *any* reference vector $\mathbf{v}$ in (2), but the one that is optimal in the Cross-Entropy sense given $\boldsymbol{u}$ and $\boldsymbol{w}$ is

$$\mathbf{v}^* = \operatorname*{argmax}_{\mathbf{v}} \mathbb{E}_{\boldsymbol{w}}\, [H(\boldsymbol{Y})]\, W(\boldsymbol{Y};\boldsymbol{u},\boldsymbol{w}) \log f(\boldsymbol{Y};\mathbf{v}). \tag{4}$$

The corresponding pdf $f(\cdot;\mathbf{v}^*)$ has the smallest Cross-Entropy distance to the ideal (zero variance) Importance Sampling pdf (see for example Rubinstein and Melamed (1998))

$$g^*(\boldsymbol{y}) = \frac{H(\boldsymbol{y}) f(\boldsymbol{y};\boldsymbol{u})}{\ell}. \tag{5}$$

We can estimate the optimal Cross-Entropy reference vector, as the solution of the iterative procedure

$$\mathbf{v}_t = \operatorname*{argmax}_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^{N} H\big(\boldsymbol{Y}_{(i)}\big)\, W\big(\boldsymbol{Y}_{(i)};\boldsymbol{u},\mathbf{v}_{t-1}\big)\, \log f(\boldsymbol{Y}_{(i)};\mathbf{v}), \tag{6}$$

where at each iteration $t$ a random sample from $f(\cdot;\mathbf{v}_{t-1})$ is taken. The solution of (6) can often be determined *analytically*.

In the *rare-event* setting $H(\boldsymbol{Y})$ is of the form $H(\boldsymbol{Y}) = I\{S(\boldsymbol{Y}) \geq \gamma\}$ where $I$ is the indicator function and $\gamma$ is a constant, then

$$\ell = \mathbb{P}[S(\boldsymbol{Y}) \geq \gamma] \tag{7}$$

is a small probability. The function $S$ is called the *performance function*. For rare-event estimation problems, (6) is difficult to carry out because of the rareness of the event and most of the indicators $H(\boldsymbol{Y}_{(i)})$ will be zero. For such problems a two-phase Cross-Entropy procedure is employed, in which apart from $\mathbf{v}$ the *level* parameter $\gamma$ is also

updated, creating a sequence of pairs $\{(v_t, \gamma_t)\}$ with the goal of estimating the optimal Cross-Entropy reference parameter $\mathbf{v}^*$. Starting with $\mathbf{v}_0 = \boldsymbol{u}$ (the original or nominal parameter), the updating formulas are as follows:

Given a random sample $\boldsymbol{Y}_{(1)}, \ldots, \boldsymbol{Y}_{(N)}$ from $f(\cdot; \mathbf{v}_{t-1})$, we concentrate on the best performing $\rho$-portion of the samples. Let $\gamma_t$ be the sample $(1-\rho)$-quantile of the performances $S(\boldsymbol{Y}_{(i)})$, $i = 1, \ldots, N$, provided the sample quantile is less than $\gamma$; otherwise we set $\gamma_t$ to $\gamma$. In other words, let

$$\gamma_t = \min\left\{\gamma,\ S_{(\lceil(1-\rho)N\rceil)}\right\}, \tag{8}$$

where $S_{(j)}$ is the $j$-th *order-statistic* of the performances. Using the *same sample*, we let

$$\mathbf{v}_t = \underset{\mathbf{v}}{\operatorname{argmax}}\ \frac{1}{N}\sum_{i=1}^{N} I\left\{S\left(\boldsymbol{Y}_{(i)}\right) \geq \gamma_t\right\} W\left(\boldsymbol{Y}_{(i)}; \boldsymbol{u}, \mathbf{v}_{t-1}\right) \log f\left(\boldsymbol{Y}_{(i)}; \mathbf{v}\right). \tag{9}$$

Again, it is important to understand that in many cases an explicit formula for $\mathbf{v}_t$ can be given, that is, we do not need to "solve" the optimisation problem (9). Provided $\rho$ is small and $N$ is large enough, $\mathbf{v}_t$ in (9) converges to the optimal $\mathbf{v}^*$ in (4) (see Rubinstein and Kroese (2004)).

## 2. Estimation of network reliability

### 2.1. Network reliability

Consider an undirected graph (or network) $\mathcal{G}(V, E, K)$, where $V$ is the set of $n$ vertices (or nodes), $E$ is the set of $m$ edges, and $K \subseteq V$ is a set of *terminal* nodes, with $|K| \geq 2$. Associated with each edge $e \in E$ is a binary random variable $X_e$, denoting the *failure state* of the edge. In particular, $\{X_e = 1\}$ is the event that the edge $e$ is operational, and $\{X_e = 0\}$ is the event that it has failed. We label the edges from 1 to $m$, and call the vector $\boldsymbol{X} = (X_1, \ldots, X_m)$ the state of the network. Let $\mathcal{S}$ be the set of all $2^m$ possible states of $E$.

Next, we assume that the random variables $\{X_e,\ e \in E\}$ are mutually independent. Let $p_e$ and $q_e$ denote the reliability and unreliability of $e \in E$ respectively. That is

$$\begin{aligned} p_e &= \mathbb{P}[X_e = 1], \\ q_e &= \mathbb{P}[X_e = 0] = 1 - p_e. \end{aligned}$$

The reliability $r$ of the network is defined as the probability of $K$ being *connected* by operational edges. Thus,

$$r = \mathbb{E}[\varphi(\boldsymbol{X})] = \sum_{\boldsymbol{x}\in\mathcal{S}} \varphi(\boldsymbol{x})\, \mathbb{P}[\boldsymbol{X} = \boldsymbol{x}], \tag{10}$$

where

$$\varphi(\boldsymbol{x}) = \begin{cases} 1 & \text{if } K \text{ is connected,} \\ 0 & \text{otherwise.} \end{cases}$$

This is the standard formulation of the reliability of unreliable systems, see for example (Barlow and Proschan, 1975). The function $\varphi$ is called the *structure function* of the unreliable system. Note that the reliability of the network is completely determined by the individual edge reliabilities since we do not consider node failures.

For highly reliable networks it is sometimes more useful to analyse or estimate the system unreliability

$$\overline{r} = 1 - r.$$

Let $Q$ be an unbiased estimate of $\overline{r}$ obtained through Monte Carlo simulations, an important measure of the "efficiency" of the simulation is its *relative error*

$$\mathrm{re}(Q) = \sqrt{\frac{\mathrm{Var}(Q)}{(\mathbb{E}[Q])^2}}.$$

**Example 1** (Bridge Network)**.** Consider the simple network in figure 1, called a *bridge network*. The bridge network will serve as a convenient reference example to which we will return throughout the paper. Here we have five unreliable edges, labelled $1, \ldots, 5$. The network is operating if the two terminal nodes $A$ and $B$ are connected by operational edges.

It is not difficult to see that the structure function $\varphi$ is given by

$$\varphi(\boldsymbol{x}) = 1 - (1 - x_1x_3x_5)(1 - x_2x_3x_4)(1 - x_1x_4)(1 - x_2x_5).$$

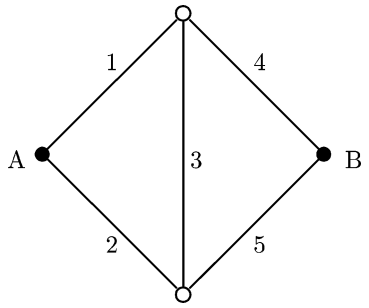


Figure 1. Two-terminal bridge network.

Applying the inclusion-exclusion principle to the mincuts, the system unreliability is equal to

$$\begin{aligned}\overline{r} = {} & q_1q_2 + q_2q_3q_4 - q_1q_2q_3q_4 + q_1q_3q_5 - q_1q_2q_3q_5 \\ & + q_4q_5 - q_1q_2q_4q_5 - q_1q_3q_4q_5 - q_2q_3q_4q_5 + 2q_1q_2q_3q_4q_5. \end{aligned} \tag{11}$$

### 2.2. Crude Monte Carlo simulation

Let us assume the typical situation where the edges are highly reliable, that is, the $q_e$ are close to 0. In that case, the appropriate quantity to estimate is the system unreliability $\overline{r}$ which will be close to 0, rather than $r$ which will be close to 1. The simplest way to estimate $\overline{r}$ is to use Crude Monte Carlo (CMC) simulation. Let $\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)}$ be independent identically-distributed random vectors with the same distribution as $\boldsymbol{X}$. Then

$$Q = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \varphi\left(\boldsymbol{X}_{(i)}\right)\right) \tag{12}$$

is an unbiased estimator for $\overline{r}$ and its *relative error* is

$$\mathrm{re}(Q) = \sqrt{\frac{\mathrm{Var}(Q)}{(\mathbb{E}[Q])^2}} = \sqrt{\frac{\overline{r}(1-\overline{r})/N}{\overline{r}^2}} = \sqrt{\frac{1-\overline{r}}{N\,\overline{r}}}\,.$$

This shows that for small $\overline{r}$, a large sample size is needed to estimate $\overline{r}$ accurately, since the event that the terminal nodes are not connected is a rare event.

### 2.3. Permutation Monte Carlo simulation

A more efficient way of estimating the network unreliability is *Permutation Monte Carlo* (Elperin, Gertsbakh, and Lomonosov, 1991). The idea is as follows. Consider the network $\mathcal{G}(V, E)$ in which each edge $e$ has an exponential repair time with repair rate $\lambda(e) = -\log(q_e)$. At time $t = 0$ all edges are failed and assume that all repair times are independent of each other. The state of $e$ at time $t$ is denoted by $X_e(t)$ and the state of the edge set $E$ at time $t$ is given by the vector $\boldsymbol{X}(t)$, defined in a similar way as before. Thus $(\boldsymbol{X}(t))$ is a Markov process with state space $\{0, 1\}^m$. This process is called the *Construction Process* (CP) of the network.

Let $\Pi$ denote the *order* in which the edges are constructed (become operational), and let $A_0, A_0 + A_1, \ldots, A_0 + \cdots + A_{m-1}$ be the times at which those edges are constructed. Hence the $A_i$ are *sojourn times* of $(\boldsymbol{X}(t))$. $\Pi$ is a random variable which takes values in the space of permutations of $E$.

For any permutation $\pi = (e_1, e_2, \ldots, e_m)$ define

$$\begin{aligned} E_0 &= E, \\ E_i &= E_{i-1} \backslash \{e_i\}, \quad 1 \le i \le m-1, \\ \lambda(E_i) &= \sum_{e \in E_i} \lambda(e), \end{aligned}$$

and let

$$b(\pi) = \min_i \{\varphi(E_i) = 1\}$$

be the so-called *critical number* of $\pi$. From the general theory of Markov processes it is not difficult to see that

$$\mathbb{P}[\Pi = \pi] = \prod_{j=1}^{m} \frac{\lambda(e_j)}{\lambda(E_{j-1})}. \tag{13}$$

Moreover, conditional on $\{\Pi = \pi\}$, the sojourn times $A_0, \ldots, A_{m-1}$ are independent and each $A_i$ is exponentially distributed with parameter $\lambda(E_i)$, $i = 0, \ldots, m-1$.

Note that the probability of each edge $e$ being operational at time $t = 1$ is $p_e$. It follows that the *network* reliability at time $t = 1$ is the same as in (10). Hence, by conditioning on $\Pi$ we have

$$r = \mathbb{E}[\varphi(\boldsymbol{X}(1))] = \sum_{\pi} \mathbb{P}[\Pi = \pi]\, \mathbb{P}[\varphi(\boldsymbol{X}(1)) = 1 \mid \Pi = \pi], \tag{14}$$

and

$$\overline{r} = 1 - r = \sum_{\pi} \mathbb{P}[\Pi = \pi]\, \mathbb{P}[\varphi(\boldsymbol{X}(1)) = 0 \mid \Pi = \pi]. \tag{15}$$

Using the definitions of $A_i$ and $b(\pi)$, we can write the last probability in terms of convolutions of exponential distribution functions. Namely, for any $t \ge 0$ we have

$$\begin{aligned} \mathbb{P}[\varphi(\boldsymbol{X}(t)) = 0 \mid \Pi = \pi] &= \mathbb{P}\big[A_0 + \cdots + A_{b(\pi)-1} > t \mid \Pi = \pi\big] \\ &= 1 - \underset{0 \le i < b(\pi)}{\mathbf{Conv}} \{1 - \exp[-\lambda(E_i)\, t]\}. \end{aligned} \tag{16}$$

Let

$$G(\pi) = \mathbb{P}[\varphi(\boldsymbol{X}(1)) = 0 \mid \Pi = \pi], \tag{17}$$

as given in (16). Equation (15) can then be rewritten as

$$\overline{r} = \mathbb{E}[G(\Pi)], \tag{18}$$

and this shows how the Permutation Monte Carlo simulation scheme works. Namely, let $\Pi_{(1)}, \ldots, \Pi_{(N)}$ be independent-identically distributed random permutations, each distributed according to $\Pi$. Then

$$\widehat{\overline{r}} = \frac{1}{N} \sum_{i=1}^{N} G\big(\Pi_{(i)}\big) \tag{19}$$

is an unbiased estimator for $\overline{r}$.

### 2.4. *Merge process simulation*

A closer look at the evolution of the Construction Process reveals that many of the above results remain valid when we *merge* various states into "super states" at various stages of the process. This is known as the *Merge Process* (MP). We briefly describe the main ideas below (see Lomonosov, 1994 for a detailed description).

For a subset of edges $F \subseteq E$, the *proper partition* (in connected components) of the subgraph $\mathcal{G}(V, F)$ (including isolated nodes, if any) is denote by $\sigma = \{V_1, V_2, \ldots, V_k\}$. Let $I_i$ denote the edge-set of the induced subgraph $\mathcal{G}(V_i)$. The set $I_\sigma = I_1 \cup \ldots \cup I_k$ of *inner* edges, that is, the edges within the components, is the closure of $F$ (denoted by $\langle F \rangle$). Denote its complement (the *inter-component* edges) by $E_\sigma = E \backslash I_\sigma$.

Let $\mathbb{L}(\mathcal{G})$ be the collection of all proper partitions of $\mathcal{G}(V, E)$. It is not difficult to see that $\mathbb{L}(\mathcal{G})$ is a *lattice*, ordered by the relation $\tau \prec \sigma \iff I_\tau \subset I_\sigma$ (that is, $\sigma$ is obtained by merging components of $\tau$).

Consider the Construction Process $(\boldsymbol{X}(t))$ of the network. By restricting the process $(\boldsymbol{X}(t))$ to $\mathbb{L}(\mathcal{G})$ we obtain another Markov process $(\mathbb{X}(t))$, called the *Merge Process* (MP) of the network. This process starts from the initial state $\sigma_0$ of isolated nodes and ends at the terminal state $\sigma_\omega$ corresponding to $\mathcal{G}(V, E)$.

For each $\sigma \in \mathbb{L}(\mathcal{G})$ the sojourn time in $\sigma$ has an exponential distribution with parameter $\lambda(\sigma) = \sum_{e \in E_\sigma} \lambda(e)$, independent of everything else. Moreover, the transition probability from $\tau$ to $\sigma$ (where $\sigma$ is a direct successor of $\tau$) is given by:

$$\frac{\lambda(\tau) - \lambda(\sigma)}{\lambda(\tau)}.$$

Next, in analogy with the results for the Construction Process, we define a *trajectory* of $(\mathbb{X}(t))$ as a sequence $\theta = (\sigma_0, \sigma_1, \ldots, \sigma_b)$, where $b = b(\theta)$ is the first index $i$ such that $\sigma_i$ is "up", that is, the network is operational. By defining $\varphi(\mathbb{X}(t)) = \varphi(\boldsymbol{X}(t))$, we thus have

$$\overline{r} = \mathbb{P}[\varphi(\mathbb{X}(1)) = 0] = \mathbb{E}[G(\Theta)], \tag{20}$$

where $\Theta$ is a random trajectory of $(\mathbb{X}(t))$. For each outcome $\theta = (\sigma_0, \ldots, \sigma_b)$ of $\Theta$, $G(\theta)$ is given by

$$G(\theta) = \mathbb{P}[\varphi(\mathbb{X}(1)) = 0 \mid \Theta = \theta] = \mathbb{P}\big[A_0 + \cdots + A_{b(\theta)-1} > 1 \mid \Theta = \theta\big], \tag{21}$$

where $A_i$ is the sojourn time at $\sigma_i$. Therefore, $G(\theta)$ is given by

$$1 - \underset{0 \le i < b(\theta)}{\mathbf{Conv}} \{1 - \exp[-\lambda(\sigma_i)\, t]\} \tag{22}$$

evaluated at $t = 1$.

## 3. Estimating network reliability using the cross-entropy method

### *3.1. Cross-entropy and crude Monte Carlo*

If we use the Construction Process idea of Section 2.3 in the Crude Monte Carlo framework of Section 2.2, the Cross-Entropy method fits naturally. Instead of sampling the up/down state of individual edges, we sample the up time of each edge. Then we check if the network is functioning at time $t = 1$ and this probability is the network reliability.

In other words, translate the original problem (estimating $\overline{r}$), which involves independent Bernoulli random variables $X_1, \ldots, X_m$, into an estimation problem involving independent exponential random variables $Y_1, \ldots, Y_m$. Specifically, imagine that we have a time-dependent system in which at time 0 all edges have failed and are under repair, and let $Y_1, \ldots, Y_m$, with $Y_i \sim \mathsf{Exp}(u_i^{-1})$ and $u_i = 1/\lambda(i) = -1/\log q_i$ be the independent *repair times* of the edges. Note that, by definition

$$\mathbb{P}[Y_i \ge 1] = \mathrm{e}^{-1/u_i} = q_i \quad i = 1, \ldots, m.$$

Now, for each $\boldsymbol{Y} = (Y_1, \ldots, Y_m)$ let $S(\boldsymbol{Y})$ be the (random) time at which the system "comes up" (the terminal nodes become connected). Then, we can write

$$\overline{r} = \mathbb{P}[S(\boldsymbol{Y}) \ge 1].$$

Hence, we have written the estimation of $\overline{r}$ in the standard rare event formulation of (7) and thus can directly apply the Cross-Entropy method from de Boer et al. (2005).

Instead of sampling independently for each $i$ from $\mathsf{Exp}(u_i^{-1})$, we sample from $\mathsf{Exp}(v_i^{-1})$. The vector $\mathbf{v} = (v_1, \ldots, v_m)$ is thus our reference parameter. We now construct a sequence of pairs $\{(\mathbf{v}_t, \gamma_t)\}$ such that $\mathbf{v}_t$ converges to a reference vector close to the optimal Cross-Entropy reference parameter and $\gamma_t$ eventually reaches one. Starting with $\mathbf{v}_0 = \boldsymbol{u} = (u_1, \ldots, u_m)$, at each iteration $t$ we draw a random sample $\boldsymbol{Y}_{(1)}, \ldots, \boldsymbol{Y}_{(N)}$ from the pdf $f(\cdot; \mathbf{v}_{t-1})$ of $\boldsymbol{Y}$ and update the level parameter using (8) and the reference

parameter using (9), which in this case has the analytical solution

$$v_{t,j} = \frac{\sum_{i=1}^{N} I\{S(\boldsymbol{Y}_{(i)}) \geq \gamma_t\} W\left(\boldsymbol{Y}_{(i)}; \boldsymbol{u}, \mathbf{v}_{t-1}\right) Y_{(i)j}}{\sum_{i=1}^{N} I\{S(\boldsymbol{Y}_{(i)}) \geq \gamma_t\} W\left(\boldsymbol{Y}_{(i)}; \boldsymbol{u}, \mathbf{v}_{t-1}\right)}, \tag{23}$$

where $W$ is the likelihood ratio

$$W(\boldsymbol{y}; \boldsymbol{u}, \mathbf{v}) = \frac{f(\boldsymbol{y}; \boldsymbol{u})}{f(\boldsymbol{y}; \mathbf{v})} = \exp\left(-\sum_{j=1}^{m} y_j \left(\frac{1}{u_j} - \frac{1}{v_j}\right)\right) \prod_{j=1}^{m} \frac{v_j}{u_j}. \tag{24}$$

For instance, after iteration $T$ when $\gamma_T$ reaches one, we estimate $\bar{r}$ using Importance Sampling as

$$\hat{\bar{r}} = \frac{1}{N_1} \sum_{i=1}^{N_1} I\{S(\boldsymbol{Y}_{(i)}) \geq 1\} W\left(\boldsymbol{Y}_{(i)}; \boldsymbol{u}, \mathbf{v}_T\right).$$

**Example 2** (Bridge Network, CMC with CE)**.** Consider now the bridge network of Example 1. Since we have a 2-terminal network, the function $S$ can be described in terms of the maximal paths of the networks. In particular, we have

$$S(\boldsymbol{Y}) = \min_i \max_{e \in \mathcal{P}_i} X_e.$$

where $\mathcal{P}_1 = \{1, 4\}$, $\mathcal{P}_2 = \{2, 5\}$, $\mathcal{P}_3 = \{2, 3, 4\}$ and $\mathcal{P}_4 = \{1, 3, 5\}$ are the sets of maximal paths, see (Barlow and Proschan, 1975; Lieber, Rubinstein, and Elmakis, 1997). Suppose the "nominal" parameter vector is $\boldsymbol{u} = (0.3, 0.1, 0.8, 0.1, 0.2)$. From (11) the exact unreliability is $\bar{r} = 7.07868\text{e}-05$. A typical result of the simulations is given in Table 1. The Cross-Entropy parameters used were: (initial) sample size $N = 2000$ and rarity parameter $\rho = 0.01$ in (8). In both CMC and CE-CMC a final sample size of $10^6$ was used.

By using the Cross-Entropy method we have achieved, with minimal effort, a 98% reduction in variance (equivalent to a 50-fold speed-up) compared to the Crude Monte Carlo method. The algorithm stopped after two iterations, as illustrated in Table 2. Notice that the algorithm tilted the parameters of the *mincut* elements {1, 3, 5} to higher values while lowering the rest.

Table 1
Results for CMC and CE-CMC.

| | $\hat{\bar{r}}$ | r̂e | $v\hat{a}r$ |
|---|---|---|---|
| CE-CMC | 6.99111e−05 | 0.0166777 | 1.35945e−12 |
| CMC | 6.1e−05 | 0.128033 | 6.09963e−11 |
| True value | 7.07868e−05 | | |

Table 2
Convergence of the parameters.

| $t$ | $\hat{\gamma}_t$ | $\hat{\mathbf{v}}_t$ | | | | |
|---|---|---|---|---|---|---|
| 0 | – | 0.3 | 0.1 | 0.8 | 0.1 | 0.2 |
| 1 | 0.507 | 0.964833 | 0.216927 | 1.20908 | 0.0892952 | 0.567551 |
| 2 | 1.000 | 1.19792 | 0.120166 | 1.57409 | 0.0630103 | 1.15137 |

### 3.2. *Cross-Entropy and permutation Monte Carlo*

We now want to apply the Cross-Entropy method to the Permutation Monte Carlo simulation of Section 2.3. Instead of estimating $\bar{r}$ using (19), we estimate it using Importance Sampling, where we apply a change of measure – determined by the Cross-Entropy method – to the distribution of the random permutation $\Pi$. There are many ways to define a distribution on the space of permutations, see also Remark 2.1 below. However, note that the original distribution of $\Pi$ is determined by the exponential distribution of $\boldsymbol{Y}$. In fact, $\Pi$ can be viewed as a function of $\boldsymbol{Y}$. To see this, generate $Y_1, \ldots, Y_m$ independently according to $Y_i \sim \mathsf{Exp}(u_i^{-1})$ and order the $Y_i's$ such that $Y_{\Pi_1} \leq Y_{\Pi_2} \leq \cdots \leq Y_{\Pi_m}$. Then take $\Pi(\boldsymbol{Y}) = (\Pi_1, \ldots, \Pi_m)$ as our random permutation. We write (18) as

$$\bar{r} = \mathbb{E}_{\boldsymbol{u}}[G(\Pi(\boldsymbol{Y}))] = \mathbb{E}_{\boldsymbol{u}}[S(\boldsymbol{Y})], \tag{25}$$

where we *redefine* $S(\boldsymbol{Y})$ as $G(\Pi(\boldsymbol{Y}))$, with $G$ as in (17). A natural way of defining a change of measure is to choose different parameters $v_i$ (instead of the nominal $u_i$) for the exponential distributions of the edge lifetimes, in a similar way to Section 3.1. Thus $\mathbf{v} = (v_1, \ldots, v_m)$ is still the vector of mean "repair" times. However, we have a slightly different situation from Section 3.1, because instead of having to estimate a rare event probability $\mathbb{P}[S(\boldsymbol{Y}) \geq 1]$ we now have to estimate the (small) expectation $\mathbb{E}[S(\boldsymbol{Y})]$. We can no longer use a *two-phase* procedure (updating $\gamma$ and $\mathbf{v}$) but instead use a one-phase procedure (6) in which we only update $\mathbf{v}_t$. The analytic solution to (6) for the $i$-th component of $\mathbf{v}_t$ is

$$v_{t,i} = \frac{\sum_{k=1}^{N} S\big(\boldsymbol{Y}_{(k)}\big)\, W\big(\boldsymbol{Y}_{(k)}; \boldsymbol{u}, \mathbf{v}_{t-1}\big)\, Y_{(k)i}}{\sum_{k=1}^{N} S(\boldsymbol{Y}_{(k)})\, W\big(\boldsymbol{Y}_{(k)}; \boldsymbol{u}, \mathbf{v}_{t-1}\big)}, \tag{26}$$

where $Y_{(k)i}$ is the $i$-th component of $\boldsymbol{Y}_{(k)}$. To improve convergence in random sampling situations, it is often beneficial to use a smoothing parameter $\alpha$ to blend the old with the new estimates. That is we take

$$\mathbf{v}_t' = \alpha \mathbf{v}_t + (1 - \alpha)\mathbf{v}_{t-1}$$

as the new parameter vector for the next iteration.

### 3.3. Cross-entropy and the Merge Process

For each permutation $\pi$ in the Creation Process (PMC), there is a corresponding trajectory $\theta$ in the Merge Process. Let $\Theta : \pi \mapsto \theta$ be the mapping that assigns to each permutation $\pi$ the corresponding unique trajectory $\theta$. Then (25) can be rewritten as

$$\overline{r} = \mathbb{E}_{\boldsymbol{u}}[G(\Theta(\Pi(\boldsymbol{Y})))] = \mathbb{E}_{\boldsymbol{u}}[S(\boldsymbol{Y})], \tag{27}$$

where $S(\boldsymbol{Y})$ has been redefined as $G(\Theta(\Pi(\boldsymbol{Y})))$, with $G$ as in (21). Then the same Cross-Entropy procedure (26) described in Section 3.2 can be applied to the Merge Process as well.

**Example 3** (Bridge Network, MP and PMC with CE)**.** We return to the bridge network of Example 2. Table 3 lists the results for the standard MP and PMC simulations, compared with their counterparts with Importance Sampling in which the reference parameters are determined by the Cross-Entropy method. The nominal reference parameter remains unchanged $\boldsymbol{u} = (0.3, 0.1, 0.8, 0.1, 0.2)$, and we use the Cross-Entropy parameters $\alpha = 0.7$ and $N = 2000$. The final sample sizes are $N_1 = 10^6$ in all the original and Cross-Entropy simulations.

We have repeated this experiment various times and have consistently found that the Merge Process and the Permutation Monte Carlo have very close performance in such a small example network. We also found that the Cross-Entropy technique provides an improvement (reduction) in variance of roughly 20% in both cases.

Note that the Cross-Entropy with Crude Monte Carlo simulation still has over 100 times the variance of that in MP, CE-MP, PMC or CE-PMC simulations. This shows that no matter how much one modifies the Crude Monte Carlo with smart sampling techniques, it still cannot compare to the simple Permutation Monte Carlo sampling. It is well known that *conditional Monte Carlo* methods such as the MP and PMC always yield a variance reduction over CMC, see for example Section 5.4 of Rubinstein and Melamed (1998). However, Table 3 also suggests that even though the Cross-Entropy method does not provide the same percentage improvement as in the Crude Monte Carlo

Table 3
Results for CE-PMC and CE-MP.

| | $\hat{\overline{r}}$ | r̂e | $v\hat{a}r$ |
|---|---|---|---|
| CE-MP | 7.08223e−05 | 0.00116362 | 6.79140e−15 |
| MP | 7.08109e−05 | 0.00132036 | 8.74143e−15 |
| CE-PMC | 7.07905e−05 | 0.00120961 | 7.33236e−15 |
| PMC | 7.07850e−05 | 0.00132445 | 8.78930e−15 |
| CE-CMC | 6.99111e−05 | 0.01667770 | 1.35945e−12 |
| True value | 7.07868e-05 | | |

Table 4
Evolution of the reference parameters.

| $t$ | $\mathbf{v}_t$ | | | | |
|---|---|---|---|---|---|
| 0 | 0.3 | 0.1 | 0.8 | 0.1 | 0.2 |
| 1 | 0.357676818 | 0.073782224 | 0.863431103 | 0.068991687 | 0.254756949 |
| 2 | 0.377520896 | 0.065074510 | 0.863118100 | 0.059496186 | 0.271816687 |
| 3 | 0.386880121 | 0.059557488 | 0.852180730 | 0.057643532 | 0.278509639 |

scheme, it is still worthwhile applying the technique to the "better" Merge Process or Permutation Monte Carlo sampling schemes.

With the CE-MP or CE-PMC sampling, there is no parameter $\gamma$ to indicate when to stop the Cross-Entropy parameter tuning, therefore we need to use other strategies. Since we have imprecise knowledge of the performance function, we have to resort to simulation to evaluate that function at each point in order to optimize the function (4). On the other hand, we do not want to spend too long on the CE parameter estimation effort, compared to the real simulation. As a result, we cannot use classic convergence criteria such as: stop when two consecutive vectors are $\epsilon$ close in some norm. Fortunately, however, permutation (and trajectory) sampling depends on the relative weight of each edge and hence it is fairly insensitive to the precise values of the Importance Sampling parameter $\mathbf{v}_t$. Therefore we only require a vector that is in the "right" region.

Table 4 displays the evolution of the reference parameters for the CE-MP, where we stopped the Cross-Entropy algorithm after only three iterations, when the estimates "stabilised" (the values stop fluctuating). Notice again that the algorithm allocated more attention to the *mincut* elements $\{1, 3, 5\}$ and treated the rest as unimportant.

*Remark 2.1* (Zero-variance IS Permutation/Trajectory Distribution). For a general network, it is not difficult to find the *ideal* Importance Sampling distribution on the space of all permutations. It is given by the pmf

$$g^*(\pi) = \frac{G(\Pi) f(\pi)}{\bar{r}},$$

where $f(\pi)$ is the probability of the permutation $\pi = (e_1, \ldots, e_m)$ occurring under the original measures. For the Merge Process, $g^*(\theta)$ is constructed in the same way. Under $g^*$ the Importance Sampling estimators have zero variance. Although this has little practical value for large systems (for which we do not know $\bar{r}$), it can be of help to construct and test good Cross-Entropy sampling strategies.

## 4. Numerical experiments

In this section we give a few larger examples that might be found in communication networks. Figure 2 shows a $3 \times 3$ and a $6 \times 6$ grid network, each network has four

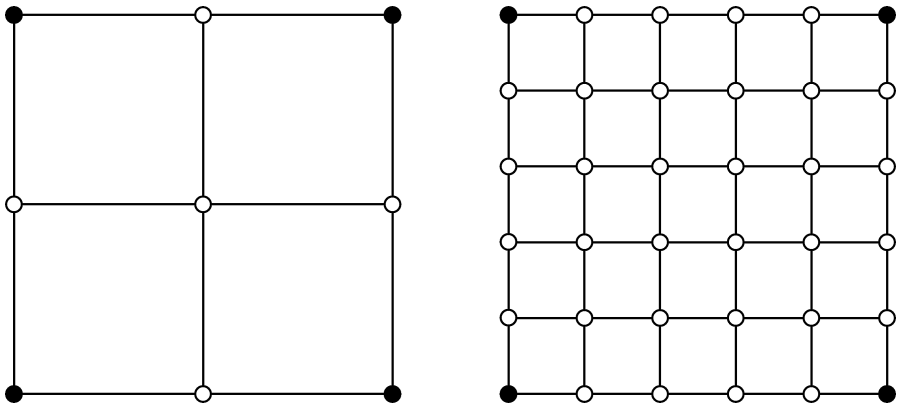

Figure 2. A $3 \times 3$ and a $6 \times 6$ grid network.

terminals at the corners. All links have the same failure probability. All experiments use a final sample size of $10^6$ and the Cross-Entropy tuning batch sample size of 5000. In the tables, $T$ denotes the CE tuning iteration and $\alpha$ denotes the smoothing parameter. The CE-CMC had a rarity parameter $\rho = 0.02$.

For verification purposes, the exact network failure probabilities are evaluated and listed as well. Note that in these trivial examples, alternative approaches such as the Burtin-Pittel approximation (Burtin and Pittel, 1972; Gertsbakh, 2000) can also be used to obtain fairly accurate results.

**Example 4** ($3 \times 3$ unreliable grid)**.** In this example, all links of the $3 \times 3$ grid network have the same failure probability $q = 10^{-3}$. A typical result of the simulation is given in Table 5.

The CMC method gives a poor variance and relative error as expected. The CE-CMC shows a 95% reduction in variance but is still too high to make it very useful. The PMC method gives a much smaller variance (0.1% of CE-CMC) and the Cross-Entropy method achieved further reduction of 20–25% on average. The MP method has an even

Table 5
Simulation results for the $3 \times 3$ unreliable grid network.

| | T | $\alpha$ | $\hat{\bar{r}}$ | rê | $v\hat{a}r$ |
|---|---|---|---|---|---|
| CE-MP | 10 | 0.1 | 4.01172e – 06 | 0.001071 | 1.84515e – 17 |
| MP | – | – | 4.01179e – 06 | 0.001122 | 2.02652e – 17 |
| CE-PMC | 10 | 0.1 | 4.01057e – 06 | 0.003420 | 1.88147e – 16 |
| PMC | – | – | 4.01609e – 06 | 0.003895 | 2.44653e – 16 |
| CE-CMC | 4 | 1 | 2.8296e – 06 | 0.150517 | 1.81393e – 13 |
| CMC | – | – | 4e – 06 | 0.499999 | 3.99998e – 12 |
| True value | | | 4.01199e – 06 | | |

Table 6
Simulation results for the $3 \times 3$ reliable grid network.

| | T | $\alpha$ | $\hat{\bar{r}}$ | rê | $v\hat{a}r$ |
|---|---|---|---|---|---|
| CE-MP | 10 | 0.1 | 3.99876e – 12 | 0.001076 | 1.85116e – 29 |
| MP | – | – | 4.00279e – 12 | 0.001127 | 2.03639e – 29 |
| CE-PMC | 10 | 0.1 | 4.00797e – 12 | 0.003455 | 1.91801e – 28 |
| PMC | – | – | 3.99354e – 12 | 0.003940 | 2.47621e – 28 |
| CE-CMC | 5 | 1 | 1.75878e – 12 | 0.176095 | 9.59220e – 26 |
| CMC | – | – | 0 | undefined | undefined |
| True value | | | 4.00001e – 12 | | |

smaller variance (10% of PMC) and the Cross-Entropy method provides roughly a 20% further reduction, making the CE-MP the most efficient method to use.

**Example 5** ($3 \times 3$ reliable grid)**.** This example is the same as the last one except that the link failure probability is $q = 10^{-6}$. A typical result of the simulation is given in Table 6.

Clearly the failure probability in the order of $10^{-12}$ is well beyond the ability of the CMC method with only $10^6$ samples. The Cross-Entropy method helped to produce some meaningful results, however the relative error is still rather high. On the other hand the MP and PMC methods give accurate estimates and the Cross-Entropy method consistently reduces their variance by around 10% in MP, and 20% in PMC. With the variance in MP starting at less than 10% of that in PMC, it makes the CE-MP the best performing method.

**Example 6** ($6 \times 6$ unreliable grid)**.** This is a larger network example consisting of 36 nodes and 60 edges with equal link failure probability $q = 10^{-3}$. A typical result of the simulation is given in Table 7.

Again the CMC and CE-CMC methods cannot provide accurate estimates with $10^6$ samples while the PMC, CE-PMC, MP and CE-MP give good results. Notice that in this larger example, the Cross-Entropy method reduces the variance by about 65% in the

Table 7
Simulation results for the $6 \times 6$ unreliable grid network.

| | T | $\alpha$ | $\hat{\bar{r}}$ | rê | $v\hat{a}r$ |
|---|---|---|---|---|---|
| CE-MP | 10 | 0.1 | 4.00239e – 06 | 0.001528 | 3.74067e – 17 |
| MP | – | – | 4.01041e – 06 | 0.001745 | 4.89923e – 17 |
| CE-PMC | 10 | 0.1 | 4.02088e – 06 | 0.011778 | 2.24284e – 15 |
| PMC | – | – | 3.95377e – 06 | 0.020306 | 6.44547e – 15 |
| CE-CMC | 4 | 1 | 1.32219e – 06 | 0.489052 | 4.18118e – 13 |
| CMC | – | – | 6e – 06 | 0.408247 | 5.99996e – 12 |
| True value | | | 4.00800e – 06 | | |

Table 8
Simulation results for the $6 \times 6$ reliable grid network.

| | T | $\alpha$ | $\hat{\bar{r}}$ | r̂e | $v\hat{a}r$ |
|---|---|---|---|---|---|
| CE-MP | 10 | 0.1 | 3.99869e-12 | 0.001533 | 3.75850e-29 |
| MP | – | – | 3.99755e-12 | 0.001750 | 4.89134e-29 |
| CE-PMC | 10 | 0.1 | 4.00455e-12 | 0.012755 | 2.60903e-27 |
| PMC | – | – | 4.00552e-12 | 0.020997 | 7.07316e-27 |
| CE-CMC | 5 | 1 | 8.62462e-14 | 0.907805 | 6.13007e-27 |
| CMC | – | – | 0 | undefined | undefined |
| True value | | | 4.00001e-12 | | |

PMC. Compared to the previous examples, this network has a much larger population size and variance. For instance, the $3 \times 3$ network has about $4.8 \times 10^8$ edge permutations while the $6 \times 6$ network has about $8.3 \times 10^{81}$. The MP has a reduced trajectory population size, and has a much smaller sample variance (less than 1% of PMC). The Cross-Entropy method still provides 25% reduction in variance in the CE-MP, making it the smallest of all. Here, the Cross-Entropy method demonstrates its ability to find excellent Importance Sampling parameter vectors to reduce the sample variance.

**Example 7** ($6 \times 6$ reliable grid)**.** This example is the same as the last one except that the link failure probability is $q = 10^{-6}$. A typical result of the simulation is given in Table 8.

It has very similar findings to the last example: The CMC and CE-CMC methods cannot handle such a low probability with $10^6$ samples. The PMC provides good estimates and yet the Cross-Entropy method reduces its sample variances further by about 65% in the CE-PMC. The MP starts with a much lower (1%) sample variance than that of PMC and the Cross-Entropy further reduces it by 25% in the CE-MP.

### *4.1. Summary of results*

With a better "sampling structure" and smart conditioning, the Merge Process and the Permutation Monte Carlo schemes are superior to the Crude Monte Carlo scheme. The Cross-Entropy technique further improves the performance of the MP and the PMC schemes, the degree of improvement becomes more prominent as the network size grows. Close inspection of the Importance Sampling parameter $\mathbf{v}_T$ reveals that all the *bottleneck-cut* edges have been allocated a higher importance than the rest. Possible directions for further research are a qualitative – rather than quantitative – study of the difference between the methods and a theoretical analysis of complexity issues. Both of these are out of the scope of this paper.

Another point to note is the smoothing parameter $\alpha$. If we keep $\alpha = 0.7$ as in the bridge example, the Importance Sampling parameters $\mathbf{v}$ might oscillate instead of converge to the optimal $\mathbf{v}^*$ and as a consequence give poor estimates. We found that in

larger networks, a smaller smoothing parameter such as $\alpha = 0.1$ is much more robust and always gave good results in our experiments. Numerical experience suggests that an increase in the tuning sample size $N$ can alleviate the need to reduce the smoothing parameter $\alpha$ in larger problems. Of course, this means more effort has to be spent estimating the optimal Importance Sampling parameter $\mathbf{v}^*$. However, if we leave $\alpha$ very small, more iterations are required for convergence towards $\mathbf{v}^*$. This raises the question of the most efficient way to allocate effort in estimating $\mathbf{v}^*$.

## 5. Conclusions

The Cross-Entropy technique gives substantial improvement over the Crude Monte Carlo estimation of network reliability. However, no matter how much we improve the Crude Monte Carlo sampling, it still cannot match the simple Permutation Monte Carlo or the Merge Process sampling schemes. Therefore it is still better to apply the Cross-Entropy technique to the latter schemes. In this paper, we showed how the Cross-Entropy technique can be applied to further improve the Merge Process and the Permutation Monte Carlo scheme. Furthermore, the examples suggested that the improvement can grow with the network size as the technique has the ability to quickly "home in" to important edges.

## References

Alon, G., D.P. Kroese, T. Raviv, and R.Y. Rubinstein. (2005). "Application of the Cross-Entropy Method to the Buffer Allocation Problem in a Simulation-Based Environment." *Annals of Operations Research* 134, 137–151.

Barlow, R. and A. Marshall. (1964). "Bounds for Distributions with Monotone Hazard Rate, I and II." *Ann. Maths. Statist.* 35, 1234–1274.

Barlow, R. and F. Proschan. (1975). *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart & Wilson.

Burtin, Y. and B. Pittel. (1972). "Asymptotic Estimates of the Reliability of a Complex System." *Engrg. Cybern.* 10(3), 445–451.

Colbourn, C. (1987). *The Combinatorics of Network Reliability*. Oxford University Press.

Colbourn, C. and D. Harms. (1994). "Evaluating performability: Most probable states and bounds." *Telecommunication Systems* 2, 275–300.

de Boer, P., D. Kroese, S. Manor, and R.Y. Rubinstein. (2005). "A Tutorial on the Cross-Entropy Method." *Annals of Operations Research* 134, 19–67. http://www.cemethod.org.

de Boer, P.T. (2000). "Analysis and Efficient Simulation of Queueing Models of Telecommunication Systems." Ph.D. thesis, University of Twente.

de Boer, P.T. and V.F. Nicola. (2002). "Adaptive State-Dependent Importance Sampling Simulation of Markovian Queueing Networks." *European Transactions on Telecommunications* 13(4), 303–315.

de Boer, P.T., V.F. Nicola, and R.Y. Rubinstein. (2000). "Adaptive Importance Sampling Simulation of Queueing Networks." In *Proceedings of the 2000 Winter Simulation Conference*, Orlando, Florida, pp. 646–655.

de Boer, P. T., D.P. Kroese, and R.Y. Rubinstein. (2004). "A Fast Cross-Entropy Method for Estimating Buffer Overflows in Queueing Networks." *Management Science* 50(7), 883–895.

de Mello, T.H. and R.Y. Rubinstein. (2002). "Rare Event Probability Estimation for Static Models Via Cross-Entropy and Importance Sampling." Submitted for publication.

Dubin, U. (2002). "The Cross-Entropy Method with Applications for Combinatorial Optimization." Master's thesis, The Technion, Israel Institute of Technology, Haifa, Israel.

Easton, M. and C. Wong. (1980). "Sequential Destruction Method for Monte Carlo Evaluation of System Reliability." *IEEE Transactions on Reliability* R-29, 27–32.

Elperin, T., I.B. Gertsbakh, and M. Lomonosov. (1991). "Estimation of Network Reliability Using Graph Evolution Models." *IEEE Transactions on Reliability* 40(5), 572–581.

Elperin, T., I.B. Gertsbakh, and M. Lomonosov. (1992). "An Evolution Model for Monte Carlo Estimation of Equilibrium Network Renewal Parameters." *Probability in the Engineering and Informational Sciences* 6, 457–469.

Esary, J.D., F. Proschan, and D.W. Walkup. (1967). "Association of Random Variables, with Applications." *Ann. math. Statist.* 38, 1466–1474.

Fishman, G. (1986). "A Monte Carlo Sampling Plan for Estimating Network Reliability." *Operations Research* 34(4), 581–594.

Gertsbakh, I.B. (2000). *Reliability Theory with Application to Preventive Maintenance*. Springer.

Helvik, B.E. and O. Wittner. (2001). "Using the Cross-Entropy Method to Guide/Govern Mobile Agent's Path Finding in Networks." In *3rd International Workshop on Mobile Agents for Telecommunication Applications - MATA'01*.

Hui, K.-P., N. Bean, M. Kraetzl, and D.P. Kroese. (2003). "The Tree Cut and Merge Algorithm for Estimation of Network Reliability." *Probability in the Engineering and Informational Sciences* 17(1), 24–45.

Keith, J. and D.P. Kroese. (2002). "Sequence Alignment By Rare Event Simulation." In *Proceedings of the 2002 Winter Simulation Conference*, San Diego, pp. 320–327.

Kumamoto, H., K. Tanaka, K. Inoue, and E.J. Henley, (1980). "Dagger Sampling Monte Carlo for System Unavailability Evaluation." *IEEE Transactions on Reliability* R-29(2), 376–380.

Lieber, D. (1998). "Rare-Events Estimation Via Cross-Entropy and Importance Sampling." Ph.D. thesis, The Technion, Israel Institute of Technology, Haifa, Israel.

Lieber, D., R.Y. Rubinstein, and D. Elmakis. (1997). "Quick Estimation of Rare Events in Stochastic Networks." *IEEE Transactions on Reliability* 46, 254–265.

Lomonosov, M. (1994). "On Monte Carlo Estimates in Network Reliability." *Probability in the Engineering and Informational Sciences* 8, 245–264.

Margolin, L. (2002). "Cross-Entropy Method for Combinatorial Optimization." Master's thesis, The Technion, Israel Institute of Technology, Haifa, Israel.

Wittner O. and B.E. Helvik (2002). "Cross Entropy Guided Ant-like Agents Finding Dependable Primary/Backup Path Patterns in Networks." In *Proceedings of Congress on Evolutionary Computation (CEC2002)*, Honolulu, Hawaii, IEEE.

Provan, J. and M. Ball. (1982). "The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected." *SIAM Journal of Computing* 12, 777–787.

Rubinstein, R.Y. and D.P. Kroese. (2004). "The Cross-Entropy Method: A Unified Approach to Combinatorial Optimisation, Monte-Carlo simulation and Machine Learning." Springer.

Rubinstein, R.Y. and B. Melamed (1998). "Modern Simulation and Modeling." Wiley series in probability and Statistics.

Rubinstein, R.Y. (1997). "Optimization of Computer Simulation Models with Rare Events." *European Journal of Operations Research* 99, 89–112.

Rubinstein, R.Y. (1999). "The Cross-Entropy Method for Combinatorial and Continuous Optimization." *Methodology and Computing in Applied Probability* 2, 127–190.

Rubinstein, R.Y. (2001). "Combinatorial Optimization, Cross-Entropy, Ants and Rare Events." In S. Uryasev and P.M. Pardalos (eds.), *Stochastic Optimization: Algorithms and Applications*, Kluwer, pp. 304–358.

Rubinstein, R.Y. (2002). "The Cross-Entropy Method and Rare-Events for Maximal Cut and Bipartition Problems." *ACM Transactions on Modelling and Computer Simulation* 12(1), 303–315.