

# A Combined RESTART - Cross Entropy Method for Rare Event Estimation with Applications to ATM Networks

M.J.J. Garvels\* and R.Y. Rubinstein †

October 9, 2008

\* Faculty of Mathematical Sciences,  
University of Twente, the Netherlands

† Faculty of Industrial Engineering and Management  
Technion—Israel Institute of Technology, Haifa 32000, Israel, and the Institute of Statistical  
Mathematics, 4-6-7 Minami Azabu Minato- Ku, Tokyo, 106-8569, Japan

## Abstract

We present a fast algorithm for the efficient estimation of rare-event (buffer overflow) probabilities in queueing networks. Our algorithm presents a combined version of the two well known methods, called *RESTART/splitting* and *cross-entropy*, where in the latter the optimal change of measure (importance sampling) is determined adaptively during the simulation, using the cross-entropy method. For queueing models we consider two well known importance sampling policies: the so-called *fixed (stable) queueing policy* and the so-called *switching (unstable) queueing policy* and compare their efficiencies. In particular, we show that for the switching policy the combined algorithm is typically much faster than its counterpart, the combined algorithm with fixed policy. We also show that the switching policy combined algorithm reduces, in fact, to a degenerated case presenting a simple single-level cross-entropy algorithm. Simulation results for queueing models of ATM-type are presented. Our numerical results demonstrate high efficiency of the proposed method as compared to the original RESTART/splitting.

**Keywords.** Cross-entropy, Importance Sampling, Rare Events, Simulation, Splitting.

# Contents

<b>1</b>	<b>Introduction to Rare Event Simulation</b>	<b>1</b>
<b>2</b>	<b>The RESTART method</b>	<b>1</b>
<b>3</b>	<b>Combined RESTART and Cross-Entropy Method</b>	<b>3</b>
3.1	Static models . . . . .	6
<b>4</b>	<b>Queueing models</b>	<b>12</b>
4.1	Fixed queueing policy . . . . .	12
4.2	Switching queueing policy . . . . .	14
4.3	Single level algorithm . . . . .	15
4.3.1	The efficiency of RECE for the $M M 1$ queue with unstable IS policy . . . . .	16
4.4	Discussion: stable versus unstable policy . . . . .	17
<b>5</b>	<b>Numerical Results</b>	<b>18</b>
<b>6</b>	<b>Appendix: The Cross-Entropy Method</b>	<b>29</b>
6.1	The Inverse Transform Approach . . . . .	30

# 1 Introduction to Rare Event Simulation

Performance of computer and communication systems is commonly characterized by the occurrence of rare events. For example, cell loss probability in asynchronous transfer mode (ATM) switches is typically less than  $10^{-9}$ . The performance of such systems is frequently studied through simulation. However, estimation of rare event probabilities with the naive Monte Carlo techniques requires a prohibitively large number of trials in most interesting cases. Two methods, called *splitting/RESTART* and *importance sampling* have been extensively investigated by the research simulation community in the last decade.

The basic idea of splitting proposed by Kahn and Marshal [17] is to partition the state-space of the system into a series of nested subsets and to consider the rare event as the intersection of a nested sequence of events. When a given subset is entered by a sample trajectory during the simulation, numerous random retrials are generated with the initial state for each retrial being the state of the system at the entry point. Thus, by doing so, the system trajectory has been split into a number of new sub-trajectories, hence the name splitting.

A similar idea has been developed by Villen-Altamarino and Villen-Altamarino [30], [31] into a refined simulation technique under the name *RESTART* which has been extended by different authors [6], [7], [9], [10], [11], [13], [14], [16], [28], [29] to the multiple threshold case. Although RESTART has been shown to be an efficient and flexible simulation technique for simple networks, its applicability for complex ones is still a challenging problem.

The main idea of IS, when applied to rare events, is to make their occurrence more frequent, or in other words, to "speed up" the simulation. Technically, IS aims to select a probability distribution (change of measure) that minimizes the variance of the IS estimate. Finding the right change of measure is often described by a large deviation result. This type of analysis is feasible only for relatively simple models, see [2], [15], and [27] for surveys. In [19] an adaptive IS algorithm for rare events simulation based on the cross-entropy, called the *cross-entropy* (CE) method, works very efficiently for complex *static* systems, like the stochastic PERT networks. Its applicability to complex dynamic systems such as ATM networks is under investigation. An extremely useful feature of the CE method, is that it can be readily modified for finding (estimating) the optimal solution in an NP-hard combinatorial problem (see Rubinstein [23], [24]).

RESTART is a robust and intuitively simple method for estimating rare event probabilities. It has been proven theoretically and numerically that typically it works well for rare event problems in Garvels et al. [7]. However, it is also well understood now that it requires availability of the so-called *importance function* (IF), which might be an extremely difficult task especially in the multidimensional state space. That is the main reason why we propose a modification of the RESTART method, namely to incorporate CE into RESTART with the view to get efficient and accurate estimations of rare event probabilities.

## 2 The RESTART method

Denote

1.  $\ell$ - the rare event probability.

2.  $m$  - number of stages.
3.  $x_t$ ,  $t = 0, 1, \dots, m$  - intermediate levels ( $x_0 = 0, x_m = x$ ).
4.  $p_t$  - probability of hitting  $x_t$ , starting from  $x_{t-1}$ .
5.  $R_t$  - number of successful hits of level  $x_t$ .
6.  $N_t$  - total number of restarts from level  $x_t$ .

As performance measure of interest (in queueing context) we consider the probability that starting at zero the simulation process reaches level  $x$  before returning back to zero.

In this section we cite some basic results from [6], while considering RESTART with the so-called *fixed effort* (FE). We have

$$\ell = \prod_{t=1}^m p_t, \quad (2.1)$$

$$\hat{\ell} = \prod_{t=1}^m \hat{p}_t, \quad (2.2)$$

where

$$\hat{p}_t = \frac{R_t}{N_t}, \quad (2.3)$$

$R_t = \sum_{i=1}^{N_t} I_i^{(t)}$  is the total number of successes at the  $t$ -th stage,  $I_i^{(t)}$ ,  $i = 1, \dots, N_t$  are the indicators meaning that the underlying process reaches some intermediate level  $x_t$  before returning to level  $x_{t-1}$  and  $p_t = \mathbb{E}\{I_i^{(t)}\}$ .

Our main goal is to present calculation for  $\text{Var}\bar{\ell}$  and to show how much variance can be obtained with RESTART (RE) versus the *crude Monte Carlo* (CMC). We have

$$\text{Var}\hat{\ell} = \mathbb{E}\hat{\ell}^2 - \ell^2 = \mathbb{E}\prod_{t=1}^m \hat{p}_t^2 - \ell^2. \quad (2.4)$$

Taking into account that

$$\text{Var}\hat{p}_t = N_t^{-1}p_t(1 - p_t), \quad (2.5)$$

we obtain

$$\text{Var}\hat{\ell} = \prod_{t=1}^m \left\{ \frac{p_t(1 - p_t)}{N_t} + p_t^2 \right\} - \ell^2 = \ell^2 \left( \prod_{t=1}^m \left\{ \frac{(1 - p_t)}{p_t N_t} + 1 \right\} - 1 \right). \quad (2.6)$$

Note that (2.5) presents, in fact, a CMC estimate of  $p_t$  in (2.1).

To find the optimal parameters  $m$ ,  $(x_1, \dots, x_{m-1})$  and  $(N_1, \dots, N_m)$  we need to solve the following minimization problem

$$\min \text{Var}\hat{\ell} = \min \ell^2 \left( \prod_{t=1}^m \left\{ \frac{(1 - p_t)}{p_t N_t} + 1 \right\} - 1 \right) \quad (2.7)$$

with respect to (w.r.t.)  $m$ ,  $(x_1, \dots, x_{m-1})$  and  $(N_1, \dots, N_m)$ , subject to the following constraint

$$\sum_{t=1}^m N_t = N. \quad (2.8)$$

It is not difficult to show (see [6]) that the solution of (2.7)-(2.8) is  $p_t = p = \ell^{\frac{1}{m}}$  and  $N_t = \frac{N}{m}$ ,  $\forall k = 1, \dots, m$ . With this at hand the program (2.7)-(2.8) reduces to

$$\min_m \text{Var} \hat{\ell} = \min_m \left\{ \frac{\ell^2 m^2 (1 - \ell^{\frac{1}{m}})}{\ell^{\frac{1}{m}} N} \right\}. \quad (2.9)$$

The optimal values of  $m$  and  $p$ , the minimal variance, optimal squared relative error and optimal efficiency, denoted as  $m_r$ ,  $p_r$ ,  $\text{Var}_r \hat{\ell}$ ,  $\kappa_r^2$ , and  $\epsilon_r$  are

$$m_r = -\frac{\log \ell}{2}, \quad (2.10)$$

$$p_r = e^{-2}, \quad (2.11)$$

$$\text{Var}_r \hat{\ell} = \frac{(e \log \ell)^2}{4N}, \quad (2.12)$$

$$\kappa_r^2 = \frac{N^{-1} \text{Var}_r \hat{\ell}}{\ell^2} \approx \frac{(e \log \ell)^2}{4}, \quad (2.13)$$

and

$$\epsilon_r = \frac{N^{-1} \text{Var}_r \hat{\ell}}{\ell(1-\ell)} \approx \frac{\ell(e \log \ell)^2}{4}, \quad (2.14)$$

respectively.

### 3 Combined RESTART and Cross-Entropy Method

The combined RESTART and cross-entropy (RECE) method presents, in fact, a revised version of RESTART in the sense that the Bernoulli parameters  $p_t$  in (2.1) are estimated using likelihood ratios (LR) and the cross-entropy (CE), rather than according to CMC (see (2.5)). The rest is exactly the same as in the original RESTART (RE).

We shall revise now the basic RESTART formulas (2.2)-(2.14) for the RECE version. In particular instead of formulas (2.2)-(2.3) we shall use

$$\tilde{\ell} = \prod_{t=1}^m \tilde{p}_t, \quad (3.1)$$

and

$$\tilde{p}_t = \frac{\tilde{R}_t}{N_t}, \quad (3.2)$$

respectively, where

$$\tilde{R}_t(\bar{\mathbf{v}}_t^*) = \sum_{i=1}^{N_t} I_i^{(t)} W_i^{(t)}(\mathbf{v}, \bar{\mathbf{v}}_t^*), \quad (3.3)$$

$$W_i^{(t)}(\mathbf{v}, \bar{\mathbf{v}}_t^*) = \frac{f(\mathbf{Z}_{it}, \mathbf{v})}{f(\mathbf{Z}_{it}, \bar{\mathbf{v}}_t^*)} \quad (3.4)$$

is the likelihood ratio,  $Z_{it} \sim f(\mathbf{z}, \bar{\mathbf{v}}_t^*)$ ,  $\bar{\mathbf{v}}_t^*$  is the optimal reference parameter, which is obtained either using either the importance sampling [26] or the cross-entropy (CE) method [24]. If not stated otherwise we shall use the cross-entropy method, (see (3.8)-(3.9) below) and assume without loss of generality that  $N_t = N$ .

To derive the optimal  $\bar{\mathbf{v}}_t^*$  we modify the basic formulas (6.3)- (6.7) of the Appendix as follows.

(a) **Adaptive estimation of  $\Delta x_t$ .** For a fixed  $\mathbf{v}_{t-1}^*$  derive  $\Delta x_t^*$  from the following simple one-dimensional root-finding program

$$\max \Delta x_t \text{ s.t. } \mathbb{E}_{\mathbf{v}_{t-1}^*} \left\{ I_{\{\mathcal{M}(\mathbf{Z}) > \Delta x_t\}} \right\} \geq \eta, \quad (3.5)$$

where  $\mathbf{Z} \sim f(\mathbf{y}, \mathbf{v}_{t-1}^*)$ .

The stochastic counterpart of (6.3) is as follows: for fixed  $\bar{\mathbf{v}}_{t-1}^*$  derive  $\bar{\Delta}x_t^*$  from the following program

$$\max \Delta x_t \text{ s.t. } \left\{ \frac{1}{N} \sum_{j=1}^N I_{\{\mathcal{M}(\mathbf{Z}_j) \geq \Delta x_t\}} \right\} \geq \eta, \quad (3.6)$$

where  $\mathbf{Z}_j \sim f(\mathbf{y}, \bar{\mathbf{v}}_{t-1}^*)$ .

Similar to (6.5) we have

$$\bar{\Delta}x_t^* = \bar{\Delta}x_t^*(\bar{\mathbf{v}}_{t-1}^*) = \mathcal{M}_{t,(\lceil(1-\rho)N\rceil)}, \quad (3.7)$$

where  $\mathcal{M}_{t,(j)}$  is the  $j$ -th order statistics of the sequence  $\mathcal{M}_j \equiv \mathcal{M}(\mathbf{Z}_j)$ ,  $\mathbf{Z}_j \sim f(\mathbf{z}, \bar{\mathbf{v}}_t^*)$ ,  $j = 1, \dots, N$ . For  $\eta = 10^{-2}$  this reduces to

$$\bar{\Delta}x_t^* = \mathcal{M}_{t,(\lceil \frac{99}{100}N \rceil)}.$$

(b) **Adaptive estimation of  $\mathbf{v}_t^*$ .** For fixed  $\Delta x_{t-1}^*$  derive  $\mathbf{v}_t^*$  from the solution of the program

$$\max_{\mathbf{v}_t} D(\Delta x_{t-1}^*, \mathbf{v}_{t-1}^*, \mathbf{v}_t) = \max_{\mathbf{v}_t} \mathbb{E}_{\mathbf{v}_{t-1}^*} \left\{ I_{\{\mathcal{M}(\mathbf{Z}) \geq \Delta x_{t-1}^*\}} W(\mathbf{Z}, \mathbf{v}, \mathbf{v}_{t-1}^*) \ln f(\mathbf{Z}, \mathbf{v}_t) \right\}, \quad (3.8)$$

where  $\mathbf{v}_0^* = \mathbf{v}$ .

The stochastic counterpart of (3.8) is as follows: for fixed  $\bar{\Delta}x_{t-1}^*$  derive  $\bar{\mathbf{v}}_t^*$  from the following program

$$\max_{\mathbf{v}_t} \hat{D}_N(\bar{\Delta}x_{t-1}^*, \bar{\mathbf{v}}_{t-1}^*, \mathbf{v}_t) = \max_{\mathbf{v}_t} \left\{ \frac{1}{N} \sum_{j=1}^N I_{\{\mathcal{M}(\mathbf{Z}_j) \geq \bar{\Delta}x_{t-1}^*\}} W(\mathbf{Z}_j, \mathbf{v}, \bar{\mathbf{v}}_{t-1}^*) \ln f(\mathbf{Z}_j, \mathbf{v}_t) \right\}, \quad (3.9)$$

where  $\bar{\mathbf{v}}_0^* \equiv \mathbf{v}$ .

Clearly, with  $\mathbf{v}_t^*$  instead of  $\mathbf{v}$  we obtain

$$\text{Var}\{\tilde{p}_t(\mathbf{v}_t^*)\} \leq \text{Var}\{\tilde{p}_t(\mathbf{v})\} \equiv \text{Var}\{\hat{p}_t(\mathbf{v})\} = N_t^{-1} p_t (1 - p_t), \forall t = 1, \dots, m. \quad (3.10)$$

Because of that we have

$$\text{Var}\{\tilde{\ell}_t(\mathbf{v}_t^*)\} \leq \text{Var}\{\tilde{\ell}_t(\mathbf{v})\} \equiv \text{Var}\{\hat{\ell}_t(\mathbf{v})\}. \quad (3.11)$$

RECE Algorithm 3.1 below can be viewed as an extension of both RESTART and CE:

- RECE extends RESTART in the sense that the sample path (trajectories) and the associated sample function  $\mathcal{M}(\cdot)$  at each level  $x_t$  are generated using the IS density  $f(\mathbf{y}, \bar{\mathbf{v}}_t^*)$  ( $\bar{\mathbf{v}}_t^*$  is obtained from (3.9)) rather than from the original one  $f(\mathbf{y}, \mathbf{v})$ .

- RECE extends CE in the sense that in the former the collected statistics are not wasted and are used to calculate  $\tilde{p}_t$  (via likelihood ratios employing  $\bar{\mathbf{v}}_t^*$ ) at each  $t$ -th iteration, while in the latter the collected statistics are used only at the final stage  $m$  to calculate  $\ell$  (via likelihood ratios employing  $\bar{\mathbf{v}}_m^*$ ).

Note that at each stage  $t$  RECE involves a *three-step* procedure: at the first two steps, a sample is taken from the pdf  $f(\mathbf{y}, \bar{\mathbf{v}}_{t-1}^*)$  to estimate  $\Delta x_t$  and  $\mathbf{v}_t^*$ , while at the third step a different sample is taken from the *importance sampling* (IS) pdf  $f(\mathbf{y}, \bar{\mathbf{v}}_t^*)$  to estimate  $p_t$  in (2.1). Note that for  $\bar{\mathbf{v}}_t^*$  to be a reliable estimate of  $\mathbf{v}_t^*$  the sample must be at least of order  $p_t^{-1}$ .

As in Algorithm 6.1 of the Appendix we set  $\mathbf{v}_1 \equiv \mathbf{v}$  and then iterate in both  $\mathbf{v}_1$  and  $\Delta x = x_t - x_{t-1}$  as follows.

**Algorithm 3.1 :**

1. (*Common step*). Set  $t = 1$  and set  $\bar{\mathbf{v}}_{t-1}^* = \mathbf{v}$ . Generate a sample  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  from the pdf  $f(\mathbf{y}, \mathbf{v})$  and deliver the solution (3.7) of the program (3.6). Denote the initial solution by  $\Delta \bar{x}_1^* \equiv \bar{x}_1^*$ .
2. (*CE step*). Use the **same** sample  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  as in (3.6) and solve the stochastic program (3.9) (for  $t = 1$ ). Denote the solution by  $\bar{\mathbf{v}}_1^* = \bar{\mathbf{v}}_1^*(\Delta \bar{x}_1^*)$ .
3. (*RECE step*). Generate a **new** sample  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  from the pdf  $f(\mathbf{y}, \bar{\mathbf{v}}_1^*)$ , set  $\bar{\mathbf{v}}_t^* = \bar{\mathbf{v}}_1^*$  in (3.3) and deliver  $\tilde{p}_t$  according to (3.2).
4. (*RECE step*). Set  $t = t+1$  and proceed with Step 1-Step 3. At each iteration  $t$ , ( $t = 1, 2, \dots$ ) generate **two different** samples: the first from  $f(\mathbf{y}, \bar{\mathbf{v}}_{t-1}^*)$  and the second from  $f(\mathbf{y}, \bar{\mathbf{v}}_t^*)$ . Calculate  $\Delta \bar{x}_t^*$  and  $\bar{\mathbf{v}}_t^*$  from (3.9) and calculate  $\tilde{p}_t$  from (3.1)-(3.2) using the samples from  $f(\mathbf{y}, \bar{\mathbf{v}}_{t-1}^*)$  and  $f(\mathbf{y}, \bar{\mathbf{v}}_t^*)$ , respectively.
5. (*Stopping rule: CE step*). If  $\bar{x}_t^* \geq x$ , ( $\bar{x}_t^* = \sum_{k=1}^t \Delta \bar{x}_k^*$ ), set  $\bar{x}_t^* \equiv x$  and solve the stochastic program (3.9) for  $\Delta \bar{x}_t^* = x - \bar{x}_{t-1}^*$ . Denote the solution as  $\bar{\mathbf{v}}_{t+1}^*$  and stop; otherwise repeat Step 4.
6. (*Estimating  $\ell$ : RECE step*). Estimate the rare-event probability  $\ell$  according to (3.1)- (3.3).

It is readily seen that

- Deleting Step 2 we obtain  $\bar{\mathbf{v}}_t^* = \mathbf{v}$ ,  $\forall t = 1, 2, \dots$ , and thus, the RESTART algorithm with fixed effort.
- Deleting Step 3, we obtain the CE Algorithm 6.1, provided  $\Delta \bar{x}_t^*$  is replaced by  $\bar{\gamma}_t^*$  and Step 6 of Algorithm 3.1 is replaced by Step 5 of Algorithm 6.1.

Note that instead of deleting Step 2 or Step 3 entirely in Algorithm 3.1 one can keep them for some iterations, say for the first  $t$ , ( $k = 1, \dots, m$ ) iterations (level crossings), where  $m$  denotes the total number of iterations of Algorithm 3.1. For example, including Step 2 only at the first iteration of Algorithm 3.1 and eliminating it for the rest  $m - 1$  iterations, one obtains, in fact,

a RESTART algorithm which uses  $\bar{\mathbf{v}}_1^*$  instead of  $\mathbf{v}$ . We call Algorithm 3.1 the  $[m \times m]$  RECE algorithm, to distinguish it from the  $[m_c \times m]$  RECE algorithm and from the  $[m \times m_r]$  RECE algorithm, where  $m_c$ , ( $m_c = 0, 1, \dots, m$ ) and  $m_r$ , ( $m_r = 0, 1, \dots, m$ ) correspond to the number of times (iterations) Step 2 and Step 3 are included, respectively. Clearly, in the above notations, the configurations  $[0 \times m]$  and  $[m \times 0]$  correspond to the original RESTART and to the original CE, respectively.

At the first glance one might think that RECE needs a sample as twice as big as for the original RE. But, as we will see below, because of (3.12), the sample in RECE is, in fact, less. Clearly if  $m_c = \frac{1}{2}m_r$ , then the total sample sizes in both methods are the same. Note also that the overhead of solving (3.9) is small compared to that of the total number of calculations of  $\mathcal{M}(\mathbf{Z}_i)$  in (3.9). Note finally that regardless of the fact that  $\bar{\mathbf{v}}_t^*$  depends on  $\bar{\mathbf{v}}_{t-1}^*$ ,  $\tilde{\ell}$  in (3.1) will be still a *consistent* (although slightly biased for finite samples) estimate of the unknown parameter  $\ell$ .

Since the optimal parameter vectors  $\mathbf{v}_t^*$  and  $\bar{\mathbf{v}}_t^*$ , derived in (3.8)-(3.9) are typically model dependent we shall calculate them along with some other parameters, such as the minimal variances  $\text{Var}\{\tilde{p}_t(\mathbf{v}_t^*)\}$ ,  $\text{Var}\{\tilde{\ell}_t(\mathbf{v}^*)\}$ , the optimal  $m$  (denoted  $m_c$ ), the optimal  $p$  (denoted  $p_c$ ), the optimal squared relative error  $\kappa^2$  (denoted  $\kappa_c^2$ ) and the optimal relative efficiency  $\epsilon$  (denoted  $\epsilon_c$ ), (see (2.7)-(2.14)), for several simple static and dynamic (queueing) models.

We shall show that in all our examples below

$$m_c \leq m_r, \quad (p_c \leq p_m) \quad (3.12)$$

and the variance reduction obtained with the RECE method compared to the original RESTART method is quite substantial. Similar efficiency might be obtained while using RECE versions for fixed splitting [6] rather than for fixed effort and for other RESTART modifications. In Proposition 3.1 below we prove that formula (3.12) holds for quite general models. The proof of (3.12) is based on (3.10).

### 3.1 Static models

To appreciate the efficiency of RECE versus RESTART we consider several simple static models.

**Example 3.1**  $[1 \times m]$  **RECE configuration.** Suppose we are interested in estimating  $\ell = P(\mathcal{M}(Y) > x)$ , where

$$\mathcal{M}(Y) = Y \quad (3.13)$$

and the random variable  $Y$  is exponentially distributed with rate  $v$ , i.e.  $Y \sim f(y, v) = v \exp(-vy)$ . In this case

$$\ell = e^{-vx} \quad \text{and} \quad p_t = e^{-v\Delta x_t}, \quad (3.14)$$

where as before  $\Delta x_t = x_t - x_{t-1}$ ,  $x_0 = 0$ ,  $x_m = x$ .

Rather than sampling from  $f(y, v) = v \exp(-vy)$ , we assume for convenience that at each level  $x_t$  the sample is taken from the following shifted exponential pdf

$$f(y, v, x_t) = v \exp(-v(y - x_t)), \quad y \geq x_t. \quad (3.15)$$

We proceed with calculation of

- (i)  $\text{Var}\{\tilde{p}_t(\mathbf{v}_1^*)\}$ ,
- (ii)  $\text{Var}\{\tilde{\ell}(\mathbf{v}^*)\}$ ,
- (iii) The optimal  $m_c$  and the optimal  $p_c$ .

(i) Consider  $\text{Var}\{\tilde{p}_t(\mathbf{v}_1)\}$ . Taking (3.3) and (3.4) into account we obtain by straightforward calculations that

$$\text{Var}\{\tilde{p}_t(\mathbf{v}_1)\} = N_t^{-1} \frac{v^2 e^{-(2v-v_t)\Delta x_t}}{v_t(2v-v_t)} - p_t^2. \quad (3.16)$$

The optimal value of the reference parameter  $v_t^* = v_t^*(x)$  which minimizes  $\text{Var}\{\tilde{p}_t(\mathbf{v}_1)\}$  is

$$v_t^*(x) = v + \Delta x_t^{-1} - (v^2 + \Delta x_t^{-2})^{1/2}. \quad (3.17)$$

For  $x_t^{-1} \ll v$  it reduces to

$$v_t^* \approx \Delta x_t^{-1}. \quad (3.18)$$

Substituting (3.18) into (3.16) and taking into account that  $p_t = e^{-v\Delta x_t}$  we obtain

$$\text{Var}\{\tilde{p}_t(\mathbf{v}_1^*)\} = N_t^{-1} \frac{ep_t^2(\ln p_t)^2}{2(-\ln p_t) - 1} - p_t^2 \approx N_t^{-1} \frac{ep_t^2(-\ln p_t)}{2} - p_t^2. \quad (3.19)$$

The efficiency  $\epsilon_t$  of  $\tilde{p}_t(\mathbf{v}_1^*)$  relative to the CMC estimate  $\hat{p}_t(\mathbf{v})$  is

$$\epsilon_t = \frac{\text{Var}\{\tilde{p}_t(\mathbf{v}_1^*)\}}{\text{Var}\{\hat{p}_t(\mathbf{v})\}} \approx (0.5v\Delta x_t)e^{(1-v\Delta x_t)}. \quad (3.20)$$

For example, if  $p_t \equiv e^{-v\Delta x_t} = e^{-2}$ , then we have  $v\Delta x_t = 2$  and, thus

$$\epsilon_t \approx e^{-1}. \quad (3.21)$$

The total variance reduction over  $m$  stages relative to the RESTART method is

$$\epsilon_{cr}^{-1} \approx e^m. \quad (3.22)$$

If, for example,  $\ell = e^{-20}$ , then using RECE instead of RE one obtains variance reduction of order  $e^{10}$ , that is a speed up of more than  $10^4$  times.

From (2.14) it follows that the efficiency of RECE relative to CMC is

$$\epsilon_c = \frac{\epsilon_r}{\epsilon_{cr}} = \frac{\ell(e \log \ell)^2}{4e^m}, \quad (3.23)$$

We shall show next that the optimal value of  $p_t$  for this model is in the  $[1 \times m]$  configuration RECE is  $p_t = e^{-2}$ . To proceed, consider

- (ii)  $\text{Var}\{\tilde{\ell}(v^*)\}$ . Substituting (3.19) into

$$\text{Var}\tilde{\ell} = \mathbb{E}\tilde{\ell}^2 - \ell^2 = \mathbb{E} \prod_{t=1}^m \tilde{p}_t^2 - \ell^2 \quad (3.24)$$

we obtain

$$\text{Var}\tilde{\ell} = \prod_{t=1}^m \frac{ep_t^2(-\ln p_t)}{2N_t} - \ell^2 = \ell^2 \left( \prod_{t=1}^m \frac{e(-\ln p_t)}{2N_t} - 1 \right). \quad (3.25)$$

Solving the program

$$\min \text{Var} \tilde{\ell} = \min \ell^2 \left( \prod_{t=1}^m \frac{e(-\ln p_t)}{2N_t} - 1 \right) \quad (3.26)$$

with respect to  $m$ ,  $(x_1, \dots, x_{m-1})$  and  $(N_1, \dots, N_m)$ , subject to the constraint- (2.8) and taking into account that for large  $N$  we can approximate

$$\prod_{t=1}^m \frac{\ln p_t}{N_t}$$

in (3.25) by

$$\sum_{t=1}^m \frac{\ln p_t}{N_t},$$

we readily obtain that the optimal number of splittings  $m$  equals *unity*, that is there is *no splitting* and the optimal  $p$  equals to  $\ell = e^{-vx}$ . This result is intuitively clear since no computational cost was imposed while estimating  $\mathbf{v}_t^*$  from the solution of the CE program (3.9). This computational cost, denoted  $C_t$ , contains sampling from the original pdf  $f(\mathbf{y}, \mathbf{v})$ ,  $N_t$  times computation of the sample performance  $\mathcal{M}(\mathbf{Y}_j)$  and solving (3.9). We shall take into account below only the first two items and shall ignore the third one, since (3.9) is solvable analytically. As mentioned earlier, in order for  $\bar{\mathbf{v}}_t^*$  to be a reliable estimate of  $\mathbf{v}_1^*$  one needs to take in (3.9) a sample from  $f(\mathbf{y}, \mathbf{v})$  at least of order of  $N_t = O(p_t^{-1})$ , where

$$p_t = \mathbb{E}\{I_{\{\mathcal{M}(\mathbf{Y}) \geq \Delta x_t\}}\}.$$

(iii) With this at hand we assume below that  $C_t = p_t^{-1}$ , denote

$$\mathcal{L}_c = \mathbb{E} \prod_{t=1}^m \tilde{p}_t^2 C_t - \ell^2 \quad (3.27)$$

and modify the program (3.26) as

$$\min \mathcal{L}_c = \min \ell^2 \left( \prod_{t=1}^m \frac{-e \ln p_t}{2N_t p_t} - 1 \right), \quad (3.28)$$

subject to (2.8). Arguing as in (2.7)-(2.9) we readily obtain that  $p_t = p = \ell^{\frac{1}{m}}$ ,  $N_t = \frac{N}{m}$  and, thus  $\mathcal{L}_c$  reduces to

$$\mathcal{L}_c(m) = \ell^2 \left( \frac{em^2 \ln(\ell^{\frac{1}{m}})}{2N \ell^{\frac{1}{m}}} - 1 \right). \quad (3.29)$$

It remains to minimize  $\mathcal{L}_c(m)$  with respect to (w.r.t.)  $m$ . Assuming that  $m$  is a continuous rather than a discrete variable we can approximate the optimal  $m$ , denoted as  $m_c$ , from the solution of

$$\frac{d\mathcal{L}_c(m)}{dm} = 0.$$

We have

$$\frac{d\mathcal{L}_c(m)}{dm} = \ell^{-\frac{1}{m}} \log(\ell^{\frac{1}{m}}) [2m - \log \ell - \frac{\log \ell}{\ell^{\frac{2}{m}} \log(\ell^{\frac{1}{m}})}] = 0.$$

Since the third term in the brackets is negligible compared to the first two ones we can approximate  $m_c$  from the solution of

$$\frac{d\mathcal{L}_c(m)}{dm} = 2m - \log \ell = 0, \quad (3.30)$$

which coincides with (2.10), namely

$$m_c = -\frac{\log \ell}{2}. \quad (3.31)$$

In the analogy to (2.10) we have that the optimal  $p$  is

$$p_c = e^{-2}. \quad (3.32)$$

**Remark 3.1** *If one takes into account the computational cost  $C_t = p_t^{-1} = \ell^{\frac{1}{m}}$ , then the program (2.9) must be replaced in analogy to (3.29) by*

$$\min_m \mathcal{L}_r(m) = \min_m \left\{ \frac{\ell^2 m^2 (1 - \ell^{\frac{1}{m}})}{\ell^{\frac{2}{m}} N} \right\}. \quad (3.33)$$

The optimal solution of (3.33), denoted by  $m_{re}$  is

$$m_{re} = -\log \ell \quad (3.34)$$

and the optimal  $p$ , denoted as  $p_{re}$  is

$$p_{re} = e^{-1}. \quad (3.35)$$

We call the RESTART method based on the program (3.33), the revised RESTART method. It follows from comparison of (3.34) with (2.10) that in the revised RESTART method the optimal number of thresholds must be doubled compared to the original RESTART.

It also follows from (3.31) and (3.34) that the optimal number of thresholds in  $[1 \times m]$  configuration RECE is twice less than in the revised RESTART method. This in turn employs that  $\epsilon_c$  in (3.23) presents a lower bound of the relative efficiency of RECE versus the revised RESTART method.

**Example 3.2 Example 3.1 continued:**  $[1 \times m]$  **RECE configuration** Here we combine the RECE Algorithm 3.1 with *inverse transform - likelihood ratio* (ITLR) approach (see the Appendix). In the particular we apply the configuration  $[1 \times m]$  of RECE for Example 3.1. Note that in this case applying the inverse-transform method

$$Y = F_{\mathbf{v}}^{-1}(U),$$

(see (6.8)) for  $Y \sim \exp(v)$  we obtain

$$Y = -\frac{1}{v} \ln(1 - U), \quad (3.36)$$

where  $U \sim \mathcal{U}(0, 1)$ .

We shall show that in this case our basic result (3.19) holds again, that is

$$m_c = -\frac{\log \ell}{2}.$$

As in Example 3.1 we shall calculate

(i)  $\text{Var}\{\tilde{p}_t(\alpha)\}$ , and (ii)  $\text{Var}\{\tilde{\ell}(\alpha_1^*)\}$ , where  $\alpha_1^*$  is the optimal reference parameter for the Beta ( $\alpha, \beta = 1$ ) pdf given as  $h(y, \alpha) = \alpha y^{\alpha-1}$ . Note that  $\alpha_1^*$  is obtained from the solution of the CE program (3.8), where  $\mathbf{v}_1^*$  is replaced by  $\alpha_1^*$ .

(i) Consider  $\text{Var}\{\tilde{p}_t(\alpha_1^*)\}$ . Taking (3.36) and (6.12) into account we obtain in analogy to (3.16)

$$\text{Var}\{\tilde{p}_t(\alpha_1^*)\} = N_t^{-1} \frac{1 - (1 - p_t)^{2-\alpha}}{\alpha(2-\alpha)}. \quad (3.37)$$

Differentiating (3.37) over  $\alpha$  and equating the result to zero, it can be obtained that for small  $p_t$  we have

$$\alpha_1^* = \frac{1}{p_t}. \quad (3.38)$$

Substituting (3.38) into (3.37) we obtain for small  $p_t$  that

$$\text{Var}\{\tilde{p}_t(\alpha_1^*)\} \approx N_t^{-1} (e-1) p_t^2, \quad (3.39)$$

The efficiency  $\epsilon_t$  of  $\tilde{p}_t(\alpha_1^*)$  relative to the CMC estimate  $\hat{p}_t$ , (corresponding to  $\alpha = 1$ ) follows directly from (3.39) and equals

$$\epsilon_t = \frac{\text{Var}\{\tilde{p}_t(\alpha_1^*)\}}{\text{Var}\{\hat{p}_t(\alpha = 1)\}} \approx (e-1) p_t. \quad (3.40)$$

As in Example 3.1, if  $p_t \equiv e^{-v\Delta x_t} = e^{-2}$ , we have from (3.40) that

$$\epsilon_t \approx (e-1) e^{-2} \quad (3.41)$$

and the total variance reduction over  $m$  stages relative to the RESTART method is

$$\epsilon_{cr}^{-1} \approx (e-1)^{-m} e^{2m}. \quad (3.42)$$

(ii)  $\text{Var}\{\tilde{\ell}(\alpha^*)\}$ . Substituting (3.39) into (3.27) we obtain

$$\mathcal{L}_c = \mathbb{E} \prod_{t=1}^m \tilde{p}_t^2 C_t - \ell^2 = \ell^2 \left( \prod_{t=1}^m \frac{e^{-1}}{N_t p_t} - 1 \right). \quad (3.43)$$

Minimizing  $\mathcal{L}_c$  with respect to  $m$  subject to (2.8) we readily derive in analogy (3.28)- (3.31) that

$$m_c = -\frac{\log \ell}{2}.$$

Since in ITLR the same dominating pdf  $h(\mathbf{y}, \alpha)$  is used for any sample function  $\mathcal{M}(\mathbf{Y})$  (regardless of the original pdf  $f(\mathbf{y}, \mathbf{v})$ ), it can be viewed as a **model independent** approach. It is shown in [19] that using ITLR the results of type (3.41) typically hold for a broad class of sample functions  $\mathcal{M}(\mathbf{Y})$ . For this reason we may argue that

$$m_c = -\frac{\log \ell}{2}$$

holds for a variate of simulation models as well. Based on this, we propose, as rule of thumb, to take the number of levels in  $[1 \times m]$  RECE configuration twice less as in RESTART.

**Example 3.3 Example 3.2 continued:**  $[m \times m]$  **RECE configuration.** We now revise some of the basic formulas (3.39)-(3.43) of Example 3.2 for the  $[1 \times m]$  configuration to its counterpart, the  $[m \times m]$  configuration. To do so we need only to replace  $\bar{\alpha}_1^*$  with  $\bar{\alpha}_t^*$ ,  $t = 2, \dots$  and calculate (3.39)-(3.42) again. Calculating, for example, (3.41), with  $\bar{\alpha}_1^*$  replaced by  $\bar{\alpha}_t^*$ ,  $t = 2, \dots$  we obtain

$$\epsilon_t \approx (e - 1)e^{-2t}. \quad (3.44)$$

The total variance reduction over  $m$  stages relative to the RESTART method ( $[0 \times m]$  configuration) will be now (see also (3.42) for the  $[1 \times m]$  configuration )

$$\epsilon_{cr}^{-1} \approx (e - 1)^{-m} \prod_{t=1}^m e^{2t}. \quad (3.45)$$

The total number of thresholds (level crossings)  $m$  for the  $[m \times m]$  configuration can be derived from the solution of

$$\prod_{t=1}^m e^{-2t} = \ell$$

with respect to  $m$ . Clearly, for small  $\ell$ , the solution of the above will be much less than the number of thresholds

$$m = -\frac{\log \ell}{2}$$

corresponding to the  $[1 \times m]$  configuration. It is also readily seen that in contrast to the  $[1 \times m]$  configuration, where

$$\frac{\Delta x_t}{\Delta x_{t-1}} = 1,$$

here we have

$$\frac{\Delta x_t}{\Delta x_{t-1}} = e^2.$$

That is, instead of linear in  $t$  function  $x_t$ , we have an exponential in  $t$  function  $x_t$ .

We now prove our main Proposition for the configuration  $[1 \times m]$ , which can be readily extended for *any RECE configuration with fixed effort*.

**Proposition 3.1** *Consider the revised RESTART method with fixed effort. Let  $\mathbf{v}_1^*$  be the optimal reference parameter vector obtained from the CE program (3.8). Then the optimal number of thresholds  $m_r$  in the RECE method is no greater than the associated one  $m_{re}$  in the revised RESTART method, that is*

$$m_c \leq m_{re} = -\log \ell. \quad (3.46)$$

**Proof.** The proof is quite simple and is based on (3.10), that is on the fact that one always gets variance reduction with RECE method relative to the revised RE. The formal proof is as follows. Denote  $\text{Var}\{\hat{p}_t(\mathbf{v})\}$  and  $\text{Var}\{\tilde{p}_t(\mathbf{v}_1^*)\}$  by  $V_r(p_t)$  and  $V_c(p_t)$ , respectively and consider the program (2.7) -(2.8). Taking into account that  $V_r(p_t) = N_t^{-1} p_t(1 - p_t)$  we can rewrite (2.7) as

$$\min \text{Var} \hat{\ell}(V_r(p_t)) = N_t^{-1} \min \ell^2 \left( \prod_{t=1}^m \left\{ \frac{V_r(p_t)}{p_t^2 N_t} + 1 \right\} - 1 \right). \quad (3.47)$$

Replace now  $V_r(p_t) = N_t^{-1}p_t(1 - p_t)$  in (3.47) by the alternative one,  $V_c(p_t)$ . As an example of  $V_c(p_t)$  consider the function  $V_c(p_t) \equiv \text{Var}\{\tilde{p}_t(\mathbf{v}_1^*)\}$  in (3.19), and recall that  $V_c(p_t) < V_r(p_t)$ . Since the function  $\text{Var}\hat{\ell}(V_c(p_t))$  decreases in  $V_c(p_t)$  and since  $V_c(p_t) < V_r(p_t)$ , it is readily seen that the optimal solution of the program  $\min \text{Var}\hat{\ell}(V_c(p_t))$ , subject to constraint (2.8), will be no greater than the optimal solution of the program  $\min \text{Var}\hat{\ell}(V_r(p_t))$ , subject to the same constraint (2.8).  $\square$ .

## 4 Queueing models

Before proceeding with application of RECE Algorithm 3.1 for the estimation of rare event probabilities for queueing models we mention the following two well known importance sampling (IS) policies [26]: (a) *fixed or stable IS policy* and (b) *switching or unstable IS policy*.

We explain their differences, while consider, for example, *regenerative* rare event simulation for a single queue. In case (a) we assume that the reference parameter vector  $\mathbf{v}_0$  in IS is fixed and is chosen such that during the simulation the queue remains stable, that is the associated traffic intensity  $\rho_0 < 1$ , while in case (b) the constraint  $\rho_0 < 1$  is removed. Furthermore, we need to distinguish in case (b) the following two possibilities associated while generating a regenerative cycle: (i) the level  $x$  is reached, (ii) the level  $x$  is not reached. In the former case we simulate the queue with some  $\rho_0$  (typically with  $\rho_0 > 1$ , see below) from state zero until the level  $x$  is reached and then we switch back to the nominal traffic intensity  $\rho$  until the busy cycle is completed. In the latter case we simulate the queue with  $\rho_0$  for the entire busy cycle.

Note that using the switching policy in *transient* rare event simulation, that is estimation of the probability  $\ell$  of buffer overflow in a busy cycle starting the system at some initial state, say at an empty system, then we act similarly to the steady-state case. The only difference is, while crossing the level  $x$  with  $\rho_0 > 1$ , we turn the simulation off, rather than we switch back to the nominal traffic intensity to complete the busy cycle.

Although, it is well known [26] that IS estimates with switching policy are typically much faster than their counterpart with fixed policy, we shall nevertheless consider both RECE versions. We shall show, the former reduces, in fact, to a degenerated case - the  $[1 \times 1]$  configuration presenting a *single-level CE algorithm* (see Algorithm 4.1 below). *This is quite a remarkable result!* In addition, it is proved in [4] that for the exponential change of measure the single-level CE Algorithm 4.1 converges to the optimal tilting parameter (state-independent exponential change of measure).

### 4.1 Fixed queueing policy

Here we apply RECE algorithm 3.1 for estimation of  $\ell = P\{L_t > x\}$  using regenerative simulation, where  $\{L_t\}$  is the steady-state *waiting time* process. We start with a stable  $M|M|1$  queue.

**Example 4.1  $M|M|1$  queue: RECE with stable queueing policy** It is well known that

$$\ell = P\{L > x\} = \rho e^{-(\mu-\lambda)x}, \quad (4.1)$$

where  $\mu$  and  $\lambda$  are the service and the inter arrival rates and  $\rho = \frac{\lambda}{\mu}$ . Let  $\mathbf{v}_0 = (\lambda_0, \mu_0)$  be the reference parameter vector in the LR process associated with the  $M|M|1$  queue.

Recall that for the  $M|M|1$  queue the optimal tilted  $\rho_0 = \frac{\lambda_0}{\mu_0}$ , denoted by  $\rho^* = \frac{\lambda^*}{\mu^*}$ , corresponds to  $\lambda^* = \mu$  and  $\mu^* = \lambda$  for both the steady-state and the transient probabilities  $\ell$ .

We cite first some material from [3], where it was assumed that  $\mathbf{v}_0 = (\lambda, \mu_0)$ , that is only the service rate  $\mu$  was allowed to change, while  $\lambda$  was kept fixed. Under this assumption a general expression for the variance of the regenerative estimator of  $\ell(\mu_0, x)$  was derived (see (2.1) of [3]). It is however difficult to treat analytically the expression for the variance in [3] and to get the optimal parameters  $\mu^*$ ,  $\text{Var}\tilde{\ell}(\mu^*)$  and  $m_c$ , similar, say to these in (3.19)-(3.29). To get a good look into insight on how much variance reduction can be obtained with RECE method (versus the original RE), consider the  $[1 \times m]$  configuration and assume as before that  $m_c = m_r$ , where  $m_r = -\frac{\log \ell}{2}$ , ( $p_t = e^{-2}$ ). Consider next the results of Table 3.1 of [3], which presents the efficiency  $\epsilon(\rho^*, \rho, x)$  for different traffic intensities  $\rho$  and different  $x$ . Consider for example  $\rho = 0.6$  and  $\rho = 0.9$ . Then for  $p_t = e^{-2}$  we have from Table 3.1 of [3],  $\epsilon \approx 0.576$  and  $\epsilon \approx 0.536$  for  $\rho = 0.6$  ( $\rho^* = 0.711$ ) and  $\rho = 0.9$  ( $\rho^* = 0.937$ ), respectively. That is, at each stage we obtain variance reduction approximately twice compared to RESTART. The total gain over  $m$  stages will be

$$\epsilon = \left(\frac{1}{0.576}\right)^m \text{ and } \epsilon = \left(\frac{1}{0.536}\right)^m, \quad (4.2)$$

respectively. Note that these are only the lower bounds of the gains, since

1. In [3] only the parameter  $\mu_0$  was assumed to be the reference one in  $\mathbf{v}_0 = (\lambda, \mu_0)$  (and in  $\mathbf{v}_0^* = (\lambda, \mu^*)$ ). Clearly, if we replace  $\lambda$  by  $\lambda^*$  we shall get more variance reduction.
2. We assumed that  $m_c = m_r = -\frac{\log \ell}{2}$ , while, in fact,  $m_c \leq m_r$ . If we use the fact that  $m_c \leq m_r$  instead of  $m_c = m_r$  we shall get more variance reduction. To see it, take in RECE instead of  $p = e^{-2}$ , ( $m = -\frac{\log \ell}{2}$ ), say  $p = e^{-6} \approx 10^{-3}$ , ( $m = -\frac{2 \log \ell}{3}$ ) then we could get per each stage (see Table 3.1 of [3])  $\epsilon = 0.0984$  and  $\epsilon = 0.0122$  for  $\rho = 0.6$  ( $\rho^* = 0.777$ ) and  $\rho = 0.9$  ( $\rho^* = 0.951$ ), respectively. If, for concreteness  $\ell = 10^{-9}$ , then for  $p = e^{-2} \approx 10^{-1}$  and  $p = e^{-6} \approx 10^{-3}$  the total variance reduction would be approximately  $2^9$  and  $10^3$  times for both queues with  $\rho = 0.6$ , and  $\rho = 0.9$ . Clearly that  $10^3 > 2^9$  and, thus, the total gain for  $p = e^{-6}$  would be greater than for  $p = e^{-2}$ .

It is shown numerically in [25] that for  $\ell \approx 10^{-1}$  variance reduction of at *least twice* is guaranteed (with  $\mathbf{v}^*$ ) for quite general queueing models with the  $[1 \times m]$  configuration. This means that using RECE with the  $[1 \times m]$  configuration a total variance reduction of approximately  $2^m$  times will be obtained relative to RESTART.

Note that if we consider the  $[m \times m]$  configuration instead of the  $[1 \times m]$  configuration we will get results similar to Example 3.3, and in particular (see (3.45)) the former is more efficient than the latter. Note also that similar results could be obtained for the  $M|M|1$  queue while considering the transient rare event probability, like the probability of buffer overflow instead of the steady-state one. Note finally that the numerical studies in [25] (see eg., Table 4.4.6 and Table 4.4.7 of [25]) clearly indicate that the efficiency of RECE versus RESTART for the stable IS policy is quite substantial.

## 4.2 Switching queueing policy

As we mentioned we shall show that in this case RECE Algorithm 3.1 reduces to CE, while the last (see Algorithm 4.1 below) presents, in fact a single-step Algorithm. The convergence proof of Algorithm 4.1 to the optimal tilted parameter, as well as confidence intervals for the  $M|M|1$  queue will be given in [4].

**Example 4.2  $M|M|1$  queue: RECE with switching queueing policy** We consider the standard transient buffer overflow probability in a busy cycle, while starting the system at some initial state, say at an empty system. Mathematically we write  $\ell$  as

$$\ell = P(\sup_{t < C} [\mathcal{M}_t \geq x]) = \mathbb{E}_{\mathbf{v}}\{Z\}, \quad (4.3)$$

where  $Z = I_{\sup_{t < C} [\mathcal{M}_t \geq x]}$ ,  $C$  is the length of a busy cycle and  $x$  is the buffer size.

To proceed, we cite some material from Asmussen [1].

1. The buffer overflow probability  $\ell$  under the original distribution is

$$\ell = \mathbb{E}_{\mathbf{v}}Z = \mathbb{E}_{\mathbf{v}}Z^2 = \frac{1 - \rho}{1 - \rho^x} \rho^{x-1}, \quad \rho \neq 1. \quad (4.4)$$

Note that  $\ell = \mathbb{E}_{\mathbf{v}}Z = \mathbb{E}_{\mathbf{v}}Z^2$  and (4.4) holds for both,  $\rho < 1$  and  $\rho > 1$ .

2. Under the tilted traffic  $\tilde{\rho} = \rho^{-1}$  we have asymptotically in  $x$  (see [1]) that

$$\mathbb{E}_{\mathbf{v}}\tilde{Z}^2 = \rho^{2x} \frac{1 - \tilde{\rho}}{1 - \tilde{\rho}^x} \tilde{\rho}^{x-1}. \quad (4.5)$$

Here  $\tilde{Z}$  is the likelihood ratio estimate of  $\ell$ , that is

$$\tilde{Z} = I_{\sup_{t < C} [\mathcal{M}_t \geq x]} \tilde{W}(\tau, \mathbf{v}, \tilde{\mathbf{v}}),$$

$$\tilde{W}(\tau, \mathbf{v}, \tilde{\mathbf{v}}) = \tilde{W}_1 \tilde{W}_2, \quad \tilde{W}_1 = \prod_{i=1}^{\tau} \frac{f_1(Z_1, v_1)}{f_1(Z_1, \tilde{v}_1)}, \quad \tilde{W}_2 = \prod_{i=1}^{\delta(\tau)} \frac{f_2(Z_2, v_2)}{f_2(Z_2, \tilde{v}_2)},$$

$f_1$  and  $f_2$  are the inter-arrival and the service (exponential) pdf's,  $\mathbf{v} = (v_1, v_2)$  and  $\tilde{\mathbf{v}} = (\tilde{v}_1, \tilde{v}_2)$  are the original and the tilted parameters in the pdfs,  $\tau = x + \delta(\tau)$  and  $\delta(\tau)$  is the number of customers, served in a busy cycle before reaching  $x$  (until buffer overflow).

Asmussen, de Boer and Rubinstein [4] prove the so-called *instability theorem* which states that while estimating the transient buffer overflow probability in a M/G/1 queue for an arbitrary buffer size  $x$ , ( $x = 2, \dots$ ) the optimal tilted traffic intensity  $\tilde{\rho}(x)$  is *greater than unity*. Moreover,  $\tilde{\rho}(x)$  increases in  $x$  and for the  $M|M|1$  queue

$$\lim_{x \rightarrow \infty} \tilde{\rho}(x) = \rho^{-1}.$$

An immediate consequence from this theorem is: *while estimating the rare event probability  $\ell$  in (4.3) simulate an **unstable**  $M|G|1$  queue (with  $\tilde{\rho}(x) > 1$ ) rather than a stable one.* This is the reason that the switching policy can be called the unstable policy as well.

It follows from (4.4) that

1.  $\ell(x)$  decreases in  $x$  and  $\ell(x) = 1$  for  $x = 1$ .
2. For a stable  $M|M|1$  queue  $\lim_{x \rightarrow \infty} \ell(x) = 0$ , that is for a stable  $M|M|1$  queue the probability  $\ell(x)$  goes to zero asymptotically in the buffer size  $x$ .
3. For an unstable  $M|M|1$  queue

$$\lim_{x \rightarrow \infty} \ell(x) = \frac{\rho - 1}{\rho}, \quad (4.6)$$

that is for an unstable  $M|M|1$  queue one can, in fact, *hit any arbitrary large level  $x$*  with a positive probability

$$\ell = \frac{\rho - 1}{\rho}.$$

For example, if  $\rho = 1.1$ , we have  $\ell \approx 0.1$ ; and if  $\rho = 2$ , we have  $\ell \approx 0.5$  (asymptotically in  $x$ ). For an  $M|M|1$  queue with a finite buffer, the above probabilities are even higher.

As mentioned, we shall take advantage of the theorem of Asmussen, de Boer and Rubinstein [4] while designing our single level CE Algorithm 4.1 below.

For queueing networks we conjecture that under the optimal tilted parameter vector  $\mathbf{v}_0^*(x)$  ( $x > 1$ ), there exist at least one unstable queue in the network and the probability  $P_{\mathbf{v}_0^*}\{\mathcal{M}_t > x\} \gg P_{\mathbf{v}}\{\mathcal{M}_t > x\}$ . Here  $P_{\mathbf{v}}\{\mathcal{M}_t > x\}$ ,  $\mathcal{M}_t$  and  $\mathbf{v}$  are the underlying rare event probability, the underlying output process and the underlying parameter vector, respectively.

### 4.3 Single level algorithm

Based on (4.3)-(4.6) we now devise an algorithm for efficient estimation of both, the tilted  $\tilde{\rho}(x)$  and the buffer overflow probability  $\ell(x)$  for a fixed  $x$ . Our Algorithm 4.1 is based on the stochastic counterpart of (6.2) that is on

$$\max_{\mathbf{v}_0} \left\{ \hat{D}_N(\mathbf{v}, \mathbf{v}_0) = \frac{1}{N} \sum_{i=1}^N I_{\{\mathcal{M}(\mathbf{Z}_i) > x\}} W(\mathbf{Z}_i, \mathbf{v}, \mathbf{v}_1) \ln f(\mathbf{Z}_i, \mathbf{v}_0) \right\}, \quad (4.7)$$

and can be written as follows

#### Algorithm 4.1 Single level CE :

1. Set  $\mathbf{v}_1 = \mathbf{v}$  and set the initial buffer size  $\gamma_0$  small enough, say  $\gamma_0 = 2$ . Generate a sample  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  from the pdf  $f(\mathbf{y}, \mathbf{v})$  and solve the stochastic program (4.7) for  $x = \gamma_0$ . Denote the solution by  $\tilde{\mathbf{v}}_0^*(\gamma_0)$ .
2. Set  $\mathbf{v}_1 = \tilde{\mathbf{v}}_0^*(\gamma_0)$  and repeat Step 1 for the original program (4.7), that is with  $x$  instead of  $\gamma_0$ . Denote the solution by  $\bar{\mathbf{v}}_1^*$ . Take  $\bar{\mathbf{v}}_1^*$  as an estimate of the optimal reference parameter  $\mathbf{v}_0^*$ .
3. Estimate the rare-event probability  $\ell$  using the LR estimate (6.1) with  $\mathbf{v}_0^*$  replaced by  $\bar{\mathbf{v}}_1^*$ .

#### Remark 4.1 :

- To get a more accurate estimate of  $\mathbf{v}_0^*$  one can proceed with Step 2 for several additional iterations.

- Step 1 of Algorithm 4.1 can be viewed as a *pilot run*. Its aim is twofold:
  1. To find a “good” starting reference parameter vector  $\tilde{\mathbf{v}}_0^*(\gamma_0)$ .
  2. To make the  $M|M|1$  queue unstable.
- Instead of of Algorithm 4.1 one can use either the steepest descent algorithm or a converging to  $\mathbf{v}_0^*$  stochastic approximation algorithm [19].
- As an alternative to Step 1 one can use the following one: Set  $\mathbf{v}_1 = \mathbf{v}$ , choose  $\gamma_0$  small enough, say find  $\gamma_0$  from solution of the stochastic counterpart of  $\mathbb{E}_{\mathbf{v}} \left\{ I_{\{\mathcal{M}(\mathbf{Z}) > \gamma_0\}} \right\} = \eta$ , where, say  $\eta = 10^{-1}$  and solve the stochastic program (4.7) for  $x = \gamma_0$ . Denote the solution by  $\tilde{\mathbf{v}}_0^*(\gamma_0)$ .
- Algorithm 4.1 can be readily modified to estimate the root (say, buffer size  $x$ ) in  $\ell = P\{\mathcal{M} > x\}$  for given  $\ell$ .

It is shown in [4] that the single level Algorithm 4.1 is *robust* in the sense it converges to the optimal tilted parameter vector  $\mathbf{v}_0^*$  regardless of the initial buffer value  $\gamma_0$ , provided  $\gamma_0 > 1$ . Moreover, for a finite sample,  $\bar{\mathbf{v}}_1^*$  estimates quite accurately the unknown vector  $\mathbf{v}_0^*$  for which a valid confidence region can be obtained.

To see why, Algorithm 4.1 converges to  $\mathbf{v}_0^*$  in a *single iteration* we argue as follows. By choosing  $\gamma_0 > 1$  and solving the stochastic program (4.7) for  $\gamma_0 < x$  (see Step 1 of Algorithm 4.1) we obtain  $\tilde{\mathbf{v}}_0^*$  which results in an unstable queue ( $\tilde{\rho}(\gamma_0) > 1$ ). Since with an unstable queue one can reach any large level  $x$  with probability at least  $\ell = \frac{\rho-1}{\rho}$ , we can solve immediately the stochastic program (4.7) (see Step 2 of Algorithm 4.1) for which a sample  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  is generated from the pdf  $f(\mathbf{y}, \tilde{\mathbf{v}}_0^*)$ .

One may wonder, why a pilot run (Step 1 of Algorithm 4.1) is needed at all: just start with an arbitrary  $\mathbf{v}_1$ , which results to an unstable queue. The reason is that not only do we want to make the queue unstable, but also we want  $\mathbf{v}_1$  to be a “good” initial parameter vector, which insures a “wright” proportion of zeros and unities in the indicator  $I$  of the program (4.7). Note that an arbitrary  $\mathbf{v}_1$  (with  $\rho_1 = \frac{\lambda_1}{\mu_1} > 1$ ) may result into *too many unities* in  $I$  and therefore can be viewed as the dual to the original vector  $\mathbf{v}$  in the sense that solving (4.7) with  $\mathbf{v}_1 = \mathbf{v}$  one has *too many zeros* (for large  $x$ ) in the indicator  $I$  and, while solving (4.7) with an arbitrary  $\mathbf{v}_1$ , ( $\rho_1 > 1$ ) one may run into too many unities in  $I$ . Clearly, guessing a ”good” initial  $\mathbf{v}_1^*$  for a queueing network is even more difficult than for a single queue.

### 4.3.1 The efficiency of RECE for the $M|M|1$ queue with unstable IS policy

Denote as before  $x_0 = 0$ ,  $x_m = x$ ,  $\Delta x_t = x_t - x_{t-1}$ ,  $k = 0, \dots, m$ , assuming that in RESTART  $p_t = e^{-2}$  and taking into account that  $1 - \rho^{\Delta x_t} \approx 1$ , we have from (4.3) that

$$\Delta x_t \approx -\frac{2 + \ln(1 - \rho)}{\ln \rho} - 1. \quad (4.8)$$

Let  $\rho = 1/2$ . We have  $\Delta x_t \approx 5$ ,

$$\mathbb{E}_{\mathbf{v}} \mathbf{Z}_t^2 = \frac{1 - \rho}{1 - \rho^{\Delta x_t}} \rho^{\Delta x_{t-1}} \approx 2^{-5},$$

and similarly for other values of  $\rho$ .

Let  $\ell = e^{-M} = e^{-2m}$ , where  $m = M/2$  is the number of level crossings. We have

$$\mathbb{E}_{\mathbf{v}} \mathcal{Z}^2 = \mathbb{E}_{\mathbf{v}} \prod_{t=1}^m \mathcal{Z}_t^2 = \prod_{t=1}^m \mathbb{E}_{\mathbf{v}} \mathcal{Z}_t^2 = \left( \frac{1-\rho}{1-\rho^{\Delta x_t}} \rho^{\Delta x_t - 1} \right)^m.$$

For  $\rho = 1/2$  we have  $\Delta x_t = 5$  and, thus  $\mathbb{E}_{\mathbf{v}} \mathcal{Z}^2 = 2^{-5m}$ .

To see how much variance reduction could be obtained in the first iteration while using RECE instead RESTART consider

$$\epsilon_1 = \frac{\text{Var}_{\mathbf{v}} \mathcal{Z}_1}{\text{Var}_{\mathbf{v}_0^*} \tilde{\mathcal{Z}}_1} \approx \frac{\mathbb{E}_{\mathbf{v}} \mathcal{Z}_1^2}{\mathbb{E}_{\mathbf{v}_0^*} \tilde{\mathcal{Z}}_1^2}. \quad (4.9)$$

Substituting the appropriate values of  $\mathbb{E}_{\mathbf{v}} \mathcal{Z}^2$  and  $\mathbb{E}_{\mathbf{v}_0^*} \tilde{\mathcal{Z}}^2$  from (4.4) and (4.5) we obtain

$$\epsilon \approx \rho^{-1} [\rho^{-x_1} - 1] \approx \left( \frac{1}{\rho} \right)^{x_1+1} \quad (4.10)$$

For  $\rho = 0.5$  we have  $x_1 = 5$  and, thus  $\epsilon_1 \approx (\frac{1}{2})^6$ .

The total efficiency of the single-level RECE versus the  $m$ -level RESTART is

$$\epsilon = \frac{\text{Var}_{\mathbf{v}} \mathcal{Z}}{\text{Var}_{\mathbf{v}_0^*} \tilde{\mathcal{Z}}} \approx \frac{\mathbb{E}_{\mathbf{v}} \mathcal{Z}^2}{\mathbb{E}_{\mathbf{v}_0^*} \tilde{\mathcal{Z}}^2} \approx \left( \frac{1}{\rho} \right)^{m x_1 + 1}. \quad (4.11)$$

For  $\rho = 0.5$ , ( $x_1 = 5$ ) and  $m = 20$ , ( $\ell = e^{-20}$ ) we have  $\epsilon \approx (\frac{1}{2})^{51}$ . Comparing (4.11) and (4.2) it follows that the unstable policy is much more efficient than the stable one. It also follows from the numerical studies in [26] and [25] (compare, for example the results of Table 4.4.6 and Table 4.4.7 of [25] with with these of Table 9.7.6 and Table 9.7.7 of [26]) indicate that RECE with unstable IS policy is more efficient than its stable counterpart.

Note that similar results could be obtained for the  $M|M|1$  queue while considering the steady-state rare event probability instead of the transient one.

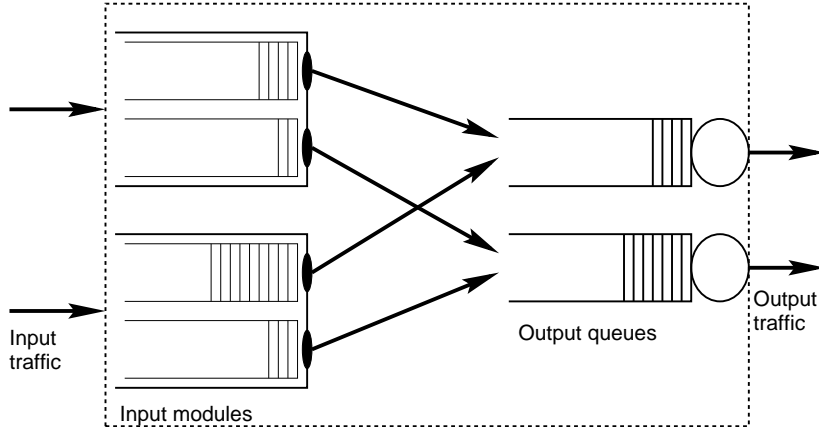
#### 4.4 Discussion: stable versus unstable policy

Although we argued that the unstable IS policy of RECE is more efficient than its stable counterpart, we shall still indicate some important cases where the stable policy might be efficient.

1. The efficiency of the unstable policy versus the stable one was demonstrated for the *state-independent* CE Algorithm 4.1, which in some case fails as compared to its counterpart - the *state-dependent* CE version [5]. For these cases one has to find some alternatives, say to use either the state-dependent CE version as in [5], or to use RECE with stable IS policy as in our numerical results below
2. The unstable single-level RECE may fail to produce accurate rare event probability estimates for queues with deterministic inter-arrival or server times. In this case one can use either the combined *push out* method [26] with RECE unstable version, or again use RECE with stable IS policy.

## 5 Numerical Results

In this section we demonstrate the efficiency of the RECE Algorithm 3.1, while simulating an ATM model, as depicted in the figure below and compare RECE with RESTART.



The above ATM model (the inside of the dashed box) is a realistic one borrowed from ECI (Israel Telecom Corporation). It contains a set of

1. Input queues.
2. Output queues.
3. Input sources

**Input queues** Traffic enters the ATM at one of the  $N$  input modules. These input modules are the physical input queues. However, for proper modeling of the service discipline of the input queues, we introduce virtual input queues inside each input module. We assume that inside each input module there are  $R$  virtual queues, one for each output queue. When a cell enters the input module, its destination port,  $k$  say, is read and the cell is put in virtual queue  $k$  immediately. In reality the cells are all in the same queue, and the service discipline has priorities; however with this virtual queue model we have a FIFO service discipline in each virtual input queue. The input module has a scheduler which decides which virtual queue to serve, provided only one cell per input module can be served at a time. The service time requirement  $S_1$  in the input module per cell is assumed to be constant. The scheduler of each input module serves its virtual queues in a round-robin fashion, skipping virtual queues that are either empty or cannot be serviced right away.

**Output queues** The output queues accept cells from multiple input modules concurrently, up to the moment they reach their capacity, which occurs when the number of cells in the buffer equals  $C$ . After this the output queues only accept new cells from the input modules, provided the number of cells in the output buffer falls below  $C$ . Immediately when the output queue becomes available again, it will poll the input modules in a round-robin fashion in order to find the first

input module that is idle and has cells destined for a particular output queue. Each cell in the output queues has a fixed service time requirement  $S_2$ , after which it leaves the system.

**Input sources** The ECI model assumes that a fixed number of input sources  $B$  are connected to each input module. The sources generate traffic in the form of cells, which are assumed to be identical and independent. Also, the behavior of each input source is identical and independent. Each input source might be either in ‘on’ or ‘off’ state, meaning that they either generate traffic or they are silent, respectively. After each ‘on’ period one ‘off’ period follows and the reverse also holds; the sources’ behavior is thus cyclic with a cycle length equal to the sum of the ‘on’ and ‘off’ period. During the ‘on’ period (assumed to be constant) a fixed number of cells, called *average burst size* (AVBS) is generated at a rate of *peak cell rate* (PCR) cells/sec. The ‘off’ period  $T$  is assumed to be an i.i.d. random variable with little variance, which is expressed as

$$T = \mathbb{E}\{T\} + \text{tol} * \mathbb{E}\{T\} * U(-1, 1). \quad (5.1)$$

Here  $\mathbb{E}\{T\}$  denotes the expected length of an ‘off’-period, tol (tolerance) is a (small) relative deviation around the expectation, and  $U(-1, 1)$  is a standard uniform random variate distributed on  $[-1, 1]$ . The expected length of an ‘off’-period follows directly from the *average cell rate* (ACR):

$$\text{ACR} = \frac{\text{cells per cycle}}{\text{length ‘on’-time} + \text{length ‘off’-time}} = \frac{\text{AVBS}}{\text{AVBS/PCR} + \mathbb{E}\{T\}}. \quad (5.2)$$

Finally, the sources are assumed to be in their steady-state when starting the simulation.

**Stochastic processes** We will now describe the model in stochastic terms.

1. *Source states.*

Since the arrival times of the sources are independent and randomly distributed during the first (initial) period, composed from the length ‘on’-time + length ‘off’-time, each source in the steady-state can be seen as one having a state randomly distributed on the source cycle starting at time  $t = 0$ . The Markovian state of a source can be viewed as the one consisting of

1. The simple ‘on’ or ‘off’ state of the source.
2. The remaining time in the current (‘on’ or ‘off’) state.

Denote the first item, the state of the source number  $i$  at time  $t$ , as  $X_t^{(i)}$ , where  $X_t^{(i)}$  equals unity, when source  $i$  is in the ‘on’-state at time  $t$  and zero otherwise. It is clear that  $X_t^{(i)}, t \geq 0, i = 1, 2, \dots, BN$  present a sequence of i.i.d. Bernoulli random variables with success probability

$$p_i = p = \frac{\text{ACR}}{\text{PCR}}, \quad (5.3)$$

since it presents a ratio of the fixed time spent in the ‘on’-state divided by the average cycle time. The second item (the remaining time) can be easily generated from the steady-state remaining time distribution given the current state  $X_t^{(i)}$ .

2. *Cell destinations.* The  $i$ -th cell entering an input module  $j$  is associated with an output queue destination (random variable), say  $D_i^{(j)}$ ,  $j = 1, 2, \dots, N, i = 1, 2, \dots$ ; these are supposed

to be i.i.d. on  $\{1, 2, \dots, R\}$ . Thus, each cell  $i = 1, 2, \dots$  arriving at each input module  $j = 1, 2, \dots, N$ ,  $D_i^{(j)}$  has a discrete  $R$ -point distribution, denoted  $\text{Multi}(\mathbf{q})$  with the parameter vector  $(q_1, q_2, \dots, q_R)$ , where  $q_i$  is the probability that a random cell will destinate to the output queue  $i$ ,  $i = 1, 2, \dots, R$ .

3. *Buffer contents.* The vector describing the number of cells in the buffers at time  $t$  we denote as  $Q_t^{(i)}$ ,  $i = 1, 2, \dots, (N + 1)R$ . Here the first  $NR$  components correspond to the virtual input queues (clustered per input module) and the last  $R$

corresponds to the output queues. At time  $t = 0$  we assume all buffers to be empty.

The ATM system process  $(\mathbf{Z}_t, t \geq 0)$ , which describes the entire system, consists of the process  $(\mathbf{X}_t, t \geq 0)$ , the process  $(\mathbf{Q}_t, t \geq 0)$ , and the sequence  $(\mathbf{D}_i, i = 1, 2, \dots)$ . Note that we aggregated all related processes and sequences into vector form. Components of each of the vectors can be matched to the queues involved.

The crucial parameter of RESTART/splitting, called the *importance function* [8] is defined as  $\mathcal{M}(\mathbf{Z}_t) := Q_t^{(1)}$  and presents the number of cells in the first virtual queue in the first input module. The importance function can be further associated with the following event  $\{\mathcal{M}(\mathbf{Z}_t) > x\}$ , i.e., the first virtual queue reaches a high level  $x$ . This event is used to make the decision on when to split a sample path into multiple sub-paths.

**Objectives of the simulation** Our goal is to compare the efficiencies of the RESTART with fixed effort, the standard CE and the RECE method for both the transient and the steady-state overflow probability. In the transient simulation the overflow probability, denoted as  $\ell_t(x)$  is  $\ell_t(x) = P\{\mathcal{M}(\mathbf{Z}_t) > x\} = P\{Q_t^{(1)} > x\}$ ,  $0 \leq t \leq T_0$ , (the buffer size  $\mathcal{M}$  in the first queue reaches a high level  $x$  before returning to zero, where the stopping time  $T_0$  is defined as  $\inf_{t>0}\{\mathcal{M}(\mathbf{Z}_t) \leq 0\}$ ), while in the steady-state simulation the overflow probability, denoted as  $\ell_s(x)$  is  $\ell_s(x) = \lim_{t \rightarrow \infty} P\{\mathcal{M}(\mathbf{Z}_t) > x\}$ .

To estimate  $\ell_t(x)$  with RESTART and RECE we use standard splitting (i.e., we split a path whenever  $\mathcal{M}(\mathbf{Z}_t)$  crosses a threshold  $x_k$ ,  $k = 1, 2, \dots, m$ ), and we combine the standard splitting with CE as per Algorithm 3.1, respectively. The steady-state estimate  $\ell_s(x)$  is based on batch-means simulation, as described in [26].

**Importance Sampling (IS)** We apply IS first to a simplified version of the original ATM model, called *Model A*, and then to the original one, called *Model B*.

*Model A* The only difference between model A and model B is that in the former the *burst size* and the *off time* input are changed (see (5.1), (5.2)). Namely,

- *The burst size* is assumed to be distributed  $\text{Geometric}(g_1)$  with  $\frac{1}{g_1}$  equal to the original AVBS.
- *Off time* is assumed to be distributed  $\text{Geometric}(g_2)$  with  $\frac{1}{g_2}$  equal to the original  $T$ , which is scaled by a factor  $1/\text{PCR}$  (the inter-cell transmission time).

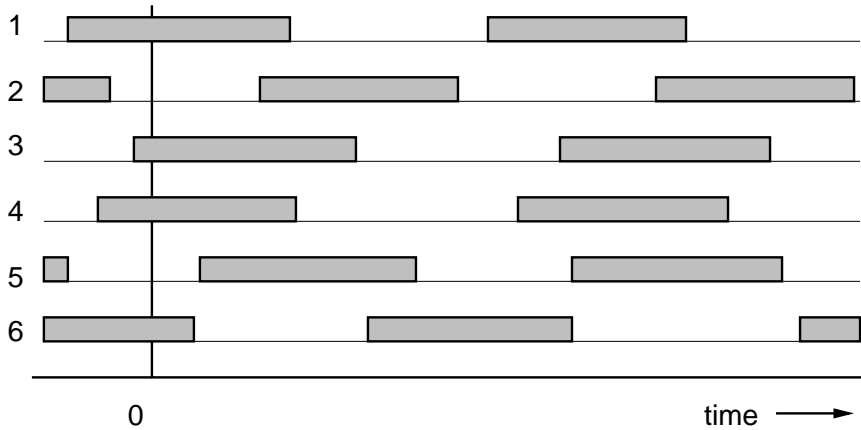
As we shall see below, tilting the above two parameters  $(g_1)$  and  $(g_2)$  of the geometric distribution suffices for RECE Algorithm 3.1 to become a *pure single level CE* one. Clearly by tilting additional distribution parameters one can get more variance reduction.

*Model B* In contrast to model A this model has too little randomness in order for RESTART or CE to work satisfactorily. The combined RECE version, based on RESTART and IS with the tilted parameters updated according to CE works, however quite well, as our numerical results indicate below.

The randomness in our model can be represented by the following four random variables:

1. Initial state of the sources denoted as  $X_0^{(i)}, i = 1, 2, \dots, NR$ .
2. Remaining time in the initial state of the sources.
3. Off-times of the sources defined by the parameter  $tol$  and the uniform random variable  $U(-1, 1)$ , (see (5.1)).
4. Destinations of the cells denoted as  $D_i^{(j)}, j = 1, 2, \dots, N, i = 1, 2, \dots$

Note that the first two items are associated with the initial input distribution, and the last two with run-time behavior. The initial system state is very important, since it determines the level of synchronicity between the sources' cycles. As the level of synchronicity increases, the 'on'-times will overlap more often and as a result, the total burst of the input traffic will increase. This burst is exactly what will cause the buffer to overflow. Notice also that for this model, the steady-state synchronicity is associated with the sources having similar starting states. In fact, the parameter  $p$  (the probability of a source being in the 'on'-state at time zero) is exactly a measure for the synchronicity. We clarify this argument with the figure below, which depicts the sample source dynamics of six sources.



The 'on' periods are shown as a grey area. We can see that the estimate of  $p$ , denoted as  $\bar{p}$ , is  $\bar{p} = 2/3$ , since four sources are active at the steady-state time zero. Clearly when more sources are active at the same time (and thus also at time zero), their cycles are more aligned and the burst of the total traffic is increased.

In our experiments we found little gain in tilting the parameters of distributions of the random variables labeled 2 and 3. We will next describe in more detail tilting for the parameters of distributions of the random variables labeled 1 and 4.

Consider the random variables labeled 1 at  $t = 0$ , namely the random vector  $\mathbf{X}_0 = (X_0^{(i)}, i = 1, 2, \dots, NR)$ . It is distributed  $\text{Ber}(\mathbf{p})$ , where  $\mathbf{p} \in [0, 1]^N$  represents the state of the sources at

time zero, while starting the simulation. (Recall that in the ECI model all components of the vector  $\mathbf{p}$  are defined in (5.3) and equal to  $p$ .) We introduce a tilting vector  $\mathbf{p}_0 \in [0, 1]^N$  and thus, generate from  $Y_0^{(i)} \sim \text{Ber}(p_{J(i)})$ , where we use the notation  $X \sim Y$  when variable  $X$  has the same distribution as variable  $Y$ , and  $J(i)$  denotes the input module number that source  $i$  is connected to. In this fashion, we choose a new success probability  $p_{0j}$  for the state of the source at time zero  $X_0^{(i)}$ , depending on which input module  $j = 1, 2, \dots, N$  it is connected to.

Consider finally the set of random variables  $D_i^{(j)}$ ,  $j = 1, 2, \dots, N$ ,  $i = 1, 2, \dots$  labeled 4 and associated with destination output queues. Note that for  $R = 2$  (two output queues),  $D_i^{(j)}$  is Bernoulli distributed. For  $R > 2$  we have an  $R$  point discrete distribution  $D_i^{(j)} \sim \text{Multi}(\mathbf{q})$  where  $\mathbf{q} = (q_1, q_2, \dots, q_R)$ , and  $q_k$ ,  $k = 1, 2, \dots, R$ , is the probability that a random cell entering the system has a destination output queue  $k$ . We propose to tilt the vector  $\mathbf{q} = (q_1, q_2, \dots, q_R)$  of  $D_i^{(j)}$  as  $\mathbf{q}_0 = (\mu_j, q_2 \frac{1-\mu_j}{1-q_1}, \dots, q_R \frac{1-\mu_j}{1-q_1})$ , that is to tilt only the first probability  $q_1$  and to rescale the others. By doing so we expect occurrence of the overflow in the first virtual queue more often when  $q_1$  increases. This is the same as to say that faster filling of the first output queue will cause a shutdown and subsequent filling up of all the first virtual queues.

Note that the tilted routing probability vector  $\boldsymbol{\mu}$ , is associated with the so-called 'back-pressure', which is associated with the event: the output queue is full and stops accepting cells from the virtual input queues. At this moment the scheduler of the input modules will stop serving the virtual input queue destined for the full output queue, and as a result the number of cells in that virtual input queue builds up rapidly, causing an overflow.

In our simulation studies for the model B we tilted in RECE simultaneously, both the burst and the routing probability parameter vectors, since we found that

1. Tilting only the former, the efficiency (see RTV in our tables below) decreases by one-two orders compared to the efficiency of RECE with both parameter vectors being tilted.
2. Tilting only the latter, the efficiency increases only by one-two orders compared to the efficiency of RESTART.

Note finally that the redundant parameters, like those labeled 2 and 3 (the parameter of the remaining time distribution in the initial state of the sources and the parameter *tol*), were eliminated (screened out) automatically at the early stage of the simulation using the Screening Algorithm proposed in [20].

**Simulation setup** We implemented both, RESTART (with the fixed effort and fixed splitting) and RECE (Algorithm 3.1) in our simulation tool (The simulation machine was a Sun computer, equipped with the Solaris 2.6 operating system, a 336 MHz UltraSparc II processor and 3 GB RAM memory). Thresholds for both methods were found automatically, as well as the optimal IS tilting parameter vector  $\mathbf{v} = (\mathbf{p}_0, \mathbf{q}_0)$  for each stage (level crossing) in RECE simulation. We employed the truncation method in the splitting method as described in [8], [9]. In order to increase the simulation efficiency (by truncating the unpromising paths) we chose the depth parameter [9], [8] equal to three. For RESTART we used  $10^4$  samples per stage to determine the thresholds and  $10^6$  per stage for the actual simulations. For the steady-state simulations we used

the batch-means technique with fixed sampling time of 3200 seconds per run. Note finally that in contrast to model A we used for model B the stable switching policy, since we could not find a parameter setting for which model B becomes unstable.

In our tables below we denote: the estimated probability as  $\hat{\ell}(x)$ , the relative error as  $\text{RE} := \sqrt{\widehat{\text{var}}(\hat{\ell}(x))}/\hat{\ell}(x)$ , and the relative time variance product as  $\text{RTV} := \text{RE}^2 t_{\text{sim}}$ , where  $t_{\text{sim}}$  stands for the total simulation time. Note that the RTV is equal to the time-variance product, (see e.g., [12]). Note also that the ratio of two RTV's, which we will refer to as the gain, measures the relative efficiency of two estimators, like RESTART and RECE and has the following properties:

1. For large sample sizes it converges to a constant.
2. For a fixed relative error  $r$  it becomes  $\text{RTV}(\hat{\ell}_1)/\text{RTV}(\hat{\ell}_2) = t_{\text{sim},2}/t_{\text{sim},1}$ , i.e. the gain is exactly the factor by which the simulation time is reduced by using the estimator  $\hat{\ell}_1$  instead of estimator  $\hat{\ell}_2$ .

We present next simulation results for model A and model B, respectively.

*Model A* We chose the following parameters:  $N = 1, R = 1, B = 1, \text{PCR} = 1.6, \text{AVBS} = 160, \text{ACR} = 0.8, C = \infty$ , that is model A presents, in fact, a single server queue with geometric bursts and geometric off-times. As mentioned before, the original parameter vector  $\mathbf{g} = (g_1, g_2)$  of the geometric distributions, can be readily derived from the AVBS, PCR and ACR parameters. For  $\text{PCR} = 1.6, \text{AVBS} = 160, \text{ACR} = 0.8$ , we have that  $\mathbf{g} = (g_1, g_2) = (.9901, .9901)$ . We used  $10^5$  samples in each stage of RECE and RESTART, and a target success probability of  $e^{-2} \approx 0.14$  per stage, which is optimal for the independent case of RESTART as shown in e.g. [8].

Table 1 presents the steady-state simulation results of both RECE and RESTART methods as functions of the buffer size  $x$ . As expected, we found that RECE Algorithm 3.1 becomes a *single level pure CE* with  $\bar{\mathbf{g}}_0^*$  denoting the estimate of true unknown optimal tilting parameter vector  $\mathbf{g}_0^*$ . We also found that the same phenomenon (reduction of RECE to a single level CE) holds for different values of the two-dimensional vector  $\mathbf{g}$ , as well as tilting (the burst size and the off two-dimensional vector) in the exponential family of distributions [26] instead of the geometric one  $\mathbf{g}$ . Note that for both the RECE and RESTART methods the steady-state parameters were obtained using the duality method as presented in [8] instead of the batch-means method. For the RESTART we used the truncation variant with cut-off depth equal to four [8]. In this way, the number of stages RESTART used is  $\lceil \log(\hat{\ell}_s(x))/\log(e^{-2}) \rceil = \lceil -\frac{1}{2} \log(\hat{\ell}_s(x)) \rceil$ .

It follows from the results of Table 1 that RECE (single level CE) over-performs RESTART in both, variance (RE) and efficiency (RTV). Note, however, that RESTART is quite efficient here as well. This is due to the fact that one can generate the remaining times at the current state from the memoryless geometric distributions and estimate empirically the Bernoulli distribution, which governs the current state upon hitting a threshold.

*Model B* In our simulation results for model B we fixed the following parameters  $p = 0.5, q_1 = 0.5, \text{PCR} = 0.02, \text{ACR} = 0.01, \text{AVBS} = 160, \text{tol} = 0.2, C = 10, S_1 = S_2 = 1$  leaving the parameters  $N, B$  and  $x$  free in order to have more flexibility for the rare event probabilities  $\ell_t(x)$  and  $\ell_s(x)$ .

Tables 2 and 3 present the performance of the RECE and the RESTART methods in the transient and the steady-state regime, respectively, by tilting in RECE both the parameter vector

Level	Optimal RECE				Optimal RESTART		
$x$	$\bar{\mathbf{g}}_0^*$	$\hat{\ell}_s(x)$	RE	RTV	$\hat{\ell}_s(x)$	RE	RTV
2400	(.9944, .9837)	6.471e-8	5.7e-3	7.3e-4	6.221e-8	2.2e-2	3.1e-2
3200	(.9938, .9838)	3.305e-10	5.7e-3	1.1e-3	3.064e-10	2.4e-2	6.0e-2
4000	(.9938, .9838)	1.667e-12	5.3e-3	1.2e-3	1.652e-12	2.7e-2	1.0e-1

Table 1: Steady-state simulation results for the RECE and the RESTART methods.

$\mathbf{p}$  of the Ber ( $\mathbf{p}$ ) and the routing probability vector  $\boldsymbol{\mu}$ .

Parameters			Optimal RECE			Optimal RESTART		
$x$	N	B	$\hat{\ell}_t(x)$	RE	RTV	$\hat{\ell}_t(x)$	RE	RTV
250	2	80	3.770e-7	3.3e-2	2.0e1	3.235e-7	2.4e0	8.0e5
250	3	60	1.356e-7	12.6e-2	1.3e2	8.907e-9	7.0e-1	4.0e4
250	4	45	5.903e-9	9.9e-2	8.3e1	9.582e-8	4.1e-1	1.1e3

Table 2: Transient simulation results for the RECE and the RESTART methods.

Parameters			Optimal RECE			Optimal RESTART		
$x$	N	B	$\hat{\ell}_s(x)$	RE	RTV	$\hat{\ell}_s(x)$	RE	RTV
250	2	80	3.818e-7	4.0e-2	4.4e1	1.777e-8	6.4e-1	1.7e3
250	3	60	4.978e-6	2.7e-2	2.5e1	5.028e-6	3.0e-1	3.2e2
250	4	45	6.812e-7	7.1e-2	1.8e2	4.493e-7	4.5e-1	1.0e4

Table 3: Steady-state simulation results for the RECE and the RESTART methods.

From the results of these tables it is readily seen that in the transient simulation the RECE method improves dramatically over the RESTART method, usually with a gain in RTV over  $10^3$ . The gain in the the steady-state is a little smaller, namely approximately  $10^2$  and is due to the longer simulation runs. Clearly, the RESTART method cannot get much gain since there is very little randomness in the course of the simulation and hitting the rare event heavily depends on the initial distribution. In such a situation [8] the efficiency of the splitting is similar to the CMC. In contrast, the RECE method can change substantially the original parameter vector  $\mathbf{p}$  by an optimal tilting one  $\mathbf{p}_0^*$ , which is the major contributor of the efficiency parameter RTV and,

thus to increase the probability of some sources being in the state ‘on’ at the start of simulation, signifying that the most likely path for an overflow to occur is by having many active sources simultaneously.

Tables 4 and 5 present the optimal parameter setting found from simulation by RECE Algorithm 3.1 for the data in Tables 2 and 3, respectively. Here  $k$ ,  $k = 1, 2, \dots$  denotes the stage (iteration) number,  $x_k$  denotes the upper threshold level in stage  $k$  ( $x_{k-1}$  is a starting threshold level for stage  $k$ ) and  $n_k$  denotes the number of samples used in stage  $k$ .

It follows from the results of Tables 4 and 5 that for  $N = 2$  RECE requires only a single stage and thus reduces again to *pure single level CE*, while for  $N = 3$  and  $N = 4$  RECE requires two stages (iterations). In all cases we see that tilting of the vector  $\mathbf{p}$  is stronger compared to tilting the vector  $\boldsymbol{\mu}$ . The latter is tilted more when the stage width  $\Delta x_k = x_k - x_{k-1}$  is low and is tilted less when  $\Delta x_k$  is large. Another interesting observation for the two-stage procedure ( $k = 2$ ) is that for the transient and the steady-state simulation, the first threshold level  $x_1$  is relatively low and high with respect to  $x$  ( $x = 250$ ), respectively, (compare, for example,  $x_1 = 10$  and  $x_1 = 207$ , given in the last rows of Tables 4 and 5), respectively. This is due to the fact that in the steady-state many paths reach low levels during the fixed simulation path, whereas in the transient case this is much harder due to the very many short samples.

Queues	Stages	Thresholds			Tilting parameters	
		$x_{k-1}$	$x_k$	$n_k$	$\bar{\mathbf{p}}_{0k}^*$	$\bar{\boldsymbol{\mu}}_{0k}^*$
2	1	0	250	1e7	(.6731, .5059)	(.5061, .5016)
3	1	0	10	1e7	(.6389, .5624, .5410)	(.5838, .5537, .5491)
3	2	10	250	1e5		(.5094, .5032, .5028)
4	1	0	10	1e7	(.6872, .5282, .5579, .5376)	(.5785, .5423, .5356, .5263)
4	2	10	250	1e5		(.5090, .5032, .5039, .5025)

Table 4: Optimal parameters found from the transient simulation by RECE Algorithm 3.1 for the data in Table 2.

Table 6 presents simulation results for RECE with  $N = 4$ ,  $B = 45$  and different  $R$  in the transient regime. The parameter setting was the same as in Table 2. The sample size have been multiplied by four in order to obtain accuracy similar to that in Table 2.

Table 7 presents data similar to Table 6 by varying the tolerance parameter  $tol$  for  $N = 4$ ,  $R = 2$  and  $B = 45$ .

It follows from the results of Table 6 that performance of RECE increases (RTV decreases) slightly as  $R$  increases, (the best performance of RECE corresponds to  $R = 4$ , four input and four output queues). It also follows from the results of Table 7 that the output parameters  $\hat{\ell}_t(x)$ , RE and RTV change very little while varying  $tol$ . Note again, that as before, for  $N = 4$ , in both cases RECE reduces to a two-stage procedure, and we tilted simultaneously both parameter vectors  $\mathbf{p}$  and  $\boldsymbol{\mu}$  since we found that the efficiency of RECE is typically worse by at least a factor

Queues	Stages	Thresholds			Tilting parameters	
		$x_{k-1}$	$x_k$	$n_k$	$\bar{\mathbf{p}}_{0k}^*$	$\bar{\boldsymbol{\mu}}_{0k}^*$
2	1	0	250	1e6	(.6605, .5127)	(.5133, .5055)
3	1	0	235	1e6	(.5929, .5377, .5360)	(.5080, .5043, .5009)
3	2	235	250	1e6		(.5099, .5029, .5028)
4	1	0	207	1e6	(.6165, .5467, .5448, .5402)	(.5159, .5080, .5094, .4938)
4	2	207	250	1e6		(.5097, .5010, .5042, .5012)

Table 5: Optimal parameters found from the steady-state simulation by RECE Algorithm 3.1 for the data in Table 3.

Parameters		Optimal RECE		
$x$	R	$\hat{\ell}_t(x)$	RE	RTV
250	2	7.795e-9	7.5e-2	1.9e2
250	3	1.501e-8	6.2e-2	1.3e2
250	4	1.582e-8	5.3e-2	9.5e1

Table 6: Simulation results with RECE in the transient regime for different  $R$  and  $N = 4, B = 45$ .

while tilting only  $\mathbf{p}$ .

Parameters		Optimal RECE		
$x$	tol	$\hat{\ell}_t(x)$	RE	RTV
250	0.2	5.903e-9	9.9e-2	8.3e1
250	0.1	6.347e-9	13.6e-2	1.6e2
250	0.05	6.466e-9	13.2e-2	1.5e2
250	0	5.932e-9	9.0e-2	7.0e1

Table 7: Simulation results with RECE in the transient regime for different values of the tolerance parameter  $tol$  and  $N = 4, R = 2, B = 45$ .

We shall compare now RECE with the standard CE method. Table 8 presents the steady-state simulation results for the RECE and the CE methods, where the RECE results were taken from Table 3, (we omitted the case  $N = 2$  since for  $N = 2$  RECE reduces to the pure CE).

It readily follows from the results of Table 8 that RECE is approximately as two times as

Parameters			Optimal RECE			Optimal CE		
$x$	N	B	$\hat{\ell}_s(x)$	RE	RTV	$\hat{\ell}_s(x)$	RE	RTV
250	3	60	4.978e-6	2.7e-2	2.5e1	4.750e-6	3.3e-2	4.5e1
250	4	45	6.812e-7	7.1e-2	1.8e2	7.513e-7	1.0e-1	4.1e2

Table 8: Steady-state simulation results for the RECE and the CE methods.

efficient as the pure CE method. For completeness, we present in Table 9 the tilting parameters found by the CE algorithm similar to that found in Table 5 by the RECE one.

Queues	Thresholds			Tilting parameters	
	$x_0$	$x_1$	$n_1$	$\bar{P}_{01}^*$	$\bar{\mu}_{0k}^*$
3	0	250	1e6	(.5951, .5358, .5385)	(.5096, .5057, .5026)
4	0	250	1e6	(.5579, .5253, .5164, .5233)	(.5133, .5052, .5038, .5014)

Table 9: Optimal parameters found from the steady-state simulation by CE Algorithm for the data in Table 8 .

Most of the above tables deal with rare events probabilities of order,  $10^{-8}$  and higher. Consider next rare events probabilities of order,  $10^{-11}$  and less. For such small probabilities we found that neither, RECE Algorithm 3.1 with *fixed effort* no the standard CE, estimate reliably the unknown rare event probability in a reasonable time limit. In fact, we found that CE converges too slow, while RECE with fixed effort runs into memory difficulties, which occurs by either increasing the buffer size (decreasing the overflow probability) or/ and by increasing the network size (adding more queues). This happens, since at the end of each stage one needs to save all successful paths' information. As a result, the memory grows linearly in the system state size and in the number of samples in the current stage. To overcome this difficulty we use instead of RECE with *fixed effort* RECE with *fixed splitting*, in which one does not have to fix in advance the number of samples at each stage of Algorithm 3.1, but rather fix the number of offspring (splits) of every successful path. By doing so, the memory problem disappears. That is, in RECE Algorithm 3.1 with fixed splitting, the rare event probability estimation (see step 3) is performed using fixed splitting, while the determination of the tilting parameters and thresholds is performed using fixed effort RECE. For more details on fixed splitting see [8].

Table 10 presents steady-state simulation results for the RECE method with fixed splitting for the following two configurations: (i)  $(N \times R \times B \times x) = (2 \times 2 \times 80 \times 500)$  and (ii)  $(N \times R \times B \times x) = (16 \times 16 \times 13 \times 250)$ , and such that the rare event probability  $\hat{\ell}_s(x)$  is smaller than  $10^{-11}$ . It follows from the results of Table 10 that for both configurations :  $(N \times R \times B \times x) = (2 \times 2 \times 80 \times 500)$  and  $(N \times R \times B \times x) = (16 \times 16 \times 13 \times 250)$  the number of

stages  $k = 2$ . Recall that for the configuration  $N \times R = 2 \times 2$  in all previous tables with  $x = 250$  we obtained  $k = 1$ .

Parameters					Optimal RECE (fixed splitting)		
$x$	$N$	$R$	$B$	$k$	$\hat{\ell}_s(x)$	RE	RTV
500	2	2	80	2	4.053e-13	1.1e-1	4.1e3
250	16	16	13	2	8.702e-12	5.0e-2	6.1e0

Table 10: Steady-state simulation results for the RECE method with fixed splitting.

Table 11 presents the optimal steady-state parameters found by RECE Algorithm 3.1 with fixed splitting at each stage  $k$  for the configuration  $(N \times R \times B \times x) = (16 \times 16 \times 13 \times 250)$ .

Queues	Stage	Thresholds			Tilting parameters	
N	$k$	$x_{k-1}$	$x_k$	$n_k$	$\bar{p}_{0k}^*$	$\bar{\mu}_{0k}^*$
16	1	0	214	4e6	(.8891, .5546, ..., .5445)	(.5282, .4951, ..., .5077)
	2	214	250	19		(.5357, .5969, ..., .5028)

Table 11: Optimal steady-state parameters found by RECE Algorithm 3.1 with fixed splitting for the configuration  $(N \times R \times B \times x) = (16 \times 16 \times 13 \times 250)$ .

Based on the results of these two stage tilting parameter values one can readily see that RECE with fixed splitting indeed handles nicely both, the model size  $(N, R, B)$  and also the overflow level  $x$ , provided the rare event probability is of order  $10^{-11}$ .

Let us turn back to the set of random variables  $D_i^{(j)}$ ,  $j = 1, 2, \dots, N$ ,  $i = 1, 2, \dots$  associated with destination output queues. Intuitively, one would expect a significant change in the vector  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_N)$  relative to the original vector, causing a specific input queue to fill up (to overflow) quicker than the others and even becoming unstable in the sense that RECE becomes a pure CE. This, however, does not happen as Tables 4 and 5 indicate. We explain this fact as follows:  $\boldsymbol{\mu}$  is not changed substantially by the CE method since the likelihood ratio for any successful path consists of exactly one factor caused by  $\mathbf{p}_0$  and thousands of factors caused by  $\boldsymbol{\mu}$ , simply because for every path typically only a single initial state is chosen and thousands of cells need to be routed. In order to keep the variance of the likelihood ratio (LR) low over the successful paths (necessary for efficient importance sampling simulation) clearly  $\boldsymbol{\mu}$  cannot be changed more than slightly, exactly what we observe with the CE algorithm.

At this end, it is worthwhile mentioning that the presented RECE method can be viewed as *dynamic or "stage-dependent" tilting* one in the sense that when the number of trajectories (samples) generated for each threshold coincides with the number of hits no real splitting but rather "stage-dependent" IS occurs.

## 6 Appendix: The Cross-Entropy Method

Here we present the cross-entropy (CE) introduced in [24] for estimating the optimal parameter vector in an importance sampling estimate. Let

$$\ell(x) = P\{\mathcal{M} > x\} = \mathbb{E}I_{\{\mathcal{M} > x\}}$$

represents the probability of the rare-event  $I_{\{\mathcal{M}(\mathbf{Y}) > x\}}$ ,  $\mathcal{M}(\mathbf{Y})$  is the sample performance and  $\mathbf{Y}$  is a given random vector with known distribution. The importance sampling estimate of  $\ell(x) = P\{\mathcal{M} > x\}$  is

$$\bar{\ell}_N(\mathbf{v}, \mathbf{v}_0) = \frac{1}{N} \sum_{i=1}^N I_{\{\mathcal{M}(\mathbf{Z}_i) > x\}} W(\mathbf{Z}_i, \mathbf{v}, \mathbf{v}_0), \quad (6.1)$$

where

$$W(\mathbf{Z}, \mathbf{v}, \mathbf{v}_0) = \frac{f(\mathbf{Z}, \mathbf{v})}{f(\mathbf{Z}, \mathbf{v}_0)}$$

is the likelihood ration and  $\mathbf{v}_0$  is called the reference parameter. To find the optimal reference parameter  $\mathbf{v}_0^*$  which minimizes the cross-entropy (CE),

$$\max_{\mathbf{v}_0} \left\{ D(\mathbf{v}, \mathbf{v}_0) = \mathbb{E}_{\mathbf{v}_1} \left\{ I_{\{\mathcal{M}(\mathbf{Z}) > x\}} W(\mathbf{Z}, \mathbf{v}, \mathbf{v}_1) \ln f(\mathbf{Z}, \mathbf{v}_0) \right\} \right\}, \quad (6.2)$$

we introduce [24] an auxiliary sequence of vectors  $\{\gamma_t\}$ ,  $t \geq 0$  and iterating in both  $\gamma_t$  and  $\mathbf{v}_t$ . We start by choosing, say  $\mathbf{v}_1 = \mathbf{v}$  and some initial  $\gamma_0$  ( $\gamma_0 < x$ ), such that under the original pdf  $f(\mathbf{y}, \mathbf{v})$ , the probability  $\ell(\gamma_0) = \mathbb{E}_{\mathbf{v}} \{ I_{\{\mathcal{M} > \gamma_0\}} \}$  is not too small, say  $\ell(\gamma_0) = \eta \approx 10^{-2}$  and then iterate in both  $\mathbf{v}_1$  and  $\gamma$  as follows.

(a) **Adaptive estimation of  $\gamma_t$ .** For a fixed  $\mathbf{v}_{t-1}^*$  derive  $\gamma_t^*$  from the following simple one-dimensional root-finding program

$$\max \gamma_t \text{ s.t. } \mathbb{E}_{\mathbf{v}_{t-1}^*} \left\{ I_{\{\mathcal{M}(\mathbf{Z}) > \gamma_t\}} \right\} \geq \eta, \quad (6.3)$$

where  $\mathbf{Z} \sim f(\mathbf{y}, \mathbf{v}_{t-1}^*)$  and, say  $10^{-2} \leq \eta \leq 10^{-1}$ .

The stochastic counterpart of (6.3) is as follows: for fixed  $\bar{\mathbf{v}}_{t-1}^*$  derive  $\bar{\gamma}_t^*$  from the following program

$$\max \gamma_t \text{ s.t. } \left\{ \frac{1}{N} \sum_{j=1}^N I_{\{\mathcal{M}(\mathbf{Z}_j) \geq \gamma_t\}} \right\} \geq \eta, \quad (6.4)$$

where  $\mathbf{Z}_j \sim f(\mathbf{y}, \bar{\mathbf{v}}_{t-1}^*)$ .

It is readily seen that

$$\bar{\gamma}_t^* = \bar{\gamma}_t^*(\bar{\mathbf{v}}_{t-1}^*) = \mathcal{M}_{t, \lceil (1-\eta)N \rceil}, \quad (6.5)$$

where  $\mathcal{M}_{t, (j)}$  is the  $j$ -th order statistics of the sequence  $\mathcal{M}_j \equiv \mathcal{M}(\mathbf{Z}_j)$ ,  $\mathbf{Z}_j \sim f(\mathbf{z}, \bar{\mathbf{v}}_t^*)$ ,  $j = 1, \dots, N$ . For  $\eta = 10^{-2}$  this reduces to

$$\bar{\gamma}_t^* = \mathcal{M}_{t, \lceil \frac{99}{100}N \rceil}.$$

(b) **Adaptive estimation of  $\mathbf{v}_t^*$ .** For fixed  $\gamma_{t-1}^*$  derive  $\mathbf{v}_t^*$  from the solution of the program

$$\max_{\mathbf{v}_t} D(\gamma_{t-1}^*, \mathbf{v}_{t-1}^*, \mathbf{v}_t) = \max_{\mathbf{v}_t} \mathbb{E}_{\mathbf{v}_{t-1}^*} \left\{ I_{\{\mathcal{M}(\mathbf{Z}) \geq \gamma_{t-1}^*\}} W(\mathbf{Z}, \mathbf{v}, \mathbf{v}_{t-1}^*) \ln f(\mathbf{Z}, \mathbf{v}_t) \right\}, \quad (6.6)$$

where  $\mathbf{v}_0^* \equiv \mathbf{v}$ .

The stochastic counterpart of (6.6) is as follows: for fixed  $\bar{\gamma}_{t-1}^*$  derive  $\bar{\mathbf{v}}_t^*$  from the following program

$$\max_{\mathbf{v}_t} \widehat{D}_N(\bar{\gamma}_{t-1}^*, \bar{\mathbf{v}}_{t-1}^*, \mathbf{v}_t) = \max_{\mathbf{v}_t} \left\{ \frac{1}{N} \sum_{j=1}^N I_{\{\mathcal{M}(\mathbf{Z}_j) \geq \bar{\gamma}_{t-1}^*\}} W(\mathbf{Z}_j, \mathbf{v}, \bar{\mathbf{v}}_{t-1}^*) \ln f(\mathbf{Z}_j, \mathbf{v}_t) \right\}, \quad (6.7)$$

where  $\bar{\mathbf{v}}_0^* \equiv \mathbf{v}$ .

The resulting algorithm for estimating  $\ell(\gamma)$  can be written as

**Algorithm 6.1 :**

1. Generate a sample  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  from the pdf  $f(\mathbf{y}, \mathbf{v})$  and deliver the solution (6.5) of the program (6.4). Denote the initial solution by  $\bar{\gamma}_0^*$ . Set  $t=1$ .
2. Use the **same** sample  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  as in (6.4) and solve the stochastic program (6.7) for  $\gamma_{t-1} = \bar{\gamma}_{t-1}^*$ . Denote the solution by  $\bar{\mathbf{v}}_t^* = \bar{\mathbf{v}}_t^*(\bar{\gamma}_{t-1}^*)$ .
3. Generate a **new** sample  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  from the pdf  $f(\mathbf{y}, \bar{\mathbf{v}}_t^*)$  and deliver the solution  $\bar{\gamma}_t^*$  in (6.5) of the program (6.4). Denote the solutions by  $\bar{\gamma}_t^*$ .
4. If  $\bar{\gamma}_t^* \geq x$ , set  $\bar{\gamma}_t^* \equiv x$  and solve the stochastic program (6.7) for  $\bar{\gamma}_t^* = x$ . Denote the solution as  $\bar{\mathbf{v}}_{t+1}^*$  and stop; otherwise set  $t \equiv t + 1$  and reiterate from step 2. After stopping:
  - Estimate the rare-event probability  $\ell$  using the LR estimate (6.1), with  $\mathbf{v}_0$  replaced by  $\bar{\mathbf{v}}_{t+1}^*$ .

## 6.1 The Inverse Transform Approach

According to the *inverse transform* (IT) method a random variable  $Y \sim F(y, \mathbf{v})$  can be written as

$$Y = F_{\mathbf{v}}^{-1}(U), \quad (6.8)$$

where  $U \sim \mathcal{U}(0, 1)$  and  $F^{-1}$  is the inverse of the cdf  $F$ .

Formula (6.8) can be readily extended to the multidimensional case. We consider below only the case where the components of  $\mathbf{Y} \sim F(\mathbf{y}, \mathbf{v})$  are independent. We have that  $\mathcal{M}(\mathbf{Y})$  and  $\ell(\mathbf{v}) = P\{\mathcal{M}(\mathbf{Y}) > x\}$  can be written as

$$\mathcal{M}(\mathbf{Y}) = \mathcal{M}(F_{\mathbf{v}}^{-1}(\mathbf{U})) = L(\mathbf{U}, \mathbf{v}) \quad (6.9)$$

and

$$\ell(\mathbf{v}) = \mathbb{E}_{\mathbf{v}}\{I_{\{\mathcal{M}(\mathbf{Y}) > x\}}\} = \mathbb{E}_{\mathbf{U}}\{I_{\{\mathcal{M}(F_{\mathbf{v}}^{-1}(\mathbf{U})) > x\}}\} = \mathbb{E}_{\mathbf{U}}\{I_{\{L(\mathbf{U}, \mathbf{v}) > x\}}\}, \quad (6.10)$$

respectively. Here  $\mathbf{U} = (U_1, \dots, U_n)$  and  $U_k$ ,  $k = 1, \dots, n$  are iid each distributed  $U_k \sim \mathcal{U}(0, 1)$ .

Combining (6.10) with LR, we can present  $\ell(\boldsymbol{\nu})$  and  $\hat{\ell}_N(\boldsymbol{\nu})$  as

$$\ell(\boldsymbol{\nu}) = \mathbb{E}_{\boldsymbol{\nu}}\{I_{\{L(\mathbf{Z}, \boldsymbol{\nu}) > x\}} W(\mathbf{Z}, \boldsymbol{\nu})\}, \quad (6.11)$$

and

$$\hat{\ell}_N(\boldsymbol{\nu}) = N^{-1} \sum_{i=1}^N I_{\{L(\mathbf{Z}_i, \boldsymbol{\nu}) > x\}} W(\mathbf{Z}_i, \boldsymbol{\nu}) , \quad (6.12)$$

respectively. Here

$$W(\mathbf{Z}, \boldsymbol{\nu}) = \frac{1}{h(\mathbf{z}, \boldsymbol{\nu})} \quad (6.13)$$

is the LR,  $h(\mathbf{z}, \boldsymbol{\nu})$  is the pdf dominating the uniform pdf  $\mathcal{U}(0, 1)$ , say

$$h(\boldsymbol{\nu}) = \text{Beta}(\alpha, \beta)$$

and  $\boldsymbol{\nu} = (\alpha, \beta)$  is the *reference parameter* vector. We call (6.12)- (6.13), the *inverse transform - likelihood ratio* (ITLR) estimate.

Note that Algorithm 6.1 remain the some for the ITLR approach, provided the CE programs (6.6) and (6.7) are replaced by

$$\max_{\boldsymbol{\nu}_t} D(\gamma_{t-1}^*, \boldsymbol{\nu}_{t-1}^*, \boldsymbol{\nu}_t) = \max_{\boldsymbol{\nu}_t} \mathbb{E}_{\boldsymbol{\nu}_{t-1}^*} \left\{ I_{\{L(\mathbf{Z}) \geq \gamma_{t-1}^*\}} \ln h(\mathbf{Z}, \boldsymbol{\nu}_t) \right\} , \quad (6.14)$$

and

$$\max_{\boldsymbol{\nu}_t} \hat{D}_N(\bar{\gamma}_{t-1}^*, \bar{\boldsymbol{\nu}}_{t-1}^*, \boldsymbol{\nu}_t) = \max_{\boldsymbol{\nu}_t} \left\{ \frac{1}{N} \sum_{j=1}^N I_{\{L(\mathbf{Z}_j) \geq \bar{\gamma}_{t-1}^*\}} \ln h(\mathbf{Z}_j, \boldsymbol{\nu}_t) \right\} , \quad (6.15)$$

respectively. Here

$$\mathbf{Z}_i \sim h(\mathbf{z}, \bar{\boldsymbol{\nu}}_{t-1}) .$$

## References

- [1] S. Asmussen *Extreme Value Theory for Queues via Cycle Maxima*, Extremes, 1, Kluwer, 137-168, 1998.
- [2] S. Asmussen, and R.Y. Rubinstein *Complexity properties of steady-state rare events simulation in queueing models*, Advances in Queueing: Theory, Methods and Open Problems, (J. Dshalalow, editor), Volume I, CRC Press, 429-462, 1995.
- [3] Asmussen, S., Rubinstein, R.Y. and Wang, Ch., “Estimating Rare Events via Likelihood Ratios: From M/M/1 Queues to Bottleneck Networks”, *Journal of Applied Probability*, Vol. 31, pp. 797–815, 1994.
- [4] Asmussen, S., P. T. de Boer and Rubinstein, R.Y. (2000). Fast Convergence of the Cross-Entropy Method to the Optimal State-Independent Change of Measure. (*in preparation*).
- [5] de Boer, P.-T. (2000), Analysis and efficient simulation of queueing models of telecommunications systems, Ph.D. thesis, University of Twente.
- [6] Garvels M.J.J. and D.P. Kroese (1998), A Comparison of RESTART Implementations, *Proceedings of the 1998 Winter Simulation Conference*, 601–609, Washington, DC.

- [7] M.J.J. Garvels, D.P. Kroese and J.C.W. van Ommeren, On the importance function in splitting simulation, *Proceedings of the 2000 Symposium on Performance Evaluation of Computer and Telecommunications Systems*, 131–138, Vancouver, Canada.
- [8] M.J.J. Garvels (2000), The splitting method in rare event simulation, Ph.D. thesis, University of Twente.
- [9] Glasserman, P. Heidelberger, P. Shahabuddin and T. Zajic (1996), A Look at Multilevel Splitting, *Monte Carlo and Quasi Monte Carlo Methods*, Lecture Notes in Statistics, H. Neiderreiter (ed.) **127**, 99–108.
- [10] Glasserman, P. Heidelberger, P. Shahabuddin and T. Zajic (1997), A Large Deviations Perspective on the Efficiency of Multilevel Splitting, *IEEE Transactions on Automatic Control* **43** (12), 1666–1679.
- [11] Glasserman, P. Heidelberger, P. Shahabuddin and T. Zajic (1999), Multilevel Splitting for Estimating Rare Event Probabilities, *Operations Research* **47** (4), 585–600.
- [12] P.W. Glynn and W. Whitt (1992), The asymptotic efficiency of simulation estimators. *Operations Research* **40**, 505–520.
- [13] Görg C. and O. Fuss (1998), Comparison and Optimization of RESTART Run Time Strategies, *AEÜ* **52** (3), 1–9.
- [14] Görg (1999) C. Simulating Rare Event Details of ATM Delay Time Distributions with RESTART/LRE, *Proceedings of the RESIM Workshop*, 11–12 March, 1999, University of Twente, The Netherlands.
- [15] Heidelberger, P. (1993). “Fast simulation of rare events in queueing and reliability models”. *ACM Transaction of Modeling and Computer Simulation*, Vol. 5, No. 1, 43–85.
- [16] Haraszti, Z and J. Townsend (1999), Rare Event Simulation of Delay in Packet Switching Networks Using DPR-Based Splitting, *Proceedings of the RESIM Workshop*, 11–12 March, 1999, 185–190, University of Twente, The Netherlands.
- [17] Kahn H and T.E. Harris (1951), Estimation of Particle Transmission by Random Sampling, National Bureau of Standards Applied Mathematics Series.
- [18] Krizan, V. and Rubinstein, R., *Polynomial Time Algorithms for Estimation of Rare Events in Queueing Models*, *Frontiers in Queueing: Models and Applications in Science and Engineering* (J. Dshalalow, editor), CRC Press, 421-448, 1997.
- [19] Lieber D. and R. Rubinstein (2000) *Rare-Event Estimation via Cross-Entropy and Importance Sampling* Technion, Manuscript.
- [20] Lieber, D., Rubinstein, R.Y. and Elmyes akis, D. (1997). “Quick estimation of rare events in stochastic networks”, *IEEE Transaction on Reliability*, Vol. 46, No. 2, 254–265.

- [21] Podgaetsky, A. and Rubinstein R.Y., (1999) “ The Cross-Entropy and Rare Events for Maximum Cut and Bipartition Problems”, Manuscript, Technion, Haifa, Israel, 44p.
- [22] Rubinstein, R. Y., (1981) “Simulation and the Monte Carlo method”, Wiley Series in Probability and Mathematical Statistics.
- [23] Rubinstein R.Y., (1999) “Optimization of Noisy Networks via Cross-Entropy” Manuscript, Technion, Haifa, Israel, 53p
- [24] Rubinstein, R. Y. (1999) “ The Cross-Entropy Method for Combinatorial and Continuous Optimization ” *Methodology and Computing in Applied Probability*, 1, 127-190,
- [25] Rubinstein, R.Y. and Shapiro, A. (1993). *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization via the Score Function Method*, John Wiley & Sons, New York.
- [26] Rubinstein, R.Y. and Melamed, B. (1998). *Modern Simulation and Modeling* John Wiley & Sons, New York.
- [27] Shahabuddin, P. (1995). “Rare Event Simulation of Stochastic Systems,” *Proceedings of the 1995 Winter Simulation Conference, Washington, D.C.*, IEEE Press, pps 178-185.
- [28] Schreiber F. and C. Görg (1994), Rare Event Simulation: A Modified RESTART Method Using the LRE-Algorithm, *Proceedings of the 14th International Teletraffic Congress*, 787–796, North Holland.
- [29] Schreiber F and C. Görg (1996), The RESTART/LRE Method for Rare Event Simulation, *Proceedings of the 1996 Winter Simulation Conference*, 390–397, Coronado, California.
- [30] Villén-Altamirano M. and J. Villén-Altamirano (1991), RESTART: A Method for Accelerating Rare Event Simulations, *Proceedings of the 13th International Teletraffic Congress, Queueing, Performance and Control in ATM*, J.W. Cohen (ed.).
- [31] Villén-Altamirano M. and J. Villén-Altamirano About the Efficiency of RESTART, *Proceedings of the RESIM '99 Workshop*, 99–128.