The Rare Event Simulation Method RESTART: Efficiency Analysis and Guidelines for Its Application

Manuel Villén-Altamirano[1] and José Villén-Altamirano[2,*]

[1] Technical University of Madrid
Dep. Ingeniería Sistemas Telemáticos, ETSIT, Ciudad Universitaria, 28040 Madrid, Spain
manolo.villen@dit.upm.es
[2] Technical University of Madrid
Dep. Matemática Aplicada, EUI, c/ Arboleda s/n, 28031 Madrid, Spain
jvillen@eui.upm.es

Abstract. This paper is a tutorial on RESTART, a widely applicable accelerated simulation technique for estimating rare event probabilities. The method is based on performing a number of simulation retrials when the process enters regions of the state space where the chance of occurrence of the rare event is higher. The paper analyzes its efficiency, showing formulas for the variance of the estimator and for the gain obtained with respect to crude simulation, as well as for the parameter values that maximize this gain. It also provides guidelines for achieving a high efficiency when it is applied. Emphasis is placed on the choice of the importance function, i.e., the function of the system state used for determining when retrials are made. Several examples on queuing networks and ultra reliable systems are exposed to illustrate the application of the guidelines and the efficiency achieved.

Keywords: Rare Event, Splitting, RESTART, Simulation, Performance, Reliability.

1 Introduction

Performance requirements of broadband communication networks and ultra reliable systems are often expressed in terms of events with very low probability. Probabilities of the order of 10^{-10} are often used to specify packet losses due to traffic congestion or system failures. Analytical or numerical evaluation of these probabilities is only possible for a very restricted class of systems. Simulation is an effective alternative, but acceleration methods are necessary because crude simulation requires prohibitive execution time for accurate estimation of very low probabilities.

One such method is importance sampling; see [1] for an overview. The basic idea behind this approach is to alter the probability measure governing events so that the formerly rare event occurs more often. A drawback of this technique is the difficulty of selecting an appropriate change of measure since it depends on the system being simulated. Researchers have, therefore, focused on finding good heuristics for particular types of models.

[*] His participation in this work was partially supported by the Comunidad of Madrid, Grant Riesgos CM (P2009/ESP-1685).

D. Kouvatsos (Ed.): Next Generation Internet, LNCS 5233, pp. 509–547, 2011.

Another method is RESTART (REpetitive Simulation Trials After Reaching Thresholds). Let us roughly define the 'importance' of a state as the chance of the process entering the rare set after it has been in this state (a more precise definition of importance will be provided later). RESTART introduces a nested sequence of sets of states C_i ($C_1 \supset C_2 \supset ... \supset C_M$), which determines a partition of the state space Ω into regions $C_i - C_{i+1}$; the higher the value of i, the higher the importance of the states of regions $C_i - C_{i+1}$. A more frequent occurrence of the formerly rare event is achieved by performing a number of simulation retrials each time the process enters a set C_i. The retrials finish when they exit set C_i. Note that while in crude simulation the process spends most of its time in low importance regions, in RESTART simulation an oversampling is made in high importance regions to balance the time spent by the process in all the regions.

The sets C_i are defined by comparing the value taken by a function of the system state, the importance function, with certain thresholds. The application of this method for particular models requires the choice of a suitable importance function. The suitable importance function for a model is not as dependent on particular features of the model as the suitable change of measure required when importance sampling is applied. The paper shows formulas of the importance function for estimating overflow probabilities in Jackson and non-Jackson networks, and also for the study of highly dependable systems.

RESTART has a precedent in the splitting method described in [2]. Splitting also defines importance regions and performs retrials, but these are not made in the same way. They are only made the first time the process enters each set C_i, and they do not finish when they exit set C_i, but continue until the end of the simulation. Consequently, as indicated in [3], oversampling is performed not only in high importance regions, but also in low importance regions that are visited after the higher importance ones, leading to a loss of efficiency. This feature has limited its use to the simulation of processes in which a negligible amount of time is spent in low importance regions visited after the higher importance ones. This amount of time is only negligible in simulations made by means of short replicas, e.g., regenerative simulations of very simple systems, or short transient simulations.

RESTART was introduced by Bayes, A. J. in 1970 [4]. Villén-Altamirano, M. and Villén-Altamirano, J. coined in 1991 the name RESTART [5] and made a theoretical analysis that yields the variance of the estimator and the gain obtained with one threshold. The analysis was extended for multiple thresholds in 1994 [6]. The papers also derive optimal values of the parameters (thresholds and the number of retrials). By using these results, guidelines can be derived for optimizing the importance function and the parameter values. This analysis led to efficient applications of RESTART. While few applications with poor gains [4] or even failures [7] were reported before 1991, a large number of applications with dramatic gains have subsequently been reported. Examples of these applications are [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33] and [34].

The rest of the paper is organized as follows. Section 2 describes the method and Section 3 proves the unbiasedness of the estimator. Section 4 is devoted to show the efficiency of RESTART. Exact formulas for the variance of the estimator and the gain obtained are presented, as well as values for thresholds and the number of retrials that maximize the gain. Section 5 provides guidelines for an effective application of RESTART. It shows how the formula of the gain can be expressed as an ideal gain divided by four factors, which can be considered inefficiency factors. Guidelines are given to reduce each of the factors. Special emphasis is placed on the most critical factor, the one related to the chosen importance function. Section 6 exposes several examples on queuing networks and ultra reliable systems to illustrate the application of the guidelines and the efficiency achieved. Finally, conclusions are stated in Section 7.

2 Description of RESTART

Consider the simulation of a stochastic process $Z = (Z(t),\ t \geq 0)$, with discrete state space and either discrete or continuous parameter. The process may be Markovian or non-Markovian. As in any simulation, regardless of the use of RESTART, $Z(t)$ is simulated by means of a Markovian process $X(t)$ which includes, in addition to the state variables of $Z(t)$, those needed to determine $Z(t_1)$ for $t_1>t$. These additional state variables include:

- The time of occurrence of any future event[1] that has already been scheduled at or before time t;
- The part of the history of the process that has to be incorporated into the system state at t to make $X(t)$ Markovian.

For a given process $Z(t)$, different ways of implementing the simulation model may lead to different processes $X(t)$. Although RESTART may be applied for any process $X(t)$, the application can be more efficient if $X(t)$ is defined following the guidelines given in Section 5.6. In the rest of the paper it is assumed that the process $X(t)$ is given.

Let Ω denote the state space of $X(t)$. A nested sequence of sets of states C_i, $\left(C_1 \supset C_2 \supset \ldots C_M\right)$ is defined, which determines a partition of the state space Ω into regions $C_i - C_{i+1}$; the higher the value of i, the higher the importance of the region $C_i - C_{i+1}$. These sets are defined by means of a function $\Phi : \Omega \rightarrow \Re$, called the importance function. Thresholds T_i $(1 \leq i \leq M)$ of Φ are defined such that each set C_i is associated with $\Phi \geq T_i$.

The probability $\Pr\{A\}$ of the rare set A can be defined in many ways. For example, in a transient simulation, it can be defined as the probability that the system enters the rare set at least once in a given time interval. It is also often defined, both in transient and steady-state simulations, either as the probability of the system being in a state of the set A at a random instant or at the instant of occurrence of certain events denoted reference events. An example of a reference event is a packet arrival. If the rare set is a buffer being full, we are not usually interested in the probability of the buffer being full at a random instant but at a packet arrival. RESTART can be applied in all these

[1] In this paper the term event refers to a simulation event, i.e., an instantaneous occurrence that may change the state Z. The system state resulting from the change will be called system state at the event.

cases. However, for simplicity, the notation will only refer to the last definition. Analogously, the probability $\Pr\{C_i\}$ of set C_i is defined as the probability of the system being in a state of the set C_i at a reference event.

A reference event at which the system is in a state of the set A or set C_i is referred to as an event A or event C_i, respectively. Two additional events, B_i and D_i, are defined as follows:

B_i : event at which $\Phi \geq T_i$ having been $\Phi < T_i$ at the previous event;

D_i : event at which $\Phi < T_i$ having been $\Phi \geq T_i$ at the previous event.

RESTART works as follows:

- A simulation path, called main trial, is performed in the same way as if it were a crude simulation. It lasts until it reaches a predefined "end of simulation" condition.
- Each time an event B_1 occurs in the main trial, the system state is saved, the main trial is interrupted, and $R_1 - 1$ retrials of level 1 are performed. Each retrial of level 1 is a simulation path that starts with the state saved at B_1 and finishes when an event D_1 occurs.
- After the $R_1 - 1$ retrials of level 1 have been performed, the main trial continues from the state saved at B_1. Note that the total number of simulated paths $[B_1, D_1)$, including the portion $[B_1, D_1)$ of the main trial, is R_1. Each of these R_1 paths is called a trial $[B_1, D_1)$. The main trial, which continues after D_1, leads to new sets of retrials of level 1 if new events B_1 occur.
- Events B_2 may occur during any trial $[B_1, D_1)$. Each time an event B_2 occurs, an analogous process is set in motion: $R_2 - 1$ retrials of level 2, starting in B_2 and finishing in D_2, are performed, leading to a total number of R_2 trials $[B_2, D_2)$. The trial $[B_{1,} D_1)$, which continues after D_2, may lead to new sets of retrials of level 2 if new events B_2 occur.
- In general, R_i trials $[B_i, D_i)$ $(1 \leq i \leq M)$ are performed each time an event B_i occurs in a trial $[B_{i-1}, D_{i-1})$. The number R_i is constant for each value of i.
- A retrial of level i also finishes if it reaches the "end of simulation" condition before the occurrence of event D_i. The term trial $[B_i, D_i)$, often used in the rest of the paper, indistinctively refers to a complete or to a prematurely finished trial $[B_i, D_i)$.
- In case that the process up crosses more than one threshold in a time step, it must be taken into account that several events B_i (with different values of i) simultaneously occur. If, for instance, an event at which $\Phi \geq T_{i+1}$ occurs having been $\Phi < T_i$ at the previous event, this event is both an event B_i and an event B_{i+1}. As it is an event B_i, R_i-1 retrials of level i have to be performed starting in this event B_i/B_{i+1} and finishing when an event D_i occurs. As the referred event is also an event B_{i+1} we have to consider that an event B_{i+1} has occurred in each of

the R_i trials $[B_i, D_i)$, thus $R_i(R_{i+1}-1)$ retrials of level i+1 have also to be performed, all of them starting in the referred event B_i/B_{i+1} and finishing when an event D_{i+1} occurs.

Figure 1 illustrates a RESTART simulation with $M = 3$, $R_1 = R_2 = 4$, $R_3 = 3$, in which the chosen importance function Φ also defines set A as $\Phi \geq L$. Bold, thin, dashed and dotted lines are used to distinguish the main trial and the retrials of level 1, 2 and 3, respectively.

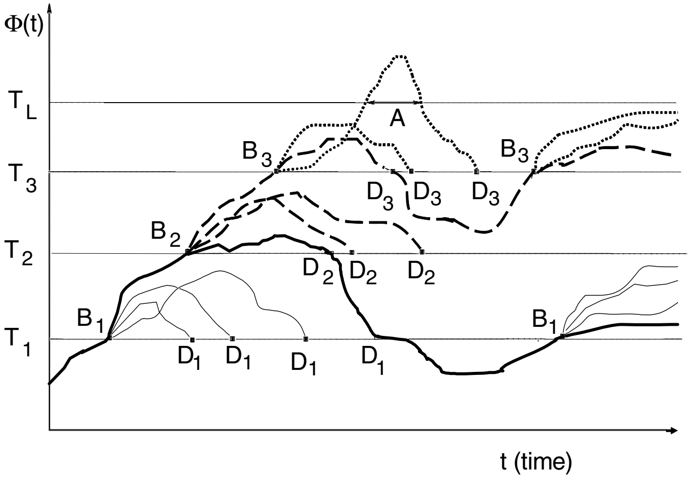


Fig. 1. Simulation with RESTART

Note that the oversampling made by RESTART in the region $C_i - C_{i+1}$ (C_M if $i = M$) is given by the accumulative number of trials:

$$r_i = \prod_{j=1}^{i} R_j \quad (1 \leq i \leq M) .$$

Thus, for statistics taken on all the trials, the weight assigned to the occurrence of an event when it occurs in the region $C_i - C_{i+1}$ (C_M if $i = M$) must be $1/r_i$.

Although sets C_i must be usually chosen satisfying $A \subset C_M$, there are applications where a higher efficiency is achieved if $A \not\subset C_M$ (see [13], [16], [38]). For simplicity the formulas shown in this paper for the variance of the estimator and for the gain obtained only apply to the case in which $A \subset C_M$. Formulas for the variance of the estimator for the general case in which either $A \subset C_M$ or $A \not\subset C_M$ are provided in [36].

The “end of simulation” condition or the condition for the start or the end of a simulation portion (as e.g., the initial transient phase or a batch of a batch means simulation or a replica of a transient simulation) may be defined in the same way as in crude simulation. For example, the condition may be that a predefined value of the simulated time or of the number of simulated reference events is reached. These conditions hold for a trial when the sum of the time (or of the number of reference events) simulated in the trial and in all its predecessors reaches the predefined value.

Some more notations:

- $R_0 = 1$, $r_0 = 1$, $C_0 = \Omega$, $C_{M+1} = A$;
- $P_{h/i}$ $(0 \le i \le h \le M+1)$: probability of the set C_h at a reference event, knowing that the system is in a state of the set C_i at that reference event. As $C_h \subset C_i$, $P_{h/i} = \Pr\{C_h\}/\Pr\{C_i\}$;
- $P_{A/i} = P_{M+1/i}$;
- $P = P_{M+1/0} = P_{A/0} = \Pr\{A\}$;
- N_A: total number of events A that occur in the simulation (in the main trial or in any retrial);
- N_A^0 : number of events A that occur in the main trial;
- N_i^0 $(1 \le i \le M)$: number of events B_i that occur in the main trial;
- N: number of reference events simulated in the main trial;
- a_i $(1 \le i \le M)$: expected number of reference events in a trial $[B_i, D_i)$;
- X_i $(1 \le i \le M)$: random variable indicating the state of the system at an event B_i;
- $\Omega_i (1 \le i \le M)$: set of possible system states at an event B_i;
- $P^*_{A/X_i} (1 \le i \le M)$: importance of state X_i, defined as the expected number of events A in a trial $[B_i, D_i)$ when the system state at B_i is X_i. Note that P^*_{A/X_i} is also a random variable which takes the value P^*_{A/x_i} when $X_i = x_i$;
- $P^*_{A/i}$ $(1 \le i \le M)$: expected importance of an event B_i:

$$P^*_{A/i} = E\left[P^*_{A/X_i}\right] = \int_{\Omega_i} P^*_{A/x_i} dF(x_i) ,$$

  where $F(x_i)$ is the distribution function of X_i. Note that $P^*_{A/i} = E[N_A^0]/E[N_i^0]$ and that $P^*_{A/i} = a_i P_{A/i}$;
- $V\left(P^*_{A/X_i}\right)$ $(1 \le i \le M)$: variance of the importance of an event B_i:

$$V\left(P^*_{A/X_i}\right) = E\left[\left(P^*_{A/X_i}\right)^2\right] - \left(P^*_{A/i}\right)^2 .$$

3 Unbiasedness of the Estimator

The estimator of the probability of the rare set A in a RESTART simulation depends on how this probability has been defined. For the definition adopted in this paper, the estimator for P is in the general case, in which either $A \subset C_M$ or $A \not\subset C_M$:

$$\hat{P} = \tfrac{1}{N}\left(\sum_{i=0}^{M} \frac{N_{Ai}}{r_i}\right), \tag{1}$$

where N_{Ai} is the number of events A occurred in the set $C_i - C_{i+1}$ (C_M if $i = M$) in any trial and N takes a fixed value, which controls the "end of simulation" condition. Note that the weight assigned to N_{Ai} is $1/r_i$ given that N_{Ai} includes events occurred in all the trials, while the weight given to N is 1 since N only includes the reference events occurred in the main trial. In the case that $A \subset C_M$ formula (1) becomes:

$$\hat{P} = \frac{N_A}{N\ r_M}. \tag{2}$$

The unbiasedness of the estimator is proved in (35) for the case in which $A \subset C_M$ and in (36) for the general case. Let us see here the proof made in (35) for $A \subset C_M$. It is made by induction: the estimator of P in a crude simulation is $\hat{P} = N_A^0 / N$, which is an unbiased estimator. As the crude simulation is equivalent to a RESTART simulation with $M=0$, and formula (2) becomes $\hat{P} = N_A^0 / N$ for $M=0$, the estimator of P in a RESTART simulation is unbiased for 0 thresholds. Thus, it is enough to prove that if it is unbiased for M-1 thresholds, it is also unbiased for M thresholds.

Consider a simulation with M thresholds (T_1 to T_M). If the retrials of level 1 (and their corresponding upper-level retrials) are not taken into account, we have a simulation with M-1 thresholds (T_2 to T_M). Let N_A and $\hat{P}$ denote the number of events A and the estimator of P respectively in the simulation with M thresholds, and N_A^{M-1} and $\hat{P}^{M-1}$ the number of events A and the estimator of P in the simulation with M-1 thresholds.

Define α_m as the random variable which indicates the sum of the number of events A occurring in the mth trial [B_1, D_1) performed from each event B_1 of the simulation counting all the events A occurring in the corresponding upper-level retrials. Note that, among the R_1 trials [B_1, D_1) performed from each event B_1 in the M threshold simulation, only the one being a portion of the main trial belongs to the M – 1 threshold simulation. Assigning m = 1 to this trial:

$$N_A^{M-1} = \alpha_1; \quad N_A = \sum_{m=1}^{R_1} \alpha_m .$$

As the R_1 trials [B_1, D_1) made from each event B_1 start with identical system state, $E[\alpha_1] = E[\alpha_2] = ... = E[\alpha_{R_1}]$. Thus, as R_1 is constant, $E[N_A] = R_1\ E\left[N_A^{M-1}\right]$ and consequently:

$$E\left[\hat{P}\right] = \frac{E[N_A]}{N\prod_{i=1}^{M} R_i} = \frac{E\left[N_A^{M-1}\right]}{N\prod_{i=2}^{M} R_i} = E\left[\hat{P}^{M-1}\right] = P.$$

This proves that $\hat{P}$ is also unbiased in a RESTART simulation with *M* thresholds.

It is important to apply the same "end of simulation" condition to the main trial and to all the trials, as explained in Section 2. Otherwise the formula $E[\alpha_1] = E[\alpha_2] = \ldots = E[\alpha_{R_1}]$ would not be satisfied and thus the estimator would not be unbiased.

4 Efficiency of RESTART

The efficiency of an acceleration method is determined by the computational time required for estimating a certain rare event probability *P* with a given width of the confidence interval. As the width of the confidence interval depends on the variance of the estimator, formulas of this variance, $V(\hat{P})$, are shown in Section 4.1 and of the cost (in computational time) of the simulation in Section 4.2. In Section 4.3 the costs incurred by a RESTART simulation and by a crude simulation for estimating a same rare event probability with the same width of the confidence interval are compared to derive the efficiency gain obtained with the application of RESTART. As the efficiency gain depends on the number of thresholds and on the number of retrials used in the RESTART simulation, the values of these parameters that optimize the gain are shown in Section 4.4.

4.1 Variance of the Estimator

The variance of the estimator for the case in which $A \subset C_M$ was derived in [35] and for the general case in which the condition $A \subset C_M$ is not necessarily satisfied in [36]. Let us present here the formula of the variance when $A \subset C_M$ as well as the main steps followed in [35] to derive it. The variance of the estimator is also derived by induction: first a formula is derived for 0 thresholds (crude simulation) and generalized for *M* thresholds; then it is proved that if the generalized formula holds for *M*-1 thresholds, it also holds for *M* thresholds.

Variance for 0 Thresholds (Crude Simulation). In a crude simulation the variance of the estimator is given by:

$$V(\hat{P}) = V\left(\frac{N_A^0}{N}\right) = \frac{V\left(N_A^0\right)}{N^2} = \frac{P}{N}\,\frac{V\left(N_A^0\right)}{E\left[N_A^0\right]} = \frac{K_A P}{N},$$

where $K_A = V\left(N_A^0\right) / E\left[N_A^0\right]$. In simulations defined with a constant time duration *t*, K_A is the index of dispersion on counts, IDC(*t*), of the process of occurrence of events *A* for the time *t* simulated. In any case, K_A is a measure of the autocorrelation of the process of occurrence of events *A*. If the process is uncorrelated, K_A is close to 1 (exactly, $K_A = 1 - P$).

The definition of K_A also applies to a RESTART simulation, where N_A^0 is the number of events *A* in the main trial.

Variance for *M* Thresholds. The variance of $\hat{P}$ in a RESTART simulation with *M* thresholds is given by:

$$V(\hat{P})=\frac{K_A P}{N}\left[\frac{1}{r_M}+\sum_{i=1}^{M}\frac{s_i P_{A/i}(R_i-1)}{r_i}\right], \tag{3}$$

with:

$$s_i=\frac{1}{K_A P_{A/i}}\frac{V\left(E\left[N_A^0|\chi_i\right]\right)}{E\left[N_A^0\right]} \quad (1\le i\le M)\,, \tag{4}$$

where $\chi_i=\left(N_i^0,\left(X_i^1,X_i^2,...,X_i^{N_i^0}\right)\right)$, N_i^0 being the random variable indicating the number of events B_i occurred in the main trial of a simulation randomly taken and $(X_i^1,X_i^2,...,X_i^{N_i^0})$ being the vector of random variables describing the system states at those events B_i. A further development of formula (4) is shown later to gain insight on s_i.

The formula provided for 0 thresholds is an application of formula (3) to the case of $M = 0$ (where $r_M = r_0 = 1$). Let us now see that if formula (3) holds for *M*-1 thresholds it also holds for *M* thresholds.

Consider the two related *M*-1 and *M* threshold simulations described in Section 3. The variance of the estimator in the *M*-1 threshold simulation $V(\hat{P}^{M-1})$ and in the *M* threshold simulation $V(\hat{P})$ can be written as:

$$V(\hat{P}^{M-1})=V\left(E\left[\hat{P}^{M-1}|\chi_1\right]\right)+E\left[V(\hat{P}^{M-1}|\chi_1)\right]\ . \tag{5}$$

$$V(\hat{P})=V\left(E[\hat{P}|\chi_1]\right)+E[V(\hat{P}|\chi_1)]\,. \tag{6}$$

An intuitive explanation of formulas (5) and (6) is that the variance $V(\hat{P})$ for both *M*-1 and *M* threshold simulations can be considered to be the result of two contributions, reflected by the two terms of each of these formulas:

- The first term reflects the variance associated with the set of events B_1 occurred in the simulation. As the number of retrials made in B_1 does not affect this variance, the first term of the two formulas is equal for both crude and RESTART simulation.
- The second term reflects the variance of the number of events *A* occurred when the set of events B_1 is given. As this variance is reduced by performing retrials in B_1, the second term is R_1 times smaller for the *M* threshold simulation than for the *M*-1 threshold simulation.

This intuitive reasoning, which is confirmed in a rigorous way in [35], leads to:

$$V\left(E\left[\hat{P}|\chi_1\right]\right) = V\left(E\left[\hat{P}^{M-1}|\chi_1\right]\right). \tag{7}$$

$$E\left[V(\hat{P}|\chi_1)\right] = \frac{1}{R_1} E\left[V\left(\hat{P}^{M-1}|\chi_1\right)\right]. \tag{8}$$

Note that, as $V(\hat{P}^{M-1})$ does not depend on χ_1, an increment of the first term of formula (5) due to a different χ_1 leads to a decrement of the same value of the second one. Consequently an increment of the first term of formula (6) leads to a decrement R_1 times smaller of the second one. It means that a greater variance of the importance at events B_1 makes less efficient the application of RESTART. An explanation of this fact is that a greater variance of the importance at events B_1 leads to a higher correlation between trials made from a given B_1 and, as the retrials made from a given B_1 are less effective if they are correlated, the application of RESTART is less efficient. The same applies to a great variance of the importance at events B_i for any other given value of i.

Starting with the formula of $V(\hat{P}^{M-1})$, obtained by adapting formula (3) to the case of M-1 thresholds numbered from 2 to M, and using formulas (5), (6), (7) and (8), formula (3) for the variance of the estimator $V(\hat{P})$ is derived in [35].

Analysis of Factors. s_i**.** In order to gain insight on factor s_i, formula (4) of this factor has been further developed in [35], leading to:

$$s_i = \frac{a_i}{K_A}\left[K'_i + \frac{V(P^*_{A/X_i})}{(P^*_{A/i})^2}\gamma_i\right] \qquad (1 \le i \le M)\,, \tag{9}$$

where $K'_i = V(N_i^0)/E[N_i^0]$ $(1 \le i \le M)$ and:

$$\gamma_i = 1 + 2\sum_{m=1}^{\infty} \frac{E\left[Max\left(0, N_i^0 - m\right)\right]}{E\left[N_i^0\right]} \, \frac{ACV_m\left(P^*_{A/X_i}\right)}{V\left(P^*_{A/X_i}\right)} \qquad (1 \le i \le M),$$

$ACV_m\left(P^*_{A/X_i}\right)$ being the autocovariance of P^*_{A/X_i} at lag m.

Let us analyze formula (9):

- Factor K'_i: This factor is a measure of the autocorrelation of the process of the occurrence of events B_i in the main trial. If the process is uncorrelated, K'_i is close to 1 (exactly, $K'_i = 1 - P_{i/0}/a_i$). In most applications, the process has a weak positive autocorrelation and K'_i is slightly greater than 1.
- Factor γ_i: If the random variables X_i were independent all the covariances $ACV_m\left(P^*_{A/X_i}\right)$ would be zero and thus $\gamma_i = 1$. In general, γ_i is a measure of the dependence of the importance of the system states X_i of events B_i occurring in

the main trial. In most practical applications, there may be some dependence between system states of close events B_i but this dependence is negligible for distant events B_i. Thus γ_i is usually close to 1 or at least of the same order of magnitude as 1.

- Ratio $V\left(P^*_{A/X_i}\right)\Big/\left(P^*_{A/i}\right)^2$: It greatly depends on the chosen importance function and may have an important impact on the efficiency of RESTART. An ideal choice of the importance function and of the process $X(t)$ would lead to $V\left(P^*_{A/X_1}\right)=0$ and thus $s_1 = K_1/K_A$, which is around 1 in many applications. Thus values of $s_1 >> 1$ could indicate inefficiency in the application of RESTART due to an improper choice of the importance function.

4.2 Simulation Cost

Let us define the cost C of a simulation as the computational time required for the simulation, taking as time unit the average computational time per reference event in a crude simulation of the system. With this definition of time unit, the cost of a crude simulation with N reference events is $C = N$.

In a RESTART simulation, the average cost of a reference event is always greater as overheads are involved in the implementation of RESTART: (1) for each event, an overhead mainly due to the need to evaluate the importance function and to compare it with the threshold values, and (2) for each retrial, an overhead mainly due to the restoration of event B_i (which includes to restore the system state at B_i and, as explained in Section 5.6, to re-schedule the scheduled events). To account for these overheads, the average cost of a reference event in a RESTART simulation is inflated (1) by a factor $y_e > 1$ in any case and (2) by an additional factor $y_{ri} >1$ if the reference event occurs in a retrial of level i.

Using the above definition of time unit, the average cost per reference event is $y_0 = y_e$ in the main trial and $y_i = y_e\, y_{ri}$ $(1 \leq i \leq M)$ in a retrial of level i. As the expected number of reference events in the retrials of level i of a RESTART simulation (with N reference events in the main trial) is $N\, P_{i/0} r_{i-1}(R_i - 1)$, the expected cost of the simulation is:

$$C = N\left[y_0 + \sum_{i=1}^{M} y_i P_{i/0} r_{i-1}(R_i - 1)\right]. \tag{10}$$

Remark: Factors y_i affect the simulation cost when it is measured in terms of required computational time, but not when it is measured in terms of number of events to be simulated. In this case $y_i = 1 \quad (0 \leq i \leq M)$.

4.3 Simulation Gain with RESTART

A measure of the efficiency for computing $\hat{P}$ is given by the relative confidence-normalized cost, *RCNC,* which is defined as $C\, V(\hat{P})\big/P^2$. To compare the *RCNC* of several estimators is equivalent to comparing the computational costs for a fixed relative width of the confidence interval. *RCNC* is equal to K_A / P in crude simulation, given that $V(\hat{P}) = K_A\ P/N$ and $C = N$, and can be obtained from formulas (3) and (10) in RESTART simulation.

The gain G obtained with RESTART can be defined as the ratio of the RCNC with crude simulation to the *RCNC* with RESTART. Defining $s_0 = 0$, $s_{M+1} = 1$ and $y_{M+1} = 0$ the following formula of the gain is obtained:

$$G = \frac{1}{P\left(\sum_{i=0}^{M} \frac{s_{i+1}u_i\left(1-P_{i+1/i}\right)}{P_{i+1/0}r_i}\right)\left(\sum_{i=0}^{M} y_i v_{i+1} P_{i/0} r_i \left(1-P_{i+1/i}\right)\right)}, \tag{11}$$

where:

$$u_i = \frac{1-\frac{s_i}{s_{i+1}}P_{i+1/i}}{1-P_{i+1/i}}\ ;\ v_{i+1} = \frac{1-\frac{y_{i+1}}{y_i}P_{i+1/i}}{1-P_{i+1/i}} \qquad (0 \le i \le M)\ .$$

4.4 Quasi-Optimal Parameters

To maximize the gain G in formula (11), factors s_i and y_i must be minimized and optimal values for $P_{i/0}$ (or equivalently $P_{i+1/i}$) and r_i need to be derived. Let us focus in this section on the optimal values of $P_{i+1/i}$ and r_i. These optimal values, that are function of s_i and y_i, have been derived in [35]. However, in a practical application, the values of s_i and y_i are difficult to evaluate. Therefore, approximations of the optimal values of $P_{i+1/i}$ and r_i that are independent of s_i and y_i and given by simple expressions are recommended. As these approximations of the optimal parameters provide a gain close to that obtained with the optimal ones they are called quasi-optimal parameters. These parameters have also been derived in [35] assuming that the product $s_{i+1}\, u_i$ takes the same value for every i ($0 \le i \le M$) and that the same occurs for the product $y_i\, v_{i+1}$. With these assumptions, quasi-optimal parameters maximizing the gain have been derived from (11) in these three steps:

1. For fixed values of $P_{i+1/i}$, quasi-optimal values of r_i are derived. For deriving them the derivative of the gain in formula (11) with respect to r_i is made equal to zero for $1 \le i \le M$ and the resulting system of equations is solved. The solution obtained is:

   $$r_i = \sqrt{\frac{1}{P_{i/0}P_{i+1/1}}} \qquad (1 \le i \le M)\ . \tag{12}$$

   In practice, as the number of retrials R_i must be integer, a value close to that given by (12) that satisfies this restriction must be chosen for r_i

2. For these values of r_i quasi-optimal values of $P_{i+1/i}$ for a fixed number of thresholds are derived. For this purpose, r_i is substituted in (11) by the second term of

(12) and $P_{1/0}$ by $P\Big/\prod_{i=1}^{M} P_{i+1/i}$. The derivative of the resulting expression of the gain with respect to $P_{i+1/i}$ is made equal to zero for $1 \le i \le M$, obtaining $P_{i+1/i} = P_{1/0}$. It means that for a fixed number of thresholds "quasi-optimal" gain is obtained when all the probabilities $P_{i+1/i}$ have the same value, which is:

$$P_{i+1/i} = P^{\frac{1}{M+1}} \qquad (0 \le i \le M)\,. \tag{13}$$

3. For these values of r_i and $P_{i+1/i}$ quasi-optimal value of M is derived. Substituting also (13) in (11), we can observe that the larger the value of M, the greater the gain. Thus $P_{i+1/i}$ must be as close as possible to 1, i.e., the thresholds must be set as close as possible. In practice, there are two limitations on how close the thresholds can be set: one is due to the values that Φ can take when it is a discrete function; the other is due to the restrictions on the value of R_i derived from the chosen thresholds. This value must be an integer number greater than one, given that $R_i = 1$ means that T_i is not really a threshold.

The quasi-optimal gain, obtained when r_i and $P_{i+1/i}$ are given by (12) and (13) respectively and M tends to infinite, is given by:

$$G = \frac{1}{P\left(-AVG(s)\ln P + 1\right)\left(-AVG(y)\ln P + y_0\right)}, \tag{14}$$

where $AVG(s)$ and $AVG(y)$ are the arithmetical means of s_i and y_i $(1 \le i \le M)$ respectively.

5 Guidelines for an Effective Application of RESTART

The quasi-optimal gain given by formula (14) assumes that quasi-optimal parameters are used. In practice, quasi-optimal parameters are not possible: the importance function may be discrete and it prevents from setting thresholds with $P_{i+1/i}$ very close to 1; even when the importance function is continuous it is not possible to set infinite thresholds, as mentioned above; moreover, the evaluation of r_i is based on an estimation of $P_{i/0}$. Although this estimation can be made by means of pilot runs, there will be always some error in the estimation and thus r_i will not be exactly the quasi-optimal one. In addition the resulting R_i has to be rounded to an integer number. This section studies how the gain is affected by the errors and limitations in the setting of the optimal parameters as well as by the computer overhead produced by the implementation of RESTART and by the chosen importance function. Section 5.1 defines four factors reflecting the influence of these features in the gain and Sections 5.2 to 5.6 analyze each of the factors and provide guidelines for reducing them.

5.1 Factors Affecting the Efficiency of RESTART

As indicated in [35]. the general formula of the gain (formula (11)) can be re-written as follows:

$$G=\frac{1}{f_V f_O f_R f_T}\,\frac{1}{P\left(-\ln P+1\right)^2}, \tag{15}$$

where:

$$f_V=\frac{\sum_{i=1}^{M}\frac{s_i\left(R_i-1\right)}{P_{i/0}\,r_i}+\frac{s_{M+1}}{P\,r_M}}{\sum_{i=1}^{M}\frac{R_i-1}{P_{i/0}\,r_i}+\frac{1}{P\,r_M}}, \qquad f_O=\frac{\sum_{i=1}^{M} y_i\,P_{i/0}\,r_{i-1}\left(R_i-1\right)+y_0}{\sum_{i=1}^{M} P_{i/0}\,r_{i-1}\left(R_i-1\right)+1}. \tag{16}$$

$$f_R=\frac{\left(\sum_{i=1}^{M}\frac{R_i-1}{P_{i/0}\,r_i}+\frac{1}{P\,r_M}\right)\left(\sum_{i=1}^{M} P_{i/0}\,r_{i-1}\left(R_i-1\right)+1\right)}{\left(\sum_{i=0}^{M}\frac{1-P_{i+1/i}}{\sqrt{P_{i+1/i}}}+1\right)^2}, \quad f_T=\frac{\left(\sum_{i=0}^{M}\frac{1-P_{i+1/i}}{\sqrt{P_{i+1/i}}}+1\right)^2}{\left(-\ln P+1\right)^2}. \tag{17}$$

The term $1/\left(P\left(-\ln P+1\right)^2\right)$ can be considered the ideal gain, which matches with the quasi-optimal gain (formula (14)) when $s_i = 1$ $\left(1\le i\le M\right)$ and $y_i = 1$ $\left(0\le i\le M\right)$. Factors f_V, f_O, f_R and f_T, all of them equal to or greater than 1 (with the exception of f_V which could be smaller than 1 in some cases), can be considered inefficiency factors that reduce the actual gain with respect to the ideal one. Each factor reflects:

- f_V: inefficiency due to the variance of the importance of the systems states at each B_i which in its turn is due to the non-optimal choice of the Markovian process $X(t)$ used for simulating the original process $Z(t)$ (see Section 2) and/or the non-optimal choice of the importance function;
- f_O: inefficiency due to the computer overhead produced by the implementation of RESTART;
- f_R: inefficiency due to the non-optimal choice of the number of retrials;
- f_T: inefficiency due to the non-optimal choice of the thresholds.

Note that the ideal gain $1/\left(P\left(-\ln P+1\right)^2\right)$ takes very high values, e.g., $4.6{\bullet}10^3$ for $P=10^{-6}$, $1.7\cdot 10^7$ for $P=10^{-10}$ and $9.1\cdot 10^{10}$ for $P=10^{-14}$. Assuming a computational time of 0.1 msec. per reference event, to estimate these probabilities with crude simulation would require a computational time of 11 hours, 13 years and 127 millennia respectively. Applying RESTART these times are reduced, assuming that the ideal gain is achieved, to 9, 23 and 44 secs. respectively. In practice these times will be greater due to the inefficiency factors but, if these factors take moderate values, the resulting computational time may be low even though their values are not close to 1.

5.2 Analysis and Guidelines to Reduce Factor f_R

Let η_i denote the ratio of the actual value of r_i to its quasi-optimal value r_{iqo} given by (12), and $\eta_{\max}$ and $\eta_{\min}$ the maximum and minimum values of η_i respectively:

$$\eta_i = \frac{r_i}{r_{iqo}} \quad (1 \le i \le M); \quad \eta_0 = 1 .$$

$$\eta_{\max} = \max_{0 \le i \le M} (\eta_i); \; \eta_{\min} = \min_{0 \le i \le M} (\eta_i) .$$

Based on left-hand formula (17) and on this notation, the following bound of f_R is derived in [35]:

$$f_R \le \frac{\eta_{\max}}{\eta_{\min}} . \tag{18}$$

This bound allows providing guidelines for assigning values to r_i taking into account that R_i must be integer. For given thresholds, (assuming that a value has already been assigned to $r_{i\text{-}1}$) the value that must be assigned for r_i is: r_i = $r_{i\text{-}1}$ R_i where $R_i = r_{iqo} / r_{i-1}$ (rounded). R_i must be rounded to its integer part $\lfloor R_i \rfloor$ or to $\lfloor R_i \rfloor + 1$ depending on which alternative leads to the minimum value of $Max(\eta_i, 1/\eta_i)$.

Formula (18) also indicates that the impact on the gain of a non-optimal choice of r_i due to errors in the estimation of $P_{i/0}$ is moderate if the errors are not very large; thus a rough estimation of $P_{i/0}$ may be sufficient for this purpose.

5.3 Analysis and Guidelines to Reduce Factor f_T

Based on right-hand formula (17), the following bound of f_T is derived in [35]:

$$f_T \le \frac{(1 - P_{\min})^2 / P_{\min}}{(\ln P_{\min})^2} , \tag{19}$$

where:

$$P_{\min} = \underset{0 \le i \le M}{Min} (P_{i+1/i}) .$$

The value of f_T is moderate even for values of P_{min} far from 1. For example, $f_T \le 1.04$ for P_{min} = 0.5, $f_T \le 1.53$ for P_{min} = 0.1 and $f_T \le 4.62$ for P_{min} = 0.01. It means that the impact on the gain of a discrete importance function is moderate except in the case that the thresholds have to been set very far each other.

Consequently the following guidelines may be provided for setting thresholds: if the importance function is continuous, thresholds should be set at a distance given by $P_{i+1/i} = 0.5$, given that it leads to R_i = 2 without need of rounding while f_T is only 1.04. If the importance function is discrete and the probability ratio of consecutive values of Φ is greater than 0.5, thresholds that lead to R_i = 2 with minimum rounding should be set. If the probability ratio of consecutive values of Φ is smaller than 0.5, a threshold should be set for each value of Φ.

5.4 Analysis and Guidelines to Reduce Factor f_O

According to right-hand formula (16), factors f_O is a weighted mean of y_i $\left(0 \le i \le M\right)$, all the weights being positive. Thus, the following bound of f_O can be defined:

$$f_O \le \underset{0 \le i \le M}{Max} \left(y_i\right).$$

Factor f_O reflects the inefficiency due to the overhead produced by the implementation of RESTART, as explained in Section 4.2. The value taken depends on the system characteristics. For example, it is higher when the system state is described by many variables because it increases the overhead needed for restoring the system state at B_i.

Factor f_O can be reduced by the use of hysteresis, which reduces the number of events B_i in the simulation and by following some programming guidelines for reducing the overhead per event B_i. The use of hysteresis consists in defining for each threshold T_i an additional threshold $T_i' < T_i$ and extending the retrials of level i until $\Phi < T_i'$ (see, e.g., [37]). Guidelines to reduce the overhead per event B_i are explained in [37]. They are:

- To perform memory dump for saving or restoring the state at B_i instead of copying the system variables one by one;
- To perform a joint scheduling of all the pending events with negative exponentially distributed time of occurrence. When several of these events are pending to occur in the simulation, there are two programming options: to schedule all of them or to schedule only the one which will first occur. This second option is recommended when RESTART is used because it reduces the number of events simultaneous scheduled in the simulation and thus the number of them that, according to Section 5.6, must be re-scheduled at the beginning of each retrial.

Note that f_O is equal to 1 when the efficiency is measured in terms of the number of simulated events.

5.5 Analysis of Factor f_V

According to left-hand formula (16), factor f_V is a weighted mean of s_i $\left(1 \le i \le M+1\right)$, all the weights being positive. Thus, the following bound of f_V can be defined:

$$f_V \le \underset{1 \le i \le M+1}{Max} \left(s_i\right).$$

As explained in Section 4.1, the term that may motivate a high value of s_i and thus of f_V is the variance of the importance of the system states at events B_i, $V\left(P^*_{A/X_i}\right)$ or more precisely, the ratio $V\left(P^*_{A/X_i}\right) \Big/ \left(P^*_{A/i}\right)^2$. This ratio does not only depends on the chosen importance function but also on the characteristics of the process $X(t)$. The optimal importance function Φ is that for which each threshold T_i of Φ defines an importance

I_i such that, for any system state x, $\Phi(x) \geq T_i \Leftrightarrow P^*_{A/x} \geq I_i$. This importance function usually leads to very low values of $V\left(P^*_{A/X_i}\right)\big/\left(P^*_{A/i}\right)^2$ and thus of s_i and f_V. Nevertheless, if the process $X(t)$ may skip from a state to another of much higher importance, the importance of the events B_i may be much higher than I_i. For this type of processes s_i may take a high value even when the optimal importance function is chosen.

In order have a more meaningful bound of f_V the following bound of s_i has been derived in [3]:

$$s_i \leq \frac{a_i\, Max(K'_i, \gamma_i)}{K_A} \frac{Q^*_{A/i}}{P^*_{A/i}} \quad (1 \leq i \leq M), \tag{20}$$

where $Q^*_{A/i}$ is the supreme (or, in general, an upper bound) of P^*_{A/X_i}.

It is not necessary that P^*_{A/X_i} be bounded for finding a bound of s_i. If $Q^*_{A/i}$ is not a bound of P^*_{A/X_i} but $\Pr\left\{P^*_{A/X_i} > P^*_{A/x_i}\right\}$ for $P^*_{A/x_i} > Q^*_{A/i}$ decreases faster than $1\big/\left(P^*_{A/x_i}\right)^{\beta_i}$ for some $\beta_i > 2$, that is, if

$$\Pr\left\{P^*_{A/X_i} > P^*_{A/x_i}\right\} \leq \frac{\Pr\left\{P^*_{A/X_i} > Q^*_{A/i}\right\}}{\left(P^*_{A/x_i}\big/Q^*_{A/i}\right)^{\beta_i}} \qquad \left(\beta_i > 2, \;\; \forall P^*_{A/x_i} > Q^*_{A/i}\right),$$

the following bound for s_i is derived in [3]:

$$s_i \leq \frac{a_i\, Max(K'_i, \gamma_i)}{K_A} \frac{Q^*_{A/i}}{P^*_{A/i}} \frac{\beta_i}{\beta_i - 2} \quad (1 \leq i \leq M). \tag{21}$$

Note that bound (20) is a particular case of bound (21) for β_i tending to infinity. Bound (21) is less restrictive because it does not require β_i tending to infinity (which implies a bounded P^*_{A/X_i}) but it only requires $\beta_i > 2$.

5.6 Guidelines to Reduce Factor f_V

As explained in Section 5.5, the value of factor f_V depends on the variance of the importance at events B_i. To reduce this variance, all the states x_i at events B_i must have similar importance; this is achieved by:

- Using a good importance function Φ, i.e., a function for which all the states on the threshold boundary $\Phi = T_i$ have similar importance.
- Reducing importance skipping, that is, avoiding that the process $X(t)$ may skip from a given state to another of much higher importance. Importance skipping may cause some events B_i to be far from the threshold boundary, thus having importance much higher than other events B_i on (or close to) the boundary.

We have treated in previous sections the way of optimizing RESTART for a given $X(t)$. However, for a given $Z(t)$, the process $X(t)$ may be different depending on how the simulation model is implemented. The extent of importance skipping is determined by the process $X(t)$, as explained below. Thus, a proper choice of the process $X(t)$ and the importance function Φ can reduce the factor f_V and hence increase the efficiency of RESTART. Let us see how to choose $X(t)$ and how to choose the importance function.

Guidelines for the Choice of $X(t)$. As explained in Section 4.1, to reduce the variance of the importance at events B_i for a given threshold i leads to reduce the correlation among trials made from a given B_i and vice versa. Although any one of these two reductions may be used as a criterion for selecting a proper process $X(t)$, both of them are used in this section to reinforce the reasoning.

As indicated in [3], when a simulation event occurs some random decisions may have to be taken, i.e., the values of some system variables (e.g., the number of packets in an arriving burst or the time scheduled for the occurrence of a future event) may have to be randomly determined. The definition of the process $X(t)$ depends on the way these random decisions are taken during the simulation, which determines the extent of importance skipping and correlations among trials $[B_i, D_i)$. Therefore, $X(t)$ also impacts the efficiency of RESTART, as shown in [38]. Let us illustrate this by using some examples.

Let us consider the two following options for determining the random number of packets in a burst that arrives at a queue: (a) to determine the entire length of the burst at the arrival of the first packet, and (b) to determine at the arrival of each packet whether it is the last packet or there are more packets in the burst. In option (b) $X(t)$ includes the number of packets in the burst arrived so far while in option (a) also includes the number of remaining (yet to arrive) packets in the burst.

Note that in option (a) only one random decision is made at the beginning of the burst, while in option (b) a number of sequential random decisions are made (conditioned on the number of packets in the burst arrived so far), one at the arrival of each packet in the burst. Clearly, the process $X(t)$ evolves at large increments in option (a), which may cause large importance skipping. On the other hand, in option (b) the process $X(t)$ evolves at small increments, which reduces importance skipping. Therefore, option (b) is recommended in a simulation in which RESTART is applied.

Also, note that $X(t)$ is Markovian in both options, since it contains sufficient information to execute the simulation of the system. However, in option (a) some future events are scheduled before they actually happen, while in option (b) no future events are scheduled unless necessary to continue the simulation. In the application of RESTART, option (a) will cause more sharing of future events (and hence more correlation) among trials made from a given B_i. This reinforces the reason given above for justifying why option (b) is favored over option (a) in the application of RESTART.

An alternative way of implementing option (b) is to determine the burst length at the arrival of the first packet (as in option (a)) and to determine it again at the arrival of each new packet by randomly generating the remaining burst length (conditioned on the number of packets arrived so far). This implementation of option (b) is equivalent to the previous one because it yields the same process $X(t)$.

Let us now consider the scheduling of future events in the simulation of a $G/G/1$ queue, where the rare set is defined as the queue length $q(t)$ greater than a certain threshold. If the times of occurrence of next arrival, t_{NA}, and of next service completion, t_{NC} , are scheduled only once at the previous arrival and at the start of the current service, respectively, then $X(t) = (q(t), t_{NA} -t, t_{NC} -t)$. This is similar to option (a) of the previous example: $X(t)$ includes the times of already scheduled future (arrival and service completion) events, which could cause importance skipping (e.g., when a high value is randomly assigned to $t_{NC} -t$). It also increases correlation among trials $[B_i, D_i)$ due to sharing of future events. This can be avoided by using the process $X(t) = (q(t), t- t_{PA}, t- t_{CS})$ to simulate the system, where t_{PA} and t_{CS} are the times of the previous arrival and the start of the current service, respectively. This is similar to option (b) in the previous example. Here, arrival and service completion events can be rescheduled (conditioned on the elapsed times, $t- t_{PA}$ and $t- t_{CS}$, respectively) at the occurrence of every event. However, to avoid unnecessary overhead, it is sufficient to reschedule only at events B_i, at the beginning of each retrial. This rescheduling minimizes the sharing of future events by different trials from the same event B_i and hence reduces the correlation among them, which improves the efficiency of RESTART.

Guidelines for the Choice of the Importance Function. Once the variables required to describe $X(t)$ have been determined, an importance function, which is a function of these variables, must be chosen. With a proper importance function all the states on each of the threshold boundaries $\Phi = T_i$ have similar importance, and thus, if importance skipping is small, all the states $x_i \in \Omega_i$ also have similar importance for any i. It leads to small values of $V\left(P^*_{A/X_i}\right)$ and thus also s_i for any i, and consequently to a small value of f_V.

In [20] it was pointed out that "the most challenging work for future research is to find and implement an efficient algorithm to determine good importance functions for defining thresholds".

In the case of one-dimensional systems, the choice of the importance function is straightforward, because the threshold boundary has only one state. Without importance skipping, this state is also the only state of Ω_i and thus $V\left(P^*_{A/X_i}\right) = 0$.

Also, for multidimensional systems, small values of s_i and thus of f_V are achieved if all the states $x_i \in \Omega_i$ have similar importance, but this condition is not strictly necessary. States $x_i \in \Omega_i$ with moderate probability of occurrence may have much lower importance than the most frequent ones without leading to high values of $V\left(P^*_{A/X_i}\right)$ and s_i : consider that P^*_{A/x_i} is bounded and that the ratio of the probability of Ω_i^H , the set of states $x_i \in \Omega_i$ with importance P^*_{A/x_i} close to its supreme $Q^*_{A/i}$, to the probability of the whole set Ω_i is appreciable (greater than, say, 0.2 or 0.3). Then $P^*_{A/i}$, the mean importance of states $x_i \in \Omega_i$, is close to the mean importance of states $x_i \in \Omega_i^H$ (and thus also close to the supreme $Q^*_{A/i}$). Consequently, the ratio $Q^*_{A/i} / P^*_{A/i}$ is small and, according to formula (30), s_i is small.

In [43] it was stated that "in the case of multidimensional state spaces, good choices of the importance function for splitting are crucial, and are definitely non-trivial to obtain in general". It is non-trivial because an exact analytical evaluation of the importance of the states is not possible in most cases. Thus a combination of approximate analytical formulas, heuristic reasoning and feedback from simulation results must be used to choose an appropriate importance function.

An approach that can be used in queuing networks is to assume that the importance function is a linear combination of the queue lengths of the network nodes: $\Phi = \sum_{\forall i} a_i Q_i$. Several procedures can be used to assign values to the coefficients a_i:

- First of all, one of the coefficients can be made equal to 1 without loss of generality.
- If the network has few nodes, e.g., only two nodes or three nodes and, thus, only one or two coefficients, the simplest solution is to perform pilot runs to test several values of the coefficients and to choose the values for which the application is more efficient. For saving computational time of the pilot runs, they can be made for system parameter values for which the rare event is not so rare, given that the results obtained usually apply to the parameter values of interest.
- By means of heuristic reasoning the number of coefficients that have to be adjusted may be reduced. E.g., the coefficients can be made equal to zero for the nodes for which its queue length has not impact on the occurrence of the rare event or it is guessed that the impact is small. Another example is to assign the same value to coefficients corresponding to queue lengths with the same or similar impact on the occurrence of the rare event.
- If possible, approximate analytical formulas may be derived to roughly estimate the importance of some states. From these formulas the coefficient values may be evaluated by equating the importance function of states with the same importance. An alternative to this approach is to estimate the importance of some states by means of pilot runs.
- If values have been assigned to some of the coefficients, interpolation or extrapolation based on heuristic reasoning may be used to assign values to the remaining ones.
- All the assumptions or approximations made for assigning values can be checked by means of pilot runs (that usually can be made for system parameter values for which the rare event is not so rare). These pilot runs can also be used to introduce correction factors to a set of coefficients, previously obtained, to improve them. This approach may also be used when the set of parameters obtained for a model are going to be applied to another similar model.

Observe that the approximations are allowed for deriving the importance function because they could affect the efficiency of the method, but they do not affect the correctness of the estimates.

In the networks with few nodes the number of coefficients to be assigned is smaller but the values assigned to them could be more critical due to the strong dependency that usually exists among the nodes. However in more complex networks, though the number of coefficients is larger the accuracy of the values assigned to them is not so critical, as it is shown in the examples of Section 6. This fact may compensate the difficulty that complex networks could have due to the need of assigning many coefficient values. As we will see in Section 6, the two-queue Jackson tandem network was simulated in [41] defining the rare set as $q_2 \geq L$ and choosing the importance function $\Phi = q_2$. This importance function Φ led to a large value of f_V and, as reported in [41], to a very low efficiency. However in the multistage ATM switch studied in [16] an equivalent importance function led to a high efficiency. This switch has three stages of 8×8 switching elements (SE), eight of them in each stage. Each SE is an output buffered switch with eight separated buffers of size *K*. The rare set is defined as the overflow of a buffer of the third stage. The importance function Φ is defined as the queue length q of the buffer under study. An importance function equivalent to that used in the previous example was successful, despite the greater complexity of the system. This is because the cells in the buffers in the second stage do not need to go to the buffer under study in the third stage, but can go to any of the 64 buffers of this stage instead. As a result, the queue lengths of the buffers of the first or second stage have a small impact on the future queue length of a buffer of the third stage. Although the importance function used in [16] could be improved taking into account the queue length of the other queues (as will be seen in Section 6) the dependencies in this complex system are weak enough to be ignored without a significant impact on the efficiency achieved. In the two-queue tandem network, its simplicity notwithstanding, the dependence is strong and cannot be ignored.

In reliability problems, an importance function defined as a linear combination of variables representing the state of each component (1= failure, 0= operational) is not appropriate. The effect of the failure of a component is different depending which other components have also failed and this type of dependencies cannot be taken into account with a linear function of the states of the components. It is better in this case to obtain, based on some heuristic reasoning, a formula of the importance function that take into account these dependencies. In Section 6.3, a function of the states of the components obtained heuristically is proposed as importance function. Although the proposed importance function works well in all the cases studied, there are some cases in which it could be improved because, as indicated in that section, the proposed function does not account for all the features of the system state that may impact on the occurrence of the rare event. A possibility to improve it could be to obtain heuristically another function of the system state variables that accounts for those features of the system state that are not taken into account by the previous function. The final importance function could be a linear combination of the two functions. Thus the approach proposed for queuing networks consisting in the choice of an importance function built as a linear combination of variables of the system state could be generalized to the choice of a linear combination of functions of the system state.

6 Application Examples

Several examples on Jackson and non-Jackson queuing networks and on ultra reliable systems are shown in this section to illustrate the application of the guidelines given in Section 5 and the efficiency obtained. For evaluating the goodness of an application and its possibility of improvement, it is not only interesting to observe the required computational time but also the gain obtained and the values of the inefficiency factors. Section 6.1 explains how we can estimate the gain and the factors f_V and f_O. Factors f_R and f_T may be estimated by its bounds given by formulas (18) and (19) respectively.

In all the runs, the simulation length was adjusted to have a relative half width of the 95% confidence interval (relative error) equal to 10%. The interval width was evaluated using the batch means method. The experiments of the two-queue Jackson tamdem network were run on a Sun Ultra 5 workstation and the remainig ones on a Pentium(R) D CPU 3.01 GHz.

6.1 Jackson Networks

First we will see how to obtain the importance function for two-queue tandem networks by assigning the coefficient values of the linear combination of queue lengths by means of tests made with pilot runs. Then general Jackson networks are studied. As the method of assigning by means of pilot runs is not practicable when the number of nodes is large, the importance function is derived by means of approximate analytical formulas.

Two-Queue Jackson Tandem Network. In this network customers with Poisson arrival enter the first queue and, after being served, enter the second one. The mean arrival rate is λ and the service time is exponentially distributed in each queue with mean service rates μ_1 and μ_2, respectively. The load at each queue is $\rho_i = \lambda / \mu_i \, (i = 1,2)$. The buffer space at each queue is assumed to be infinite. The system state *Z*(*t*) is given by (q_1, q_2), where q_i is the number of customers at queue *i*. If rescheduling is made, the system state *X*(*t*) is also given by (q_1, q_2). This model has received considerable attention in the rare event literature, e.g., [14], [17], [19], [22], [39], [40], [41] and [42].

The difficulty of applying accelerated simulation techniques arises when the first queue is the bottleneck and the rare set definition is related to the value of q_2. In order to cope with a difficult case the loads tested were $\rho_1 = 0.5$ and $\rho_2 = 0.33$.

The network was studied in [3] for the following three definitions of the rare set *A*: $Q_1 + Q_2 \geq L$; $Q_2 \geq L$ and $\min(Q_1, Q_2) \geq L$.

In these examples thresholds and number of retrials were determined in a similar manner to that explained later for general Jackson networks.

Rare Set Defined as $Q_1 + Q_2 \geq L$. For this definition of the rare set, the most "natural" importance function is $\Phi = Q_1 + Q_2$. Let us analyze if this function is appropriate. Assume that $L = 60$ and $T_i = 30$. The possible states at an event B_i are (0,30), (1,29), (2,28), ..., (29,1), (30,0). The importance of each of these states is different. The higher the value of Q_1 (for $Q_1 + Q_2 = 30$), the higher the importance of the state, given that a customer at Q_1 has to be served by both servers before leaving the system, while a customer at Q_2 has to be served only at the second one. Thus the supreme $Q^*_{A/i}$ is the importance of state (30,0). But, given that the first queue is the bottleneck, the states with high value of Q_1 and low value of Q_2 have the highest probability. As these states have an importance close to the supreme, $\Pr\left\{\Omega_i^H \middle| \Omega_i\right\}$ is high. Thus $\Phi = Q_1 + Q_2$ seems to lead to moderate values of s_i and therefore, also f_V. Simulation results confirm that this qualitative reasoning is valid: probabilities up to 10^{-66} were accurately estimated with less than 40 minutes of computational time. The very low values of f_V (smaller than 1.02) show that the choice of $\Phi = Q_1 + Q_2$ is appropriate and that the application is very close to the optimal one.

Let us see how to estimate the gain obtained and the values of factors f_V and f_O. The gain in events or the gain in time with respect to a crude simulation is estimated as follows: in a crude simulation with $L = 14$, thus $P = 1.22 \times 10^{-4}$, and the same remaining conditions, the number of reference events (arrivals in this case) and the computational time are measured. As $V\left(\hat{P}\right) = K_A\, P/N$ the measured values are extrapolated for the value of L for which we want to estimate the factors, e.g., 220, under the assumption of K_A taking the same value. The gain is the ratio between these extrapolated values and those measured in the simulation with RESTART. A gain in events equal to 3.4•10^{61} and a gain in events equal to 5.0•10^{60} are obtained. Then we compare the measured gain with the theoretical one derived from formula (15). If we assume f_V = 1 and f_O = 1 in (15), we obtain for L = 220 a gain equal to P =3.46•10^{61} (taking and r_i given by formulas (13) and (12) respectively and thus taking f_R = 1 and f_T equal to its bound (19) evaluated for $P_{min} = P^{1/(L-1)}$). We see that the theoretical gain (for f_V = 1, f_O = 1) is 1.02 times the actual gain in events. Given that the gain in events is not affected by the factor f_O, the value 1.02 can be taken as an estimate of f_V for $L = 220$. Finally, the factor f_O can be estimated as the ratio between the gain in events and the gain in time. It leads to f sub O = 6.8.

Rare Set Defined as $Q_2 \geq L$. The simplicity of the system allows for simulating it by means of regenerative simulations and splitting, as in [40] and [41]. In [41] the chosen importance function is $\Phi = Q_2$. Let us consider that L and an intermediate threshold T_i take the values $L = 30$ and $T_i = 15$. At an event B_i, $q_2 = 15$ but Q_1 can take any value. It is clear that the probability of reaching A ($Q_2 \geq 30$) from an event B_i at which $Q_1 = 0$, $Q_2 = 15$ is very different from that of reaching A from an event B_i with, say, $Q_1 = 60$, $Q_2 = 15$. The supreme $Q_{A/i}^*$ is given by the limit of the importance of state (Q_1, 15) when Q_1 tends to infinity. The probability of states (Q_1, 15) with high value of Q_1 is much smaller than that of states with small value of Q_1, and thus $\Pr\left\{\Omega_i^H \middle| \Omega_i\right\}$ is very small. Therefore this chosen function Φ leads to a large value of s_i and, as reported in [41], to a low efficiency.

It is clear that in the definition of Φ, Q_1 must be accounted for, since its value can affect the future evolution of Q_2. Some weight, albeit smaller than the weight given to Q_2, must be given to Q_1. Along this line of reasoning, we tested $\Phi = aQ_1 + Q_2$, with $0 \leq a \leq 1$. Pilot simulations for $L = 20$ with several values of a were run to determine the appropriate value of the coefficient a. The required computational times for a = 0.8, 0.7, 0.6, 0.5 and 0.4 were 48, 23, 15, 27 and 70 seconds, respectively. The value a = 0.6 is chosen, as it provides the best results.

Probabilities of the order of 10^{-29} and of 10^{-67} were accurately estimated with 7 and 200 minutes of computational time, respectively. Although the values of f_V (3.1 and 11.2 respectively) are not so small as in the previous case, they are moderate enough to accurately estimate very low probabilities at a reasonable computational time. These low values of f_V indicate that the application is close to the optimal one.

Rare Set Defined as $Min\left(Q_1, Q_2\right) \geq L$. For this case, the function Φ was defined in [40] as $\Phi = Min\left(Q_1, Q_2\right)$. This definition of Φ leads to high values of f_V and, as reported in that paper, low efficiency. For $T_i = 20$ (with L > 20), possible states at B_i are (100, 20), (20, 20) and (20, 100). The importance of, say, states (100, 20) or (20, 100) is much higher than that of state (20, 20). The supreme $Q^*_{A/i}$ is given by the limit of the importance of either state (20+j, 20) or state (20, 20+j) when j tends to infinity. The states with highest probability are (20+j, 20) or (20, 20+j) for low values of j. Consequently $\Pr\left\{\Omega_i^H \middle| \Omega_i\right\}$ is very low. Thus this definition of Φ does not seem to be appropriate.

For $Q_1 \leq L$ and $Q_2 \leq L$, we proposed to define Φ as a linear function of Q_1 and Q_2: $\Phi = aQ_1 + Q_2$. As the relative importance of the states depends on the system parameters, the appropriate value of the coefficient a may be greater, equal or smaller than 1 depending on the load values. Extending this definition for $Q_1 > L$ or $Q_2 > L$ does not appear to be appropriate: as the rare set is defined as $Q_1 > L$ and $Q_2 > L$, when Q_1 or Q_2 exceeds L we must give a lower weight to this excess. This effect may be taken into account by introducing a coefficient $b < 1$ in the definition of the importance function:

$$\Phi = a\Phi_1 + \Phi_2, \quad \text{where} \quad \Phi_i = \begin{cases} Q_i & \text{if} \quad q_i \leq L \\ L + b(Q_i - L) & \text{if} \quad q_i > L \end{cases}. \tag{22}$$

We tested the importance function (22) using pilot runs with a = 1 and different values of b. The best results were obtained for b = 0.6. Probabilities of the order of 10^{-32} and of 10^{-63} were accurately estimated with 7 and 72 minutes of computational time, respectively.

The low values of f_V (3.5 and 6.7, respectively) show that the application is very efficient and close to the optimal one. As the results were good enough, we did not investigated the improvement that could be obtained with other values of a.

This case was also studied in [19] by means of RESTART. They used the importance function $\Phi = Q_1 + 1.5Q_2$. It led to values of f_V (estimated by us based on their reported results) between 45 and 62 times the values reported in [3]. These results may be due to the fact that, in contrast to our approach, a lower weight was not given to Q_i-L when Q_i exceeds L.

In [40] the values of f_V are much higher. From their reported results, we have estimated a value of f_V =1600 for L = 10. It is difficult to estimate f_V for larger values of L

because, as they claimed, the failure of their approach resulted in the underestimation of P and its relative error. Anyway this value of f_V is very huge and the tendency observed indicates that they must be much higher for higher values of L.

General Jackson Networks. Formulas for obtaining effective importance functions were provided for two-stage Markovian networks with any number of nodes in each stage in [25] and extended to general Jackson networks with any number of nodes in [32]. Jobs arrivals and departures are allowed in all the nodes. After being served in node *l*, jobs can go to any node *m* with probability p_{lm} or they can leave the network with probability p_{l0}. The steady-state probability of the number of jobs exceeding a level at a target node, $Q_{tg} \geq L$, was estimated.

Some approximations and assumptions were needed to derive the formulas of the importance function. First, it was assumed that the importance function is a linear function of the queue length of the nodes placed at a distance from the target node smaller than 3. Thus the queue length of a node was considered in the importance function only if the jobs leaving the node go directly to the target node (distance 1) or through only one intermediate node (distance 2). Then it was evaluated the importance of the extreme (also called boundary) states when the process enters sets C_i $\forall i$, that is, the system states at which only one queue is not empty. Finally, for calculating the coefficients of Q_i for each i in the importance function it was equated the importance of the extreme state corresponding to the target queue with the importance of each of the other extreme states. The goodness of the importance functions derived in the paper with such approximations was supported by the efficiency achieved in the simulations

The formula obtained for the two-queue Jackson tandem network with $\rho_1 > \rho_2$ and the rare set defined as $q_2 \geq L$ was: $\Phi = \frac{\ln \rho_1}{\ln \rho_2} Q_1 + Q_2$. For the loads above considered for this example the importance function given by the formula is: $\Phi = 0.63Q_1 + Q_2$. Slightly better results were obtained with this formula than with the importance function $\Phi = 0.6Q_1 + Q_2$ obtained heuristically.

Let us denote K the number of nodes with distance 1 and H the number of nodes with distance 2, for any value of K and H. Jobs with independent Poisson arrivals enter each node from the outside with arrival rates $\gamma_{1i}, i = 1, \ldots, H$ to the nodes with distance 2, $\gamma_{2j}, j = 1, \ldots, K$ to the nodes with distance 1 and γ_{tg} to the target node. The total arrival rates to each node (arrivals from the outside + arrivals from the other nodes) are denoted by: $\lambda_{1i}, i = 1, \ldots, H$, $\lambda_{2j}, j = 1, \ldots, K$ and λ_{tg}, respectively. The service times of all the nodes are assumed to be exponentially distributed with service rates $\mu_{1i}, i = 1, \ldots, H$, $\mu_{2j}, j = 1, \ldots, K$ and μ_{tg}, respectively. The buffer space in each queue is assumed to be infinite. Let us observe that When there are not nodes at a distance greater then 2, $\lambda_{1i} = \gamma_{1i} + \sum_{l=1}^{H} \lambda_{1l} p_{li} + \sum_{j=1}^{K} \lambda_{2j} p_{ji} + \lambda_{tg} p_{tgi}, \; i = 1, \ldots, H$. Analogous equations are obtained for $\lambda_{2j}, j=1,\ldots,K$ and λ_{tg}. The loads of the nodes are $\rho_{1i}=\lambda_{1i}/\mu_{1i}, i=1,\ldots,H$, $\rho_{2j}=\lambda_{2j}/\mu_{2j}, j=1,\ldots,K$ and $\rho_{tg}=\lambda_{tg}/\mu_{tg}$, respectively.

A general formula of the importance function valid for any Jackson network was derived in [32]. A simplified version of this formula (also given in that paper) that matches with the general one in almost all cases is the following:

$$\Phi=\sum_{i=1}^{H} Min\left\{1,\alpha_{1i}\frac{\ln\left(\rho_{tg}/\rho_{tgi}^{*}\right)}{\ln\rho_{tg}}\right\}Q_{1i}+\sum_{j=1}^{K} Min\left\{1,\alpha_{2j}\frac{\ln\left(\rho_{tg}/\rho_{tgj}^{\perp}\right)}{\ln\rho_{tg}}\right\}Q_{2j}+Q_{tg}, \qquad (23)$$

where:

$$\rho_{tg}=\frac{\gamma_{tg}+\sum_{j=1}^{K}\lambda_{2j}p_{jtg}+\lambda_{tg}p_{tgtg}}{\mu_{tg}}=\frac{\lambda_{tg}}{\mu_{tg}},$$

$$\rho_{tgi}^{*}=\frac{\gamma_{tg}+\sum_{j=1}^{K}Min\left\{\lambda_{2j}+\left(\mu_{1i}-\lambda_{1i}\right)p_{ij},\mu_{2j}\right\}p_{jtg}+\lambda_{tg}p_{tgtg}}{\mu_{tg}},$$

$$\rho_{tgi}^{\perp}=\frac{\gamma_{tg}+\mu_{2j}p_{jtg}+\sum_{l\neq j}\lambda_{2l}p_{ltg}+\lambda_{tg}p_{tgtg}}{\mu_{tg}},$$

$$\alpha_{1i}=1+\frac{\sum_{l\neq i}\gamma_{1l}\sum_{j=1}^{K}p_{lj}p_{jtg}+\sum_{j=1}^{K}\gamma_{2j}p_{jtg}+\gamma_{tg}}{\mu_{1i}\sum_{j=1}^{K}p_{ij}p_{jtg}},$$

$$\alpha_{2j}=1+\frac{\sum_{i=1}^{H}\gamma_{1i}\sum_{l\neq j}p_{il}p_{ltg}+\sum_{l\neq j}\gamma_{2l}p_{ltg}+\gamma_{tg}}{\mu_{2j}p_{jtg}}.$$

ρ_{tgi}^{*} and $\rho_{tgj}^{\perp}$ are, approximately, the loads of the target queue when a node *i* at distance 2 from the target node or a node *j* at distance 1, respectively, are not empty. It is more difficult to get insight of the meaning of α_{1i} and α_{2j} without following the derivation of formula (23). Nevertheless, the formulas are easy to apply because all their terms are parameters of the system.

Test Cases. Several simulation experiments on Jackson networks with different topologies and loads were conducted in [32]. The rare set *A* was defined in most cases as $Q_{tg} \geq 70$, where Q_{tg} is the number of customers at the target node. The steady state probability of A was of the order of 10^{-34} in those examples. The reason for simulating such small probabilities is to show the goodness of the importance functions obtained in the paper, given that if it is possible to estimate accurately such small probabilities with short or moderate computational time, it will take much less time to estimate more realistic probabilities.

Thresholds T_i were set for every integer value of Φ between 2 (in some cases 3) and a number varying between 71 and 75 depending on the case being simulated. Observe that, as $L=70$, the rare set A is not included in C_M given that $A \cap (C_i - C_{i+1}) \neq \phi$ if $T_i \geq 70$. Pilot runs (one or two for each case) were made to set the number of retrials. We proceeded as following: we set (for example) the thresholds 2, 3, 4, … , 74 and we made a pilot simulation. This simulation derived the optimal number of retrials according to formula (12) following the guidelines given in Section 5.2 for rounding to integer values. If the derived value of the number threshold (in the pilot simulation) of retrials from a threshold was 1 was 1, such threshold was eliminated. If the number of retrials from the last threshold was greater than 5, an additional threshold was set. The number of retrials R_i finally was 2 or 3 in all cases.

Although it is not possible to simulate all the Jackson networks to prove that the importance function given by formula (23) is always effective, test cases that a priori could have some difficulties were selected in [32]. If the importance function is effective for these cases, it is supposed that it will be also effective for most Jackson networks. The systems simulated were the following:

- a two-queue Jackson tandem network;
- a three-queue Jackson tandem network;
- a three-stage network with 4 nodes in the first and second stage and 1 or 2 nodes in the third stage;
- a Jackson network with 7 nodes with 2 nodes at distance 1 from the target node, and 4 nodes at distance 2 from the target node.
- a Jackson network with 7 nodes with 2 nodes at distance 1 but with 2 nodes at distance 3 from the target node, 2 nodes at the target node. distance 2 and 2 nodes at distance 3.
- a large Jackson network with 15 nodes: 4 of them at distance 3 from the target node, 5 at distance 2 and 5 at distance 1.
- a 2-node Jackson network with strong feedback: jobs departing any of the two nodes join the other node with a probability of 0.8.
- a six-queue Jackson tandem network, in which the first 5 nodes have the same load (2/3) and the last (target) node has a lower load (1/3). This case and the two first ones are networks for which the dependency of the target queues on the queue length of the other queues is very high because all the customers of the other queues have to go to the target queue.

In all the networks, except the last one, probabilities of the order of 10^{-34} were estimated with short or moderate computational time with the importance function given by formula (23). For the six-queue Jackson tandem network thirty minutes of computational time was needed to estimate a probability of the order of 10^{-15} with that importance function (that does not take into account the queue length of the first 3 nodes). The importance function was improved heuristically giving the weights provided by the formula for the last 3 nodes and lower extrapolated weights to the nodes that are farther from the target node. With this importance function, which accounts for the dependence of the target node on all the nodes, an accurate estimation of the same probability was obtained with 10 minutes of computational time.

The results obtained in [32] show that the worst cases are networks with very high dependencies, in which the target queue has a much lower load than the other queues, and that the best cases are usually the most complex networks with a high number of nodes because there are usually weak dependencies in these cases. Although the importance function given by formula (23) can be improved for some specific networks, it seems to be good enough for estimating very low probabilities with short or moderate computational times for most (perhaps all) Jackson networks.

In order to illustrate the results of the simulations made in [32], those corresponding to a Jackson network with 7 nodes (with 2 nodes at distance 3 from the target node) will be reproduced here.

Jobs from the outside arrive at any node of the network at a rate $\gamma_i = 1$, $i = 1,\ldots,7$. After being served in each node, a job leaves the network with probability 0.2. Otherwise the job goes to another node in accordance to the following transition matrix:

	1	2	3	4	5	6	tg
1	0.2	0.2	0.2	0.2	0	0	0
2	0.2	0.2	0.2	0.2	0	0	0
3	0.1	0.1	0.1	0.1	0.2	0.2	0
4	0.1	0.1	0.1	0.1	0.2	0.2	0
5	0.1	0.1	0.1	0.1	0	0.1	0.3
6	0.1	0.1	0.1	0.1	0.1	0	0.3
tg	0.1	0.1	0.1	0.1	0.1	0.1	0.2

Let observe that jobs that leave nodes 1 or 2 have to visit nodes 3 or 4 and nodes 5 or 6 before entering the target node. We wished to check whether to ignore the impact of the queue lengths of the nodes at a distance greater than two on the queue length of the target node is a reasonable approximation. The results are summarized in Table 1.

For each group of loads of the nodes, three importance functions were used. The first one is given by formula (23), that is, without considering the queue lengths of nodes 1 and 2 placed at a distance 3 (a = 0 in the table). The second one also takes into account nodes 1 and 2. The values of their coefficients (called a in the table) were derived with the same methodology used to derive formula (23). The third importance function only takes into account the target node and the nodes that are at distance 1 from it ($a = b = 0$).

The best results were obtained with the second importance function, the function that takes into account the number of jobs in the seven nodes. However, the importance function given by formula (23) leads to very effective results: probabilities of the order of 10^{-34} were obtained with less than 10 minutes of computational time and the values of factor f_V, though slightly greater than those obtained with the other importance function, are very small. Consequently, it does not seem that it worth further complicating the importance function by taking into account the nodes that are at distance greater than 2 from the target node. Nevertheless, though it is not necessary in almost all the cases, if we want to improve the importance function taking into account more queue lengths in the function, it can be done deriving the coefficients of those nodes with the same methodology used to derive formula (23) or heuristically giving the weights provided by the formula to the nodes at distance lower than 3 and lower weights to the nodes that are farther from the target node. The results are much worse when using the third importance function, which only accounts for the target node and the nodes that are at distance 1 from the target node.

Table 1. Results for Jackson networks. Relative error = 0.1. Rare set probability: $P(Q_{tg} \geq 70) = 3.13 \cdot 10^{-34}$. $\rho_{tg} = 0.3322$. $\Phi = a\sum_{i=1}^{2} Q_i + b\sum_{j=3}^{4} Q_j + c\sum_{k=5}^{6} Q_k + Q_{tg}$.

$\hat{P}$	ρ_i	ρ_j	ρ_k	a	b	c	Time (minutes)	f_V
$3.1\cdot10^{-34}$	0.62	0.51	0.41	0	0.31	0.47	9.6	3.6
$3.2\cdot10^{-34}$	0.62	0.51	0.41	0.07	0.31	0.47	7.5	2.7
$3.3\cdot10^{-34}$	0.62	0.51	0.41	0	0	0.47	518	204
$3.1\cdot10^{-34}$	0.47	0.51	0.41	0	0.31	0.47	5.2	1.7
$3.0\cdot10^{-34}$	0.47	0.51	0.41	0.09	0.31	0.47	4.2	1.4
$2.9\cdot10^{-34}$	0.47	0.51	0.41	0	0	0.47	207	55
$3.3\cdot10^{-34}$	0.31	0.31	0.31	0	0.51	0.61	3.3	1.1
$3.1\cdot10^{-34}$	0.31	0.31	0.31	0.15	0.51	0.61	3.0	1.0
$3.0\cdot10^{-34}$	0.31	0.31	0.31	0	0	0.61	6.1	2.4

6.2 Non-Jackson Networks

The importance function given by formula (23) has been derived equating the importance of one extreme state with the importance of each of the other extreme states. It is interesting to see whether the importance of the extreme states are affected in a similar manner when the interarrivals and/or services times are not exponentially distributed and, as a consequence, whether the importance function derived for Jackson networks fits for other networks.

In [44] networks with Poisson arrivals and Erlang service times were studied. Two of the networks above mentioned were analyzed: the three-queue tandem network and the first one of the two networks with seven nodes. In the first model, the service time at each node follows an Erlang distribution with shape parameter equal to α (2 or 3). Initially, the chosen importance function was $\Phi = aQ_1 + bQ_2 + Q_3$ evaluated according to formula (23). Then, the coefficients a and b of Φ were multiplied by a correction factor k, the same for both coefficients. The tested values of k were 0.6, 0.7, 0.8 and 0.9.

We observed that the best value of k was between 0.6 and 0.9, depending on the case. We also observed that for $\alpha = 2$, the value of k is closer to 1 than for $\alpha = 3$.

As the coefficient of variation of Pearson of the Erlang distribution is $1/\sqrt{\alpha}$, it seems that the more similar is this coefficient to that of the exponential distribution, the closer is the importance function to that given by the formula. It is also observed that lower values of the loads of the two first nodes lead to importance functions closer to those derived for the exponential distribution. Probabilities of the order of 10^{-15} were estimated with computational times between 0.4 and 21.8 minutes.

The values of factor f_O are much greater than those obtained in [32] for the same network but with exponential service times. The reason is that rescheduling is straightforward only for the exponential distribution due to the memoryless property of this model. Rescheduling service times of any other distribution is more time consuming. We can proceed as follows: a random value of the whole service time of a job is obtained. If that value is greater than the service time at the current time, the remaining service time of the job is obtained as the difference between the two amounts. Otherwise a new random value is obtained and so on. This procedure has the problem that the number of iterations is very huge when the actual service time is much larger than the mean service time, and it greatly increases factor f_O .As rescheduling is made to improve factor f_v but it is not strictly necessary, if after a fix number of trials, e.g., 50, the random value of the whole service time is always lower than the service time at the current time, the service time is not rescheduled. In this way factor f_O is significantly reduced and, as only 1 or 2% of the scheduled times are not rescheduled, the impact on f_V is negligible. Nevertheless, even with this improvement of the procedure, the value of factor f_O with Erlang service times is around four times greater than with exponential times for estimating a probability of the order of 10^{-15}.

Low or at least moderate values of factor f_V were obtained in all the cases. It shows that the application is not far from the optimal, at least for the tested cases. We observe that the worst results (greatest computational times and greatest values of factor f_V) were obtained when $\rho_3 < \rho_2 < \rho_1$, but even in these cases the computational times are moderate (21.8 minutes). The importance functions given by formula (23) lead to greater computational times, although these times are also moderate in all the cases, except in the case $\rho_3 < \rho_2 < \rho_1$, in which the computational time was greater than one day.

The second network studied in [44] is a network with 7 nodes with 2 nodes at distance 1 from the target node, and 4 nodes at distance 2, with Poison arrivals and Erlang service times.

The computational times needed for estimating probabilities of the same order of magnitude (10^{-15}) is much lower than in the previous network. The results are better in this network due to the weaker dependence between the queue lengths.

In the three cases of $\alpha = 2$ the best results were obtained with the importance function given by formula (23), while for $\alpha = 3$ the best results were obtained with coefficients of nodes at distance 1 and 2 around 10% lower than those given by formula (23), that is, with a correction factor $k = 0.9$. Nevertheless, very good results were also obtained without any correction factor. The very low values of f_V achieved (less than 1.7 in the six cases studied) show that the application is very close to the optimal one, at least for the tested cases.

In [34] the simulation study made in [44] was extended in a twofold direction. On the one hand it was also simulated two additional above mentioned networks: the large network with 15 nodes and the network with 2 nodes and very strong feedback, On the other hand we used hyperexponential and Erlang distribution for modelling the interarrival and/or service times.

In this paper a better method for improving formula (23) (formula derived for Jackson networks) is applied for non-Markovian networks: instead of multiplying some coefficients by a correction factor obtained heuristically, we also use formula (23) but the actual loads used in the formulas are substituted by "effective loads", defined as the loads ρ^e such that $\Pr\{Q \geq n\} = \left(\rho^e\right)^n$. For Jackson networks for a certain value of n. the "effective load" matches the actual load.

For the three-queue tandem network and for the network with 2 nodes and strong feedback, the efficiency obtained is much higher using effective loads than using actual loads. However similar efficiency is obtained using "effective loads" and actual loads (that is, using formula (23) without any correction) with the more complex networks of 7 and 15 nodes because the effective loads are similar to the actual loads in these networks. Overflow probabilities lower than those needed in practical problems (around 10^{-15}) were accurately estimated within short computational work. The worst results were obtained when the dependence of the target queue on the length of the other queues is very high (as occurs in a tandem network) and the load of the target queue is much lower than the others. For the worst case, less than 17 minutes of computational times were needed for estimating a probability of the order of 10^{-15} . In some of the cases the probability was estimated in less than one minute.

6.3 Ultra Reliable Systems

This section provides a simple importance function that can be useful for RESTART simulation of models of many highly dependable systems. Some examples from the literature illustrate the application of this importance function.

We consider generalized Machine Repairman Models. These models consist of multiple types of components with any number of components of each type, where each component can be in one of the following states: operational, failed, spare or dormant. An operational component becomes dormant if its operation depends upon the operation of other components and those components fail. General lifetime distributions and different failure rates can be specified for the operational, spare and dormant states. Dependencies among components and failure propagation (e.g., the failure of a component causes some other components to fail with given probabilities) are allowed. There is a set of repair services which repair failed components according to a general distribution and to some service discipline. The system is operational if certain combinations of components are operational. The concern is estimation of transient measures, such as system unreliability or unavailability at a given instant, and steady-state measures, such as steady-state unavailability and mean time between failures.

In a general system there are minimal cut sets with different cardinality. In a balanced system, where all the components have the same probability to fail, it is more probable that a system failure is due to the failure of all the components of a minimal cut set with the lowest cardinality. The "distance" to the system failure is related with the number of components that remain operational in the cut set with lowest the number of operational components. For this reason the importance function (at an instant *t*) is defined as:

$$\Phi(t) = cl - oc(t) \ , \tag{24}$$

where *cl* is the cardinality of the minimal cut set with the lowest cardinality and *oc*(*t*) is the number of components that are operational at time *t* in the cut set with the lowest number of operational components. Thresholds T_i of Φ can be defined at 1, 2, … , $cl-1$. For example, consider the network in Fig.2 that contains 8 links and 7 nodes. The system operates as long as there exists a path along operating links between node A and node B.

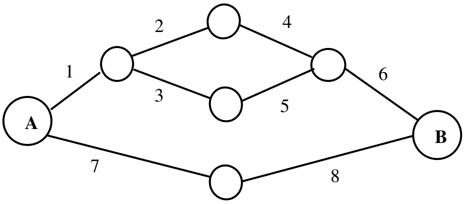


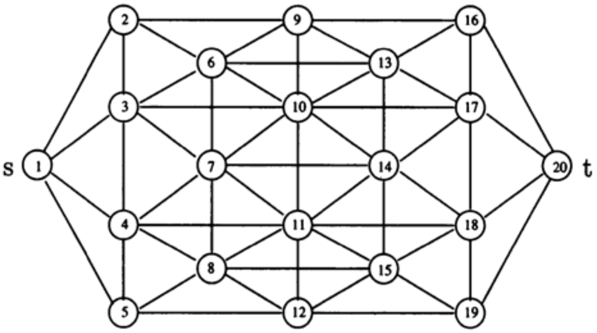
Fig. 2. Network with low redundancies

There are 4 minimal cut sets with 2 links: $(1,7),(1,8),(6,7),(6,8)$, and 8 minimal cut sets with 3 links: $(2,3,7),\ldots,(3,4,8)$. In this network $cl=2$ and we can define one threshold. The process is in the region C_1 (that is, $\Phi(t)\geq T_1$) if at least one component of any of the 4 cut sets with cardinality 2 or at least 2 components of any of the 8 cut sets with cardinality 3 are failed. As only one threshold can be defined, factor f_T would be high if the 8 links would be highly reliable.

The same importance function can be used for many unbalanced systems. The larger the difference among failure rates of the components, the greater the value of factor f_V. If the system is so unbalanced that factor f_V takes a very great value and, as a consequence, it is unfeasible to estimate the probability of interest within a reasonable computational effort, the importance function must be improved.

A limitation of the RESTART methodology for simulating highly-reliable systems is the difficulty to define thresholds close enough so that the probability of reaching the next threshold is reasonably large and, thus, close to the optimal. For this reason, L'ecuyer et al. [43] pointed out that this methodology is not appropriate for this type of systems and Xiao et al. [45] suggested that "importance splitting is hard to be adopted for dependability estimation of non-Markov systems, because thresholds function is hard to be presented under this situation". However, as it will be shown in the examples, probabilities up to the order of 10^{-16} can be accurately estimated within a reasonable computational effort.

We will describe three examples, two of them taken from [27] and the other one from [33]. Example 1, taken from [27] is the network of Fig.3 originally presented in [46], where it was simulated using importance sampling. The network contains 56 links, classified in 3 types, and a total of 107 components. Each type A link contains three identical components and fails when two components fail. Type B links contain one component. Each type C link contains two identical components and fails when one component fails. The mean lifetime is different for each type of component. The system operates as long as there exists a path along operating links between node 1 and node 20. There are 5 repair-persons, and repairs make components as good as new. Upon completing a repair, a repair-person selects the next component to repair randomly over the failed components in the network.



Type A links: (1,2), (1,3), (1,4), (1,5), (16,20), (17,20), (18,20), and (19,20)
Type B links: (2,3), (3,4), (4,5), (6,7), (7,8), (9,10), (10,11), (11,12), (13,14), (14,15), (16,17), (17,18), and (18,19).
All other links are type C.

Fig. 3. Network with redundancies

The system unreliability was estimated for different small values of intervals (0, t_e). Simulations were made assuming first a Markovian model, that is assuming that component lifetimes and repair times are exponentially distributed, and second assuming that component lifetime distributions are Raleigh (Weibull distribution with shape parameter equal to 2) and that repair time distributions are Erlang with shape parameter equal to 3. The minimal cut sets are defined on the links (not on the components). The importance function was given by formula (24). As $cl = 4$ three intermediate thresholds could be defined.

Probabilities up to the order of 10^{-11} could be accurately estimated within short or moderate computational time. Nevertheless a high value of factor f_V was observed because, for a given value of i, the states at events B_i with more importance have smaller probability to occur, see Section 5.6. The system states at events B_i with more importance are those in which operational links have greater probability to fail. It is more unlikely that the operational links of a minimal cut set are those with greater probability to fail because it requires a previous failure of the other links of the same minimal cut set, which have lower probability to fail. As commented above, for unbalanced systems the factor f_V can take high values.

The models with exponential lifetimes and service times were exactly the same models simulated in [46] with importance sampling, and the estimates of the steady-state unavailability are very close in both cases. For simulating the Weibull-Erlang models with RESTART the same procedure as for the Markovian models could be used given that the same importance function is valid in both cases. With importance sampling only results for the Markovian case have been obtained, because the analytical study required for applying importance sampling to non-Markovian systems with significant redundancies is very complicated.

Example 2 of [27] is a computing system originally presented in [47], where it was studied using importance sampling, and also studied in many papers thereafter, e.g., [48]. A block diagram of the balanced version of the computing system considered is shown in Fig. 4.

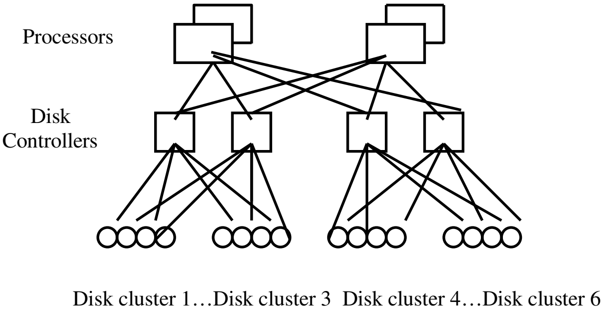


Fig. 4. Block diagram of a computing system

The system is composed of two types of processors each having passive redundancy 2; two types of disk controllers, each having active redundancy 2; and six sets of disk clusters, each having four disks. When a processor of one type fails, it causes a processor of another type to fail also with probability 0.01. The lifetime of all the components is assumed to be exponentially distributed with failure rates of processors, controllers, and disks of 1/2000, 1/2000, and 1/6000 per hour, respectively. It is assumed that each type of component can fail in one of two modes which occur with equal probability. The repair rates for all mode 1 and all mode 2 failures are 1 per hour and 0.5 per hour, respectively. There is a single repairman who fixes failed components in a random-order service. The system is defined to be operational if all data are accessible to both processor types, which means that at least one processor of each type, one controller of each type, and 3 out of 4 disk units in each of the 6 disk clusters are operational. This system was also studied in [27] with redundancy 3, i.e. the system has 3 processors and 3 controllers of each type and 5 disks in each cluster and it is necessary that 3 components of a type fail to have system breakdown, and with redundancy 4.

Probabilities of the order of 10^{-11} were estimated within reasonable computational times because the system is close to be balanced. It corroborates that the importance function $\Phi(t) = cl - oc(t)$ works quite well with balanced systems. Unlike with importance sampling, the simulation with RESTART of the same system but with higher redundancy does not require additional analytical effort.

There are two main ways in which a system may be made highly dependable in a cost-effective manner. The first is to use components that are "highly reliable" and have "low" built-in redundancies in the system. The second is to build "significant" redundancies in the system and use components that are just "reliable" instead of "highly reliable." Unlike with importance sampling, RESTART usually works better with higher redundancies (for estimating probabilities of the same order of magnitude) because it is possible to set more effective thresholds and thus have a lower value of factor f_T. In this sense, they could be considered complementary methods.

Example 3, taken from [33], studies dependability estimation for a consecutive-k-out-of-n: F repairable system with (k-1)-step Markov dependence. The system fails if and only if k or more consecutive components have failed. Exponential or Weibull distributions were considered for the lifetime of components and lognormal distribution for the repair time of a failed component. If there are h ($h < k$) consecutive failed components that precede the component i, the residual lifetime of component i will have failure rates that are greater as h increases. There is one repairman who gives priority to the most critical components. This model is an extension of that introduced in [45] to the case of non-exponential component lifetimes.

The importance function given by formula (24) was also used for simulating this system. In this model there are (n-k+1) minimal cut sets. As all of them have the same cardinality (k), the definition of the importance function can be expressed as: "the number of components that are down at in the cut set with greatest number of failed components". The main differences between the importance, P^*_{A/X_i}, of the system states x_i at events B_i states are: i) whether the failed components of the cut set are consecutive or not, given that the importance is greater if the failed components of the cut set (with greatest number of failed components) are consecutive. And ii) the total number of components in the systems that are down when the process enters each set C_i. The greater is that number, the greater is the importance of the system state. It seems that the difference between the importance of these states could be relatively small. Thus, the variance $V\left(P^*_{A/X_i}\right)$ could be small. Simulation results corroborated this conjecture: the estimated values of factor f_V were very low in all the cases and unreliabilities up to the order of 10^{-16} and steady-state unavailabilities up to the order of 10^{-14} were accurately estimated with short computational effort (13 and 12 minutes, respectively).

In contrast with importance sampling, RESTART is not so dependent on particular features of the system and allows general component lifetime distributions and other generalizations of the model. Although the importance function depends on the system being simulated, the same importance function can be applied to different models without additional analytical effort regardless of the level of redundancy, the number of repairmen and of whether the model is Markovian in nature or not. This feature could extend the use of RESTART for dependability estimation to many other systems. The importance function given by formula (24) seems to lead to good simulation results, at least for balanced systems.

For very unbalanced systems the factor f_V, related with the chosen importance function, can take high values. Further investigation is needed for improving the importance function if it is unfeasible to estimate the probability of interest within a reasonable computational effort.

7 Conclusions

The method RESTART for accelerating rare event simulations has been presented. The paper, mainly based on the research activity of the authors, has described the method, has proved the unbiasedness of the estimator and has shown the formula of its variance. Then the formula of the gain has been obtained and quasi-optimal values for thresholds and the number of retrials that are easy to use in practical applications and lead to a gain close to the optimal one have been derived.

The paper has analyzed the factors that can affect the efficiency of RESTART and has focused on the most critical factor, the one related to the variance of the importance of the states that the system can have when each threshold is hit. As this variance depends on the chosen importance function, guidelines have been provided for the choice of a suitable importance function. The applications of these guidelines has been illustrated with several examples on queuing networks and ultra reliable systems.

In the queuing network examples, simulations of different types of Jackson and non-Jackson networks with different loads of the nodes were shown. The formula of the importance function, initially derived for Jackson networks by combining heuristic arguments with analytical results, could be easily adapted to non-Jackson networks. Buffer overflow probabilities much lower than those needed in practical problems have been accurately estimated within reasonable computational work. It has been shown that the efficiency of RESTART often improves with the complexity of the systems because the dependence of the target queue on the queue length of the other queues is weaker.

In the examples on ultra reliable systems, an importance function obtained heuristically has been applied to the estimation of transient and steady-state reliability measures of different systems. The same importance function has resulted to be valid in all these models regardless of the type of system, the level of redundancy, the number of repairmen and of whether the model is Markovian or non-Markovian.

The examples have shown that efficient applications of RESTART can be achieved though the importance function has been obtained heuristically or using analytical formulas derived with rough approximations. It has also been shown that an importance function derived for a system may be used, sometimes with small modifications, to other systems of the same family or to other time distributions. These two features make easier the use of RESTART and lead to a wide applicability of the method.

References

1. Rubino, G., Tuffin, B. (eds.): Rare event simulation using Monte Carlo methods. Wiley, Chichester (2009)
2. Kahn, H., Harris, T.E.: Estimation of Particle Transmission by Random Sampling. National Bureau of Standards Applied Mathematics Series, vol. 12, pp. 27–30 (1951)
3. Villén-Altamirano, M., Villén-Altamirano, J.: On the Efficiency of RESTART for Multidimensional Systems. ACM T. on Model. and Comput. Simul. 16(3), 251–279 (2006)
4. Bayes, A.J.: Statistical Techniques for Simulation Models. Australian Computer J. 2, 180–184 (1970)

5. Villén-Altamirano, M., Villén-Altamirano, J.: RESTART: A Method for Accelerating Rare Event Simulations. In: Cohen, J.W, Pack, C.D. (eds.) 13th International Teletraffic Congress. North Holland Studies in Telecommunication, vol. 15, pp. 71–76 (1991)
6. Villén-Altamirano, M., Martínez-Marrón, A., Gamo, J.L., Fernández-Cuesta, F.: Enhancement of the Accelerated Simulation Method RESTART by Considering Multiple Thresholds. In: Labetoulle, J., Roberts, J.W. (eds.) 14th International Teletraffic Congress. Teletraffic Science and Engineering, vol. 1a, pp. 797–810. Elsevier, Amsterdam (1994)
7. Hopmans, A.C.M., Kleijnen, J.P.C.: Importance Sampling in System Simulation: A Practical Failure? In: Mathematics and Computing in Simulation XXI, pp. 209–220 (1979)
8. Villén-Altamirano, M., Villén-Altamirano, J.: A Straightforward Method for Fast Simulation of Rare Event. In: 1994 Winter Simulation Conference, pp. 282–289. IEEE Press, Los Alamitos (1994)
9. Görg, C., Schreiber, F.: The RESTART/LRE method for Rare Event Simulation. In: 1996 Winter Simulation Conference, pp. 390–397. IEEE Press, Los Alamitos (1996)
10. Kelling, C.: A Framework for Rare Event Simulation of Stochastic Petri Net using RESTART. In: 1996 Winter Simulation Conference, pp. 317–324. IEEE Press, Los Alamitos (1996)
11. Kuhlmann, T., Kelling, C.: Case Studies on Multi-dimensional RESTART Simulations. Int. J. Electron. Commun. 52(3), 190–196 (1998)
12. Naldi, M., Calonico, F.: A Comparison of the GEVT and RESTART Techniques for the Simulation of Rare Events in ATM Networks. Int. J. of the Federation of Eur. Simul. Societies. 6, 181–186 (1998)
13. Villén-Altamirano, J.: RESTART Method for the Case where Rare Events Can Occur in Retrials from any Threshold. Int. J. Electron. Commun. 52(3), 183–190 (1998)
14. Garvels, M.J.J., Kroese, D.P.: A Comparison of RESTART Implementations. In: 1998 Winter Simulation Conference, pp. 601–609. IEEE Press, Los Alamitos (1998)
15. Görg, C., Fuß, O.: Comparison and Optimization of RESTART Run Time Strategies. Int. J. Electron. Commun. 52(3), 197–204 (1998)
16. Haraszti, Z., Townsend, J.K.: The Theory of Direct Probability Redistribution and its Application to Rare Event Simulation. ACM T. on Model. and Comput. Simul. 9(2), 105–140 (1999)
17. Garvels, M.J.J., Kroese, D.P.: On the Entrance Distribution in RESTART Simulation. In: RESIM 1999 Workshop, pp. 65–88. University of Twente, Enschede (1999)
18. Görg, C., Fuß, O.: Simulating Rare Event Details of ATM Delay-Time Distribution with RESTART-LRE. In: Key, P., Smith, D. (eds.) 16th International Teletraffic Engineering. Teletraffic Science and Engineering, vol. 3b, pp. 777–786. Elsevier, Amsterdam (1999)
19. Akyamac, A.A., Haraszti, Z., Townsend, J.K.: Efficient Rare Event Simulation using DPR for Multi-dimensional Parameter Spaces. In: 16th International Teletraffic Engineering. Teletraffic Science and Engineering, vol. 3b, pp. 767–776. Elsevier, Amsterdam (1999)
20. Tuffin, B., Trivedi, K.S.: Implementation of Importance Splitting Techniques in Stochastic Petri Net Package. In: Haverkort, B.R., Bohnenkamp, H.C., Smith, C.U. (eds.) TOOLS 2000. LNCS, vol. 1786, pp. 216–229. Springer, Heidelberg (2000)
21. Akin, O., Townsed, J.K.: Efficient Simulation of Delay in TCP/IP Networks using DPR-based Splitting. In: IEEE International Conference on Communication 2002, pp. 2619–2624 (2002)
22. Garvels, M.J.J., Kroese, D.P., Ommeren, J.K.C.W.: On the Importance Function in Splitting Simulation. Eur. T. on Telecom. 13(4), 363–371 (2002)

23. Radev, D., Iliev, M., Arabadjieva, I.: RESTART Simulation in ATM networks with tandem queue. In: International Conference on Automatics and Informatics, Sofia, pp. 37–40 (2004)
24. Elayoubi, S.E., Fourestie, B.: On Trajectory Splitting for Accelerating Dynamic Simulations in Mobile Wireless Networks. In: 19th International Teletraffic Congress, pp 1717-1726, Beijing (2005)
25. Villén-Altamirano, J.: Rare Event RESTART Simulation of Two-Stage Networks. Eur. J. of Oper. Res. 179(1), 148–159 (2007)
26. Cerou, F., Guyader, A.: Adaptive Multilevel Splitting for Rare Event Analysis. Stoch. Analysis and Applic. 25(2), 417–443 (2007)
27. Villén-Altamirano, J.: Importance Functions for RESTART Simulation of Highly-Dependable Systems. Simulation 83, 821–828 (2007)
28. Zimmermann, A.: Stochastic discrete event system. Springer, Berlín (2008)
29. Lagnoux, A.: Effective Branching Method Splitting under Cost Constraint. Stoch. Processes and their Application 18(10), 1820–1851 (2008)
30. Mykkeltveit, A., Helvik, B.E.: Application of the RESTART/Splitting Technique to Network Resilience Studies NS2. In: 19th IASTED International Conference (2008)
31. Dean, T., Dupuis, P.: The design and analysis of a generalized DPR/RESTART algorithm for rare event simulation. Annals of Oper. Res. (2010) (in Press)
32. Villén-Altamirano, J.: Importance Functions for RESTART Simulation of General Jackson Networks. Eur. J. of Oper. Res. 203(1), 156–165 (2010)
33. Villén-Altamirano, J.: Dependability Estimation for Non-Markov Consecutive-K-out-of-N: F Repairable Systems by RESTART Simulation. Reliab. Eng. Syst. Saf. 95(3), 247–254 (2010)
34. Villén-Altamirano, M., Villén-Altamirano, J., Vázquez-Gallo, E.: RESTART Simulation of non-Markovian Queuing Networks. In: RESIM 2010, Isaac Newton Institute, Cambridge (2010)
35. Villén-Altamirano, M., Villén-Altamirano, J.: Analysis of RESTART Simulation: Theoretical Basis and Sensitivity Study. Eur. T. on Telecom. 13(4), 373–385 (2002)
36. Villén-Altamirano, J., Villén-Altamirano, M.: Recent Advances in RESTART Simulation. In: RESIM 2008. IRISA - INRIA, Rennes (2008)
37. Villén-Altamirano, M., Villén-Altamirano, J.: Accelerated Simulation of Rare Event using RESTART Method with Hysteresis. In: ITC Specialists' Seminar on Telecommunication Services for Developing Economies, pp. 240–251. University of Mining and Metallurgy, Krakow (1991)
38. Villén-Altamirano, M., Villén-Altamirano, J., González-Rodríguez, J., Río-Martínez, L.: del: Use of Re-Scheduling in Rare Event RESTART Simulation. In: 5th St. Petersburg Workshop on Simulation, pp. 721–728 (2005)
39. Parekh, S., Walrand, A.: A Quick Simulation Method for Excessive Backlogs in Networks of Queues. IEEE T. on Aut. Control 34, 54–66 (1989)
40. Glasserman, P., Heidelberger, P., Shahabuddin, P., Zajic, T.: A Large Deviation Perspective on the Efficiency of Multilevel Splitting. IEEE T. on Aut. Control 43(12), 1666–1679 (1998)
41. Glasserman, P., Heidelberger, P., Shahabuddin, P., Zajic, T.: Multilevel Splitting for Estimating Rare Event Probabilities. Oper. Res. 47, 585–600 (1999)
42. Kroese, D.P., Nicola, V.F.: Efficient Simulation of a Tandem Jackson Network. ACM T. on Model. and Comput. Simul. 12(2), 119–141 (2002)
43. L'ecuyer, P., Demers, V., Tuffin, B.: Rare Events, Splitting and Quasi-Monte Carlo. ACM T. on Model. and Comput. Simul. 17 (2), Article 9 (2007)

44. Villén-Altamirano, J.: RESTART Simulation of Networks of Queues with Erlang Service Times. In: 2009 Winter Simulation Conference, pp. 251–279. IEEE Press, Austin (2009)
45. Xiao, G., Li, Z., Li, T.: Dependability Estimation for non-Markov Consecutive-k-out- of-n: F Repairable Systems by Fast Simulation. Reliab. Eng. Syst. Saf. 92(3), 293–299 (2007)
46. Alexopoulos, C., Shultes, B.C.: Estimating Reliability Measures for Highly-Dependable Markov Systems using Balanced Likelihood Ratios. IEEE T. on Reliab. 50(3), 265–280 (2001)
47. Goyal, A., Shahabuddin, P., Heidelberger, P.: A Unified Framework for Simulating Markovian Models of Highly Dependable Systems. IEEE T. on Comput. 41(1), 36–51 (1992)
48. Nicola, V., Shahabuddin, P., Nakayama, M.K.: Techniques for Fast Simulation of Models of Highly Dependable Systems. IEEE T. on Reliab. 50(3), 246–264 (2001)