



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Computers & Operations Research 32 (2005) 1089–1114

computers &
operations
research

www.elsevier.com/locate/dsw

Efficient implementations of the randomization method with control of the relative error

Víctor Suñé*, Juan A. Carrasco

Departament d'Enginyeria Electrònica, Universitat Politècnica de Catalunya, Diagonal 647, plta. 9, Barcelona 08028, Spain

Abstract

Randomization is a well-known numerical method for the transient analysis of continuous-time Markov chains. The main advantages of the method are numerical stability, well-controlled computation error and ability to specify the computation error in advance. Typical implementations of the method control the truncation error in absolute value, which is not completely satisfactory in some cases. Based on a theoretical result regarding the dependence on the parameter of the Poisson distribution of the relative error introduced when a weighted sum of Poisson probabilities is truncated by the right, in this paper we develop efficient and numerically stable implementations of the randomization method for the computation of two measures on rewarded continuous-time Markov chains with control of the relative error. The numerical stability of those implementations is analyzed using a small example. We also discuss the computational efficiency of the implementations with respect to simpler alternatives.

© 2003 Elsevier Ltd. All rights reserved.

Keywords: Rewarded continuous-time Markov chains; Transient analysis; Randomization; Relative error

1. Introduction

Rewarded continuous-time Markov chains (CTMCs) are a powerful modeling formalism. A rewarded CTMC is a CTMC $X = \{X(t); t \geq 0\}$ with a reward structure imposed over it. The reward structure may include reward rates associated with states and impulse rewards associated with transitions. In this paper, we will consider rewarded CTMCs with a reward structure including only reward rates associated with states. Let Ω be the state space of X and let r_i , $i \in \Omega$ be the reward rate associated with state i . We will assume $r_i \geq 0$, $i \in \Omega$. The quantity r_i has the meaning of “rate at which reward is earned while the CTMC is in state i .” The behavior with time of the random

* Corresponding author.

E-mail addresses: sunye@eel.upc.es (V. Suñé), carrasco@eel.upc.es (J.A. Carrasco).

variable $r_{X(t)}$ can be quantified using several measures. In this paper, we will consider two such measures: the “expected transient reward rate,” $ETRR(t) = E[r_{X(t)}]$, and the “expected averaged reward rate,” $EARR(t) = E[\int_0^t r_{X(\tau)} d\tau/t]$. Those measures with particular instances of the CTMC X and the reward rate structure $r_i, i \in \Omega$ have important applications, particularly in the dependability and performability analysis of fault-tolerant computer systems. Consider, for instance, a fault-tolerant computer system which can be “up” or “down” whose evolution is modeled by a CTMC X with state space $\Omega = U \cup D$, where U includes the states in which the system is up and D includes the states in which the system is down. Then, with a reward rate structure $r_i = 0, i \in U$ and $r_i = 1, i \in D$, the $ETRR(t)$ measure would be the unavailability of the fault-tolerant computer system at time t , i.e. the probability that the system is down at time t . As another example, consider a degradable fault-tolerant multiprocessor system whose behavior is modeled by a CTMC X . Then, with a reward rate structure assigning to each state of X the speedup of the multiprocessor in that particular state, the $ETRR(t)$ measure would be the expected speedup of the multiprocessor at time t and the $EARR(t)$ measure would be the expected averaged speedup of the multiprocessor during the time interval $[0, t]$.

Computation of the measures $ETRR(t)$ and $EARR(t)$ requires the transient analysis of the CTMC X . That transient analysis can be performed using ordinary differential equation solvers and randomization [1,2] (also called uniformization). Good properties of the randomization method are excellent numerical stability, well-controlled computation error, and ability to specify the computation error in advance. The randomization method is based on the randomization result. To review that result, let $\lambda_{i,j}, i, j \in \Omega, i \neq j, \lambda_i = \sum_{j \in \Omega - \{i\}} \lambda_{i,j}, i \in \Omega$, and $\mathbf{A} = (a_{i,j})_{i,j \in \Omega}, a_{i,j} = \lambda_{i,j}, i \neq j, a_{i,i} = -\lambda_i$ be, respectively, the transition rates, output rates and transition rate matrix of X . Let $\Lambda \geq \max_{i \in \Omega} \lambda_i$ and consider the discrete-time Markov chain (DTMC) $\hat{X} = \{\hat{X}_k; k = 0, 1, 2, \dots\}$ with same state space and initial probability distribution as X and transition probability matrix $\mathbf{P} = \mathbf{I} + (1/\Lambda)\mathbf{A}$, where \mathbf{I} denotes an identity matrix of appropriate dimension. The DTMC \hat{X} is said to be the randomized DTMC of X with randomization rate Λ . Let $Q = \{Q(t); t \geq 0\}$ be a Poisson process with arrival rate Λ independent of \hat{X} . We have $P[Q(t) = k] = P_k(\Lambda t)$, with $P_k(\lambda) = e^{-\lambda} \lambda^k / k!$, where $P_k(\lambda)$ is the probability mass function of a Poisson random variable with parameter λ . Then, the *randomization result* states [3, Theorem 4.19] that X is probabilistically identical to $\{\hat{X}_{Q(t)}; t \geq 0\}$, which means that the probability of any event defined over the values of the random variables $X(t), t \geq 0$ is the same as the probability of the corresponding event over the random variables $\hat{X}_{Q(t)}, t \geq 0$.

We will review next typical implementations of the randomization method for the computation of the $ETRR(t)$ and $EARR(t)$ measures with control of the absolute truncation error. Since the computational cost of the method increases with the randomization rate Λ , Λ is taken equal to $\max_{i \in \Omega} \lambda_i$. We will start by the $ETRR(t)$ measure. Using the randomization result, the transient regime of X can be expressed in terms of the transient regime of \hat{X} as

$$P[X(t) = i] = \sum_{k=0}^{\infty} P[\hat{X}_k = i | Q(t) = k] P[Q(t) = k] = \sum_{k=0}^{\infty} P[\hat{X}_k = i] P_k(\Lambda t),$$

and using $ETRR(t) = \sum_{i \in \Omega} r_i P[X(t) = i]$:

$$ETRR(t) = \sum_{i \in \Omega} r_i \sum_{k=0}^{\infty} P[\hat{X}_k = i] P_k(\Lambda t) = \sum_{k=0}^{\infty} d(k) P_k(\Lambda t), \tag{1}$$

with

$$d(k) = \sum_{i \in \Omega} r_i P[\hat{X}_k = i].$$

In the implementation, the infinite series (1) is truncated by the right to obtain an approximate value for $ETRR(t)$:

$$ETRR_R^a(t) = \sum_{k=0}^R d(k) P_k(\Lambda t). \tag{2}$$

Letting $r_{\max} = \max_{i \in \Omega} r_i$, and using $d(k) \leq r_{\max}$, it follows that $r_{\max} \sum_{k=R+1}^{\infty} P_k(\Lambda t)$ upper bounds the truncation error and, then, R is chosen using:

$$R = \min \left\{ m \geq 0: r_{\max} \sum_{k=m+1}^{\infty} P_k(\Lambda t) \leq \varepsilon \right\},$$

where $\varepsilon > 0$ is the requested absolute error. The required probability row vectors of \hat{X} , $\mathbf{q}(k) = (P[\hat{X}_k = i])_{i \in \Omega}$, $k = 0, 1, \dots, R$ are obtained using $\mathbf{q}(0) = \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\alpha_i)_{i \in \Omega}$, $\alpha_i = P[X(0) = i]$ and

$$\mathbf{q}(k + 1) = \mathbf{q}(k)\mathbf{P}. \tag{3}$$

The truncation error upper bound increases with t , because it is r_{\max} times the probability that the number of arrivals in the time interval $[0, t]$ in a Poisson process with arrival rate Λ is $\geq R + 1$. Then, if $ETRR(t)$ has to be computed for several values of t with absolute error $\leq \varepsilon$, it is enough to control the truncation error for the largest of them.

We will review next a typical implementation of the randomization method for the $EARR(t)$ measure with control of the absolute truncation error. Using $EARR(t) = (1/t) \int_0^t E[r_{X(\tau)}] d\tau = (1/t) \int_0^t ETRR(\tau) d\tau$, (1) and $\int_0^t P_k(\Lambda \tau) d\tau = (1/\Lambda) \sum_{l=k+1}^{\infty} P_l(\Lambda t)$ (see, for instance, [4, Formula 14.512]), we can obtain

$$\begin{aligned} EARR(t) &= \frac{1}{t} \int_0^t \sum_{k=0}^{\infty} d(k) P_k(\Lambda \tau) d\tau = \frac{1}{t} \sum_{k=0}^{\infty} d(k) \int_0^t P_k(\Lambda \tau) d\tau = \frac{1}{\Lambda t} \sum_{k=0}^{\infty} d(k) \sum_{l=k+1}^{\infty} P_l(\Lambda t) \\ &= \frac{1}{\Lambda t} \sum_{k=1}^{\infty} \left(\sum_{l=0}^{k-1} d(l) \right) P_k(\Lambda t) = \frac{1}{\Lambda t} \sum_{k=1}^{\infty} \left(\sum_{l=0}^{k-1} d(l) \right) \frac{\Lambda t}{k} P_{k-1}(\Lambda t) \\ &= \sum_{k=0}^{\infty} \left(\frac{1}{k+1} \sum_{l=0}^k d(l) \right) P_k(\Lambda t). \end{aligned} \tag{4}$$

In the implementation, the summatory is truncated by the right to obtain an approximate value for $EARR(t)$:

$$EARR_R^a(t) = \sum_{k=0}^R \left(\frac{1}{k+1} \sum_{l=0}^k d(l) \right) P_k(\Lambda t).$$

Since (4) is formally identical to (1) with $d(k)$ replaced by $d'(k) = (\sum_{l=0}^k d(l))/(k+1)$ and, therefore, $d'(k) \leq r_{\max}$, the truncation point R is chosen as in the method for the $ETRR(t)$ measure. The required probability row vectors $\mathbf{q}(k)$ are obtained also as for the $ETRR(t)$ measure. Similarly, if the measure has to be computed at several time points, it is enough to control the truncation error for the largest of them. Alternative, similar implementations of the randomization method for the computation of $EARR(t)$ can be found in [5].

Computation of the Poisson probabilities $P_k(At) = e^{-At}(At)^k/k!$ is a delicate issue due to the possibility of intermediate overflows and underflows. An approach to compute “exact” Poisson probabilities is to use the method proposed in [6, pp. 1028–1029] (see also [7]), which avoids intermediate overflows and underflows and has good numerical stability. The approach is, however, relatively expensive.

The computational cost of the (standard) randomization method increases with R . Since (see, for instance, [8, Theorem 3.3.5]) the random variable $Q(t)$ has for $At \rightarrow \infty$ an asymptotic normal distribution with mean and variance At , for large At and $\varepsilon \ll 1$, the truncation point R will be approximately equal to At , implying that, for large At and $\varepsilon \ll 1$, the method will be expensive. The high computational cost of the method for large At and $\varepsilon \ll 1$ has motivated the development in the last years of several variants which can outperform the standard randomization method: selective randomization [9,10], multistepping [11, Section 3.1.2], adaptive uniformization [12], adaptive/standard uniformization [13], uniformization with steady-state detection [14,15], and regenerative randomization [16,17]. Randomization-based methods computing bounds which can be much more efficient than “exact” methods have also been proposed [18]. Despite all these variants, the standard randomization method is still competitive for many rewarded CTMC models.

The implementations of the standard randomization just reviewed are, in some cases, not completely satisfactory. The reason is that, in some cases, the user is interested in computing the measure with some reasonably small relative error, while the implementations control the absolute truncation error. In those cases, unless the user knows in advance a good estimate for the order of magnitude of the measure, he/she will be confronted with the dilemma between choosing a very small error control parameter ε to ensure that the desired accuracy will be satisfied, with the consequent increase in the computational cost (the truncation point R increases for decreasing ε), or choosing a greater value for ε and running the risk of obtaining not enough accurate computations and having to run the method again. For very large At , the price paid by using an ε smaller than strictly required is relatively small. However, for not too large At , the price can be relatively important. This is illustrated in Table 1 which gives the required R as a function of ε and At , assuming $r_{\max} = 1$.

In this paper, we will develop efficient and numerically stable implementations of the standard randomization method for the $ETRR(t)$ and $EARR(t)$ measures with control of the relative error. The rest of the paper is organized as follows. Section 2 will obtain a theoretical result concerning the dependence on the parameter of the Poisson distribution of the relative error introduced when a weighted sum of Poisson probabilities is truncated by the right. Based on that result, Section 3 will derive the implementations. Section 4 will analyze the numerical stability of the implementations using a small rewarded CTMC model of a repairable fault-tolerant computer system and will discuss the computational efficiency of the implementations with respect to simpler alternatives. Finally, Section 5 will present some conclusions. The Appendix includes some proofs.

Table 1

Truncation point R as a function of ε and At for $r_{\max} = 1$ in the reviewed implementations of the standard randomization method with control of the absolute truncation error

At	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-10}$	$\varepsilon = 10^{-12}$
0.01	1	2	3	4	4
0.05	2	3	4	5	6
0.1	3	4	5	6	7
0.5	5	7	8	10	11
1	6	9	11	12	14
5	15	19	22	25	27
10	24	28	32	36	39
50	78	87	94	101	107
100	139	151	161	170	178
500	585	610	630	649	665
1000	1120	1154	1182	1208	1230

2. A theoretical result

In this section we will develop a theoretical result concerning the dependence of the relative error with respect to the parameter λ introduced when a weighted sum of Poisson probabilities $\sum_{k=0}^{\infty} \omega(k)P_k(\lambda)$ is truncated by the right, i.e. letting

$$W_R^a(\lambda) = \sum_{k=0}^R \omega(k)P_k(\lambda) \tag{5}$$

and

$$W_R^e(\lambda) = \sum_{k=R+1}^{\infty} \omega(k)P_k(\lambda) \tag{6}$$

we will show that, for $R \geq 0$ and assuming $\omega(k) \geq 0$ and $\omega(k) > 0$ for some k , $0 \leq k \leq R$,

$$e_R(\lambda) = W_R^e(\lambda)/W_R^a(\lambda) \tag{7}$$

increases with λ , $\lambda > 0$.

We will start by the following lemma.

Lemma 1. *Let $\lambda_1 \geq \lambda_2 > 0$, $x = \lambda_1/\lambda_2$, $j \geq i \geq 0$, and $\omega(k) \geq 0$, $k \geq 0$. Then*

$$\frac{1}{x^j} e^{(1-\frac{1}{x})\lambda_1} \sum_{k=i}^j \omega(k)P_k(\lambda_1) \leq \sum_{k=i}^j \omega(k)P_k(\lambda_2) \leq \frac{1}{x^i} e^{(1-\frac{1}{x})\lambda_1} \sum_{k=i}^j \omega(k)P_k(\lambda_1).$$

Moreover, if $\omega(k)$ is uniformly upper bounded, the right inequality also holds for $j = \infty$.

Proof. If $\omega(k) = 0$ for $k, i \leq k \leq j$, the result is trivial. Therefore, let us assume $\omega(k) > 0$ for some $k, i \leq k \leq j$. Substituting $\lambda_1 = x\lambda_2$ into $\sum_{k=i}^j \omega(k)P_k(\lambda_1)$ yields

$$\begin{aligned} \sum_{k=i}^j \omega(k)P_k(\lambda_1) &= \sum_{k=i}^j \omega(k)e^{-\lambda_1} \frac{\lambda_1^k}{k!} = \sum_{k=i}^j \omega(k)x^k e^{-x\lambda_2} \frac{\lambda_2^k}{k!} \\ &= e^{-(x-1)\lambda_2} \sum_{k=i}^j x^k \omega(k)e^{-\lambda_2} \frac{\lambda_2^k}{k!} = e^{-(x-1)\lambda_2} \sum_{k=i}^j x^k \omega(k)P_k(\lambda_2). \end{aligned}$$

Then, since $x \geq 1$ and $\omega(k) \geq 0$,

$$x^i e^{-(x-1)\lambda_2} \sum_{k=i}^j \omega(k)P_k(\lambda_2) \leq \sum_{k=i}^j \omega(k)P_k(\lambda_1) \leq x^j e^{-(x-1)\lambda_2} \sum_{k=i}^j \omega(k)P_k(\lambda_2).$$

Inverting the members of the previous inequalities (this can be done because $\omega(k) > 0$ for some $k, i \leq k \leq j$ implies $\sum_{k=i}^j \omega(k)P_k(\lambda_1) > 0$ and $\sum_{k=i}^j \omega(k)P_k(\lambda_2) > 0$ and, therefore, that all three members are > 0):

$$\frac{e^{(x-1)\lambda_2}}{x^j \sum_{k=i}^j \omega(k)P_k(\lambda_2)} \leq \frac{1}{\sum_{k=i}^j \omega(k)P_k(\lambda_1)} \leq \frac{e^{(x-1)\lambda_2}}{x^i \sum_{k=i}^j \omega(k)P_k(\lambda_2)}.$$

Multiplying by $(\sum_{k=i}^j \omega(k)P_k(\lambda_2)) (\sum_{k=i}^j \omega(k)P_k(\lambda_1)) > 0$,

$$\frac{1}{x^j} e^{(x-1)\lambda_2} \sum_{k=i}^j \omega(k)P_k(\lambda_1) \leq \sum_{k=i}^j \omega(k)P_k(\lambda_2) \leq \frac{1}{x^i} e^{(x-1)\lambda_2} \sum_{k=i}^j \omega(k)P_k(\lambda_1),$$

and, noting that $(x - 1)\lambda_2 = (1 - 1/x)\lambda_1$,

$$\frac{1}{x^j} e^{(1-\frac{1}{x})\lambda_1} \sum_{k=i}^j \omega(k)P_k(\lambda_1) \leq \sum_{k=i}^j \omega(k)P_k(\lambda_2) \leq \frac{1}{x^i} e^{(1-\frac{1}{x})\lambda_1} \sum_{k=i}^j \omega(k)P_k(\lambda_1).$$

Finally, if $\omega(k)$ is uniformly upper bounded, from $\sum_{k=i}^j P_k(\lambda), j \rightarrow \infty, \lambda > 0$ being upper bounded and $\omega(k) \geq 0$, it is easy to prove that $\sum_{k=i}^j \omega(k)P_k(\lambda), j \rightarrow \infty, \lambda > 0$ is upper bounded and, being increasing, convergent, implying that both $\sum_{k=i}^j \omega(k)P_k(\lambda_2), j \rightarrow \infty$ and $(1/x^i)e^{(1-1/x)\lambda_1} \sum_{k=i}^j \omega(k)P_k(\lambda_1), j \rightarrow \infty$ will converge and that the previous right inequality will also hold for $j = \infty$. \square

The following theorem asserts the result which is the purpose of this section.

Theorem 1. Let $\lambda' \geq \lambda > 0, R \geq 0$ and $\omega(k) \geq 0, k \geq 0$. Assume $\omega(k) > 0$ for some $k, 0 \leq k \leq R$ and assume that $\omega(k)$ is uniformly upper bounded. Then, $e_R(\lambda') \geq e_R(\lambda)$.

Proof. Let $x = \lambda'/\lambda \geq 1$. Using (5) and the left inequality of Lemma 1 with $\lambda_1 = \lambda'$, $\lambda_2 = \lambda$, $i = 0$, and $j = R$,

$$W_R^a(\lambda) = \sum_{k=0}^R \omega(k)P_k(\lambda) \geq \frac{1}{x^R} e^{(1-\frac{1}{x})\lambda'} \sum_{k=0}^R \omega(k)P_k(\lambda') = \frac{1}{x^R} e^{(1-\frac{1}{x})\lambda'} W_R^a(\lambda'). \tag{8}$$

Similarly, using (6) and the right inequality of Lemma 1 with $\lambda_1 = \lambda'$, $\lambda_2 = \lambda$, $i = R + 1$, and $j = \infty$,

$$W_R^e(\lambda) = \sum_{k=R+1}^{\infty} \omega(k)P_k(\lambda) \leq \frac{1}{x^{R+1}} e^{(1-\frac{1}{x})\lambda'} \sum_{k=R+1}^{\infty} \omega(k)P_k(\lambda') = \frac{1}{x^{R+1}} e^{(1-\frac{1}{x})\lambda'} W_R^e(\lambda'). \tag{9}$$

Finally, using (7), (8), (9), $x \geq 1$, noting that $\omega(k) > 0$ for some k , $0 \leq k \leq R$ and $\lambda, \lambda' > 0$, implies $W_R^a(\lambda) > 0$ and $W_R^a(\lambda') > 0$:

$$e_R(\lambda) = \frac{W_R^e(\lambda)}{W_R^a(\lambda)} \leq \frac{\frac{1}{x^{R+1}} e^{(1-\frac{1}{x})\lambda'} W_R^e(\lambda')}{\frac{1}{x^R} e^{(1-\frac{1}{x})\lambda'} W_R^a(\lambda')} = \frac{1}{x} \frac{W_R^e(\lambda')}{W_R^a(\lambda')} \leq \frac{W_R^e(\lambda')}{W_R^a(\lambda')} = e_R(\lambda'). \quad \square$$

3. The implementations

In this section, based on Theorem 1, we will derive implementations of the standard randomization method for the computation of the measures $ETRR(t)$ and $EARR(t)$ at several time points t_1, t_2, \dots, t_n with control of the relative error. We will make the following assumptions:

- A1) $r_i > 0$ for some state i reachable from some state j with non-null initial probability.
- A2) $t_i > 0$, $i = 1, 2, \dots, n$.

Assumption A1 implies $d(k) > 0$ for some $k \geq 0$. Note that if assumption A1 is not satisfied, $d(k) = 0$ for all k and, then, $ETRR(t) = EARR(t) = 0$ for all t . Further, for $t = 0$, we have $ETRR(t) = EARR(t) = d(0)$. Therefore, when any of the assumptions A1 or A2 is not satisfied, computation of the $ETRR(t)$ and $EARR(t)$ measures is straightforward.

We will only discuss in detail the implementation for the measure $ETRR(t)$. We will start by, using Theorem 1, developing in Section 3.1 an efficient method for obtaining a global truncation point by the right R guaranteeing a relative truncation error $\leq \varepsilon_1$ at every time point t_i , where $\varepsilon_1 > 0$ is an error control parameter. Next, in Section 3.2, we will discuss the introduction of a truncation point by the left L_i and a truncation point by the right R_i^* , beyond the truncation point by the right R , for each particular time point t_i with the purpose of reducing the number of Poisson probabilities which have to be computed. Those truncations will introduce a further relative truncation error $\leq \varepsilon_2$, where $\varepsilon_2 > 0$ is another error control parameter. In Section 3.3, we will show how weights proportional to the Poisson probabilities can be used to determine the truncation points L_i and R_i^* and how, at the price of introducing an additional relative error, normalized weights can be used instead of “exact” Poisson probabilities to perform the actual computation of the truncated summatories in a

numerically stable way. Also in that section, we will show how the error control parameters ε_1 and ε_2 can be adjusted so as to guarantee an absolute relative error $\leq \varepsilon$ for every time point t_i , where $\varepsilon > 0$ is the requested relative error. Finally, in Section 3.4, we will give algorithmic descriptions of the implementations for both the $ETRR(t)$ and the $EARR(t)$ measures.

3.1. Determination of a global truncation point by the right R

According to the review of the implementation of the standard randomization method for the $ETRR(t)$ measure performed in Section 1, a truncation point R by the right guaranteeing a relative truncation error, $\sum_{k=R+1}^{\infty} d(k)P_k(\lambda t)/ETRR_R^a(t), \leq \varepsilon_1$ at a given time point $t > 0$ can be chosen using

$$R = \min \left\{ m \geq 0: ETRR_m^a(t) > 0 \wedge r_{\max} \left(\sum_{k=m+1}^{\infty} P_k(\lambda t) \right) / ETRR_m^a(t) \leq \varepsilon_1 \right\}. \tag{10}$$

That a finite R can be determined in this way follows from: 1) because $d(k) > 0$ for some k and $t > 0$, $ETRR_m^a(t)$ is (2) > 0 for some m , 2) $ETRR_m^a(t)$ is increasing with m , and 3) $\lim_{m \rightarrow \infty} \sum_{k=m+1}^{\infty} P_k(\lambda t) = 0$. Using Theorem 1 with $\omega(k) = d(k)$ and $\lambda = \lambda t$, the relative truncation error increases with t and, therefore, a global truncation point R guaranteeing a relative truncation error $\leq \varepsilon_1$ at each time point t_i can be obtained using (10) with $t = t_{\max} = \max\{t_1, t_2, \dots, t_n\} > 0$. This is formally established by the following theorem, where $\sum_{k=R+1}^{\infty} d(k)P_k(\lambda t_i)/ETRR_R^a(t_i) \geq 0$ follows from $d(k) \geq 0$.

Theorem 2. *Let*

$$R = \min \left\{ m \geq 0: ETRR_m^a(t_{\max}) > 0 \wedge r_{\max} \left(\sum_{k=m+1}^{\infty} P_k(\lambda t_{\max}) \right) / ETRR_m^a(t_{\max}) \leq \varepsilon_1 \right\}.$$

Then, for $i = 1, 2, \dots, n$,

$$0 \leq \frac{\sum_{k=R+1}^{\infty} d(k)P_k(\lambda t_i)}{ETRR_R^a(t_i)} \leq \varepsilon_1.$$

In the rest of the paper we will refer to the R defined in Theorem 2 simply as R .

Let

$$S_m(\lambda) = \sum_{k=m+1}^{\infty} P_k(\lambda).$$

Determination of R could then be performed by computing $S_m(\lambda t_{\max})$ for increasing values of m . The quantity $S_m(\lambda t_{\max})$ could be computed using $S_m(\lambda t_{\max}) = 1 - \sum_{k=0}^m P_k(\lambda t_{\max})$, but this is numerically unstable when $S_m(\lambda t_{\max})$ is $\ll 1$. Another, better, approach is to use tight upper bounds for $S_m(\lambda t_{\max})$. Let $M > 0$ satisfying $M + 1 > \lambda t_{\max}$, which implies $0 < \lambda t_{\max}/(M + 1) < 1$, and $M \geq m + 2$.

We have:

$$\begin{aligned}
 S_m(\lambda t_{\max}) &= \sum_{k=m+1}^{M-1} P_k(\lambda t_{\max}) + \sum_{k=M}^{\infty} P_k(\lambda t_{\max}) \\
 &< \sum_{k=m+1}^{M-1} P_k(\lambda t_{\max}) + \sum_{k=M}^{\infty} P_M(\lambda t_{\max}) \left(\frac{\lambda t_{\max}}{M+1}\right)^{k-M} \\
 &= \sum_{k=m+1}^{M-1} P_k(\lambda t_{\max}) + P_M(\lambda t_{\max}) \sum_{k=0}^{\infty} \left(\frac{\lambda t_{\max}}{M+1}\right)^k = \tilde{S}_{m,M}(\lambda t_{\max}) + S_M^e(\lambda t_{\max}),
 \end{aligned}$$

with

$$\begin{aligned}
 \tilde{S}_{m,M}(\lambda t_{\max}) &= \sum_{k=m+1}^{M-1} P_k(\lambda t_{\max}), \\
 S_M^e(\lambda t_{\max}) &= P_M(\lambda t_{\max}) \sum_{k=0}^{\infty} \left(\frac{\lambda t_{\max}}{M+1}\right)^k = P_M(\lambda t_{\max}) \frac{1}{1 - \lambda t_{\max}/(M+1)}.
 \end{aligned}$$

The quantity $\tilde{S}_{m,M}(\lambda t_{\max}) + S_M^e(\lambda t_{\max})$ upper bounds $S_m(\lambda t_{\max})$ with relative error $\leq S_M^e(\lambda t_{\max})/\tilde{S}_{m,M}(\lambda t_{\max})$. Then, being ν a reasonable small number, say $\nu=10^{-6}$, we can obtain a tight upper bound for $S_m(\lambda t_{\max})$ by choosing M as the smallest nonnegative integer satisfying $M+1 > \lambda t_{\max}$, $M \geq m+2$ and $S_M^e(\lambda t_{\max})/\tilde{S}_{m,M}(\lambda t_{\max}) \leq \nu$. Computing $S_m(\lambda t_{\max})$ in that way for increasing m , starting at $m=0$, until $ETRR_m^a(t) > 0$ and $r_{\max} S_m(\lambda t_{\max})/ETRR_m^a(t) \leq \varepsilon_1$ would be expensive when the required R is large, which will happen if λt_{\max} is large and $\varepsilon_1 \ll 1$. The computational cost of the determination of the truncation point R will be reduced using two improvements.

The first improvement consists in starting the computation of $S_m(\lambda t_{\max})$ at a value m_0 of m potentially larger than 0. First, observe that from $d(k) \leq r_{\max}$, we can easily obtain $ETRR_m^a(t_{\max}) \leq r_{\max} \sum_{k=0}^m P_k(\lambda t_{\max}) = r_{\max} (1 - \sum_{k=m+1}^{\infty} P_k(\lambda t_{\max})) = r_{\max} (1 - S_m(\lambda t_{\max}))$, which implies $r_{\max} S_m(\lambda t_{\max})/ETRR_m^a(t_{\max}) \geq S_m(\lambda t_{\max})/(1 - S_m(\lambda t_{\max}))$ and, to determine the truncation point R , we can restrict our attention to values of m for which $S_m(\lambda t_{\max})/(1 - S_m(\lambda t_{\max})) \leq \varepsilon_1$, i.e. $S_m(\lambda t_{\max}) \leq \varepsilon_1/(1 + \varepsilon_1)$ and do not examine values of m for which $S_m(\lambda t_{\max}) > \varepsilon_1/(1 + \varepsilon_1)$. Second, we have the following result.

Lemma 2. Let $\lambda \geq 1$ and $\delta, 0 < \delta < 1$. Let $\alpha = \lambda - \lfloor \lambda \rfloor$ and $\delta^* = e^{-0.5+1/(8\lambda)} \sqrt{\lambda/(2\pi \lfloor \lambda \rfloor)}$. Then, for $m \geq 0$, $S_m(\lambda) = \sum_{k=m+1}^{\infty} P_k(\lambda) > \delta$ if any of the following conditions hold:

- (a) $P_{\lfloor \lambda \rfloor}(\lambda) \geq \delta e^{2/\lambda}$ and $m \leq \lfloor \lambda \rfloor - 2 + \sqrt{2\lambda \log(P_{\lfloor \lambda \rfloor}(\lambda)/\delta)}$,
- (b) $P_{\lfloor \lambda \rfloor}(\lambda) \geq \delta$ or $e^{-\alpha}(1/2 + (1/3)e^{-\lfloor \lambda \rfloor} \lfloor \lambda \rfloor^{\lfloor \lambda \rfloor} / \lfloor \lambda \rfloor!) \geq \delta$, and $m \leq \lfloor \lambda \rfloor - 1$,
- (c) $\delta \leq 1 - \delta^*$ and $m \leq \lfloor \lambda \rfloor - 3/2 - \sqrt{\lambda}$,
- (d) $\delta \geq 1 - \delta^*$ and $m \leq \lfloor \lambda \rfloor - 3/2 - \sqrt{1/4 - \lambda \log(2\pi \lfloor \lambda \rfloor (1 - \delta)^2/\lambda)}$.

Proof. See Appendix A. \square

Exploiting the observation and Lemma 2 with $\lambda = \lambda t_{\max}$ and $\delta = \varepsilon_1/(1 + \varepsilon_1)$, to determine the truncation point R , we can start computing $S_m(\lambda t_{\max})$ at $m=m_0$, where m_0 is determined as follows, β being $e^{-0.5+1/(8\lambda t_{\max})} \sqrt{\lambda t_{\max}/(2\pi \lfloor \lambda t_{\max} \rfloor)}$. If $\lambda t_{\max} < 1$, then $m_0=0$. Otherwise, if $P_{\lfloor \lambda t_{\max} \rfloor}(\lambda t_{\max}) \geq (\varepsilon_1/(1 + \varepsilon_1))e^{2/(\lambda t_{\max})}$, then $m_0 = \max\{0, \lfloor \lambda t_{\max} \rfloor - 2 + \sqrt{2\lambda t_{\max} \log(P_{\lfloor \lambda t_{\max} \rfloor}(\lambda t_{\max})/(\varepsilon_1/(1 + \varepsilon_1)))} \rfloor + 1\}$. Otherwise, if $P_{\lfloor \lambda t_{\max} \rfloor}(\lambda t_{\max}) \geq \varepsilon_1/(1 + \varepsilon_1)$ or $e^{-(\lambda t_{\max} - \lfloor \lambda t_{\max} \rfloor)}(1/2 + (1/3)e^{-\lambda t_{\max} \lfloor \lambda t_{\max} \rfloor / \lfloor \lambda t_{\max} \rfloor!}) \geq \varepsilon_1/(1 + \varepsilon_1)$, then $m_0 = \lfloor \lambda t_{\max} \rfloor$, ≥ 0 because $\lambda t_{\max} \geq 1$. Otherwise, $m_0 = \max\{0, \lfloor \lambda t_{\max} \rfloor - 3/2 - \sqrt{\lambda t_{\max}} \rfloor + 1\}$, if $\varepsilon_1/(1 + \varepsilon_1) \leq 1 - \beta$, and $m_0 = \max\{0, \lfloor \lambda t_{\max} \rfloor - 3/2 - \sqrt{1/4 - \lambda t_{\max} \log(2\pi \lfloor \lambda t_{\max} \rfloor (1 - \varepsilon_1/(1 + \varepsilon_1))^2 / (\lambda t_{\max}))} \rfloor + 1\}$, if $\varepsilon_1/(1 + \varepsilon_1) > 1 - \beta$.

The second improvement is the use of $\tilde{S}_{m+1,M}(\lambda t_{\max}) = \tilde{S}_{m,M}(\lambda t_{\max}) - P_{m+1}(\lambda t_{\max})$, $M \geq m + 2$. This yields the following scheme to compute approximate values for $S_m(\lambda t_{\max})$ with relative error upper bounded by ν for increasing values of m , starting at $m=m_0$. Let κ be a constant significantly greater than 1, say $\kappa=100$. Then, for $m=m_0$, we determine the smallest M with $M + 1 > \lambda t_{\max}$ and $M \geq m_0 + 2$ satisfying $S_M^e(\lambda t_{\max})/\tilde{S}_{m_0,M}(\lambda t_{\max}) \leq \nu/\kappa$, set $S_{\text{last}} = \tilde{S}_{m_0,M}(\lambda t_{\max}) + S_M^e(\lambda t_{\max})$, $S_{\text{old}} = S_{\text{last}}$, and approximate $S_m(\lambda t_{\max})$ by S_{last} . Then, for increasing m , if $m \leq M - 2$, we compute $S_{\text{last}} = S_{\text{last}} - P_m(\lambda t_{\max})$ and, if $S_{\text{last}}/S_{\text{old}} \geq 1/\kappa$, approximate $S_m(\lambda t_{\max})$ with S_{last} . If m becomes $> M - 2$ or, being $m \leq M - 2$, $S_{\text{last}}/S_{\text{old}}$ becomes $< 1/\kappa$, we obtain a new M as the smallest integer with $M + 1 > \lambda t_{\max}$ and $M \geq m + 2$ satisfying $S_M^e(\lambda t_{\max})/\tilde{S}_{m,M}(\lambda t_{\max}) \leq \nu/\kappa$, set $S_{\text{last}} = \tilde{S}_{m,M}(\lambda t_{\max}) + S_M^e(\lambda t_{\max})$, $S_{\text{old}} = S_{\text{last}}$, approximate $S_m(\lambda t_{\max})$ with S_{last} , and continue.

For large λt_{\max} and $\varepsilon_1 \ll 1$, both m_0 and R will be approximately equal to λt_{\max} and the two improvements will have an important impact on the cost of determining the truncation point R .

3.2. Determination of truncation points L_i and R_i^* for each time point t_i

In this section, we will discuss the introduction of a truncation point by the left L_i and a truncation point by the right R_i^* , beyond the truncation point by the right R , for each particular time point t_i at which the $ETRR(t)$ measure has to be obtained, with the purpose of reducing the number of Poisson probabilities which have to be computed. The L_i and R_i^* will be selected so that the introduced relative truncation error, $(ETRR_R^a(t_i) - \sum_{k=L_i}^{R_i^*} d(k)P_k(\lambda t_i))/(\sum_{k=L_i}^{R_i^*} d(k)P_k(\lambda t_i))$, is $\leq \varepsilon_2$ with $\varepsilon_2 > 0$.

The relationship $P_{k+1}(\lambda t_i) = ((\lambda t_i)/(k + 1))P_k(\lambda t_i)$, $k \geq 0$ ensures that $P_k(\lambda t_i)$ will be increasing on k for $0 \leq k \leq \lfloor \lambda t_i \rfloor - 1$ and, then, with $L_i \leq \lfloor \lambda t_i \rfloor$, $P_k(\lambda t_i)$ will be increasing on k for $0 \leq k \leq L_i - 1$. Then, letting $d_{\max} = \max_{0 \leq k \leq R} d(k)$ (> 0 because (2), otherwise, $ETRR_R^a(t_i)$ would be 0), we have:

$$\sum_{k=0}^{L_i-1} d(k)P_k(\lambda t_i) \leq L_i d_{\max} P_{L_i-1}(\lambda t_i), \quad L_i \leq \lfloor \lambda t_i \rfloor. \tag{11}$$

With $R_i^* \geq \lfloor \lambda t_i \rfloor$, the relationship $P_{k+1}(\lambda t_i) = ((\lambda t_i)/(k + 1))P_k(\lambda t_i)$ ensures that $P_k(\lambda t_i)$ is decreasing on k for $k \geq R_i^* + 1$. Then, we have:

$$\sum_{k=R_i^*+1}^R d(k)P_k(\lambda t_i) \leq (R - R_i^*) d_{\max} P_{R_i^*+1}(\lambda t_i), \quad R_i^* \geq \lfloor \lambda t_i \rfloor. \tag{12}$$

Then, the largest $L_i \leq R$ guaranteeing $\sum_{k=0}^{L_i-1} d(k)P_k(At_i) / (\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)) \leq \varepsilon_2/2$ through (11) is:

$$L_i = \begin{cases} \max \left\{ m : 1 \leq m \leq \min\{R, \lfloor At_i \rfloor\} \wedge \frac{md_{\max}P_{m-1}(At_i)}{\sum_{k=m}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)} \leq \frac{\varepsilon_2}{2} \right\} \\ \text{if } \frac{md_{\max}P_{m-1}(At_i)}{\sum_{k=m}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)} \leq \frac{\varepsilon_2}{2} \text{ for some } m, 1 \leq m \leq \min\{R, \lfloor At_i \rfloor\} \\ 0 \text{ otherwise} \end{cases}, \quad (13)$$

and a $R_i^* \leq R$ guaranteeing $\sum_{k=R_i^*+1}^R d(k)P_k(At_i) / (\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)) \leq \varepsilon_2/2$ through (12) is:

$$R_i^* = \begin{cases} \min \left\{ m : \min\{R, \lfloor At_i \rfloor\} \leq m \leq R \wedge \frac{(R-m)d_{\max}P_{m+1}(At_i)}{\sum_{k=L_i}^m d(k)P_k(At_i)} \leq \frac{\varepsilon_2}{2} \right\} \\ \text{if } \frac{(R-m)d_{\max}P_{m+1}(At_i)}{\sum_{k=L_i}^m d(k)P_k(At_i)} \leq \frac{\varepsilon_2}{2} \text{ for some } m, \min\{R, \lfloor At_i \rfloor\} \leq m \leq R \\ R \text{ otherwise} \end{cases}. \quad (14)$$

Note that, since the L_i given by (13) is $\leq \lfloor At_i \rfloor$, $P_k(At_i)$ will be increasing on k for $0 \leq k \leq L_i - 1$. Also, when the R_i^* given by (14) is $< R$, $R_i^* \geq \lfloor At_i \rfloor$ and $P_k(At_i)$ will be decreasing on k for $k \geq R_i^* + 1$.

The following lemma establishes that the L_i and R_i^* given by (13) and (14) effectively truncate the summatory $ETRR_R^a(t_i) = \sum_{k=0}^R d(k)P_k(At_i)$.

Lemma 3. *The L_i and R_i^* given by, respectively, (13) and (14) satisfy $0 \leq L_i \leq R_i^* \leq R$.*

Proof. From (13), we have $0 \leq L_i \leq R$. From (14), we have $R_i^* \leq R$. The result follows, then, if $L_i \leq R_i^*$. For $R_i^* = R$, the result follows from, as previously noted, $0 \leq L_i \leq R$. Consider, then, the case $R_i^* < R$. That case is only possible if $(R - R_i^*)d_{\max}P_{R_i^*+1}(At_i) / (\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)) \leq \varepsilon_2/2$. But this is incompatible with $L_i > R_i^*$ because $d_{\max} > 0$. \square

We will prove next that the relative truncation error resulting from the truncation defined by the L_i and R_i^* given by, respectively, (13) and (14), $(ETRR_R^a(t_i) - \sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)) / (\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i))$ is well-defined ($\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)$ is > 0). The proof is done in a sequence of two lemmas.

Lemma 4. *The L_i and R_i^* given by, respectively, (13) and (14) satisfy: a) for $L_i > 0$, $\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i) > 0$; b) for $R_i^* < R$, $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i) > 0$.*

Proof. L_i can only be > 0 if $L_i d_{\max}P_{L_i-1}(At_i) / (\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)) \leq \varepsilon_2/2$, which, because $d_{\max} > 0$, can only happen if $\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)$ is > 0 . This proves part a). Further, R_i^* can only be $< R$ if $(R - R_i^*)d_{\max}P_{R_i^*+1}(At_i) / (\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)) \leq \varepsilon_2/2$, which, because $d_{\max} > 0$, can only happen if $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i) > 0$, proving part b). \square

Lemma 5. For the L_i and R_i^* given by, respectively, (13) and (14), $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i) > 0$.

Proof. For the case $R_i^* < R$, the result follows from Lemma 4, part b). For the case $R_i^* = R$ and $L_i = 0$, the result follows from $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i) = \sum_{k=0}^R d(k)P_k(At_i) = ETRR_R^a(t_i) > 0$, because of the way in which R is selected (Theorem 2). For the case $R_i^* = R$, $L_i > 0$, we have $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i) = \sum_{k=L_i}^R d(k)P_k(At_i) \geq \sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)$ and the result follows from Lemma 4, part a). \square

Note that in the determination of R_i^* using (14) the fraction $(R - m)d_{\max}P_{m+1}(At_i) / (\sum_{k=L_i}^m d(k)P_k(At_i))$ cannot take the form 0/0 because the numerator can only be 0 for $m = R$, in that case the denominator would be $\sum_{k=L_i}^R d(k)P_k(At_i)$, which is > 0 because $d(k) \geq 0$, Lemmas 3 and 5.

The following proposition states that the truncations introduced by the L_i and R_i^* given by (13) and (14) introduce a relative truncation error $\leq \varepsilon_2$.

Proposition 1. For the L_i and R_i^* given by, respectively, (13) and (14):

$$0 \leq \frac{ETRR_R^a(t_i) - \sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)}{\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)} \leq \varepsilon_2.$$

Proof. The left inequality follows immediately from Lemma 3, $d(k) \geq 0$ and Lemma 5. Let us prove the right inequality. Because of Lemma 3, we can write:

$$\frac{ETRR_R^a(t_i) - \sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)}{\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)} = I_{L_i > 0} \frac{\sum_{k=0}^{L_i-1} d(k)P_k(At_i)}{\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)} + I_{R_i^* < R} \frac{\sum_{k=R_i^*+1}^R d(k)P_k(At_i)}{\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)},$$

where I_c denotes the indicator function returning the value 1 if condition c is satisfied and the value 0 otherwise. The right inequality follows, then, if (a) for $L_i > 0$, $(\sum_{k=0}^{L_i-1} d(k)P_k(At_i)) / (\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i))$ is $\leq \varepsilon_2/2$, and (b) for $R_i^* < R$, $\sum_{k=R_i^*+1}^R d(k)P_k(At_i) / (\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i))$ is $\leq \varepsilon_2/2$.

To prove (a), note that from (14) $R_i^* \geq \min\{R, \lfloor At_i \rfloor\}$ and $d(k) \geq 0$ we have $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i) \geq \sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)$, > 0 by Lemma 4, part (a). Also, for $L_i > 0$, (11) and (13) ensure $\sum_{k=0}^{L_i-1} d(k)P_k(At_i) / (\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)) \leq \varepsilon_2/2$. Then, for $L_i > 0$,

$$\frac{\sum_{k=0}^{L_i-1} d(k)P_k(At_i)}{\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)} \leq \frac{\sum_{k=0}^{L_i-1} d(k)P_k(At_i)}{\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k)P_k(At_i)} \leq \frac{\varepsilon_2}{2}.$$

To prove (b), it suffices to note that, for $R_i^* < R$, (12) and (14) ensure $\sum_{k=R_i^*+1}^R d(k)P_k(At_i) / (\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)) \leq \varepsilon_2/2$. \square

Using (13), the truncation point L_i can be determined by computing Poisson probabilities $P_k(At_i)$ for decreasing k , starting at $k = \min\{R, \lfloor At_i \rfloor\}$. Using (14), the truncation point R_i^* can be determined, afterwards, by computing Poisson probabilities $P_k(At_i)$ for increasing k , starting at $k = \min\{R, \lfloor At_i \rfloor\} + 1$. For large At_i , the truncation point L_i given by (13) can be significantly larger than 0 and, for large At_i significantly smaller than At_{\max} , the truncation point R_i^* given by (14) can be significantly smaller than R . In any of those cases, the reduction on the number of Poisson probabilities $P_k(At_i)$ which

have to be computed achieved by the truncations defined by the L_i and R_i^* given by, respectively, (13) and (14) can be significant. In the rest of the paper we will refer to the L_i given by (13) simply as L_i and to the R_i^* given by (14) simply as R_i^* .

3.3. Use of weights for estimating Poisson probabilities and adjustment of the error control parameters

Since (13), (14) the expressions controlling the value of the truncation points L_i and R_i^* are invariant to a scaling of the Poisson probabilities $P_k(At_i)$, it is possible to determine those truncation points using weights (scaled Poisson probabilities) $w_k^i > 0$. Those weights can be obtained using $w_{\min\{R, \lfloor At_i \rfloor\}}^i = 1$ and

$$w_k^i = \frac{k+1}{At_i} w_{k+1}^i, \quad k = \min\{R, \lfloor At_i \rfloor\} - 1, \min\{R, \lfloor At_i \rfloor\} - 2, \dots,$$

$$w_k^i = \frac{At_i}{k} w_{k-1}^i, \quad k = \min\{R, \lfloor At_i \rfloor\} + 1, \min\{R, \lfloor At_i \rfloor\} + 2, \dots$$

Use of those weights instead of “exact” Poisson probabilities is advantageous from a computational point of view.

Consider normalized weights $\tilde{w}_k^i > 0$, $L_i \leq k \leq R_i^*$, obtained from w_k^i , $L_i \leq k \leq R_i^*$ using

$$\tilde{w}_k^i = \frac{w_k^i}{\sum_{l=L_i}^{R_i^*} w_l^i},$$

so that $\sum_{k=L_i}^{R_i^*} \tilde{w}_k^i = 1$. Those normalized weights can be used instead of “exact” Poisson probabilities when computing $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)$. Note that, because \tilde{w}_k^i are scaled versions of $P_k(At_i)$, $\sum_{k=L_i}^{R_i^*} \tilde{w}_k^i = 1$ and $\sum_{k=L_i}^{R_i^*} P_k(At_i) < 1$, we have $\tilde{w}_k^i > P_k(At_i)$. It is intuitively clear that if $\sum_{k=L_i}^{R_i^*} P_k(At_i)$ is only slightly smaller than 1, then the normalized weights \tilde{w}_k^i can be used instead of $P_k(At_i)$ to get an approximation $\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i$ for $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)$ with small relative error. Use of those normalized weights has better numerical stability than the use of “exact” Poisson probabilities and results in a more numerically stable method for computing the $ETRR(t)$ measure at time point t_i .

The following proposition gives a lower bound for $\sum_{k=L_i}^{R_i^*} P_k(At_i)$ in terms of ε_1 and ε_2 , showing that for small ε_1 and ε_2 , $\sum_{k=L_i}^{R_i^*} P_k(At_i)$ is indeed only slightly smaller than 1.

Proposition 2.

$$\sum_{k=L_i}^{R_i^*} P_k(At_i) > 1 - (\varepsilon_1 + \varepsilon_2).$$

Proof. See Appendix A. \square

Using Proposition 2, it is possible to show the following proposition which bounds the relative error $(\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i - \sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)) / (\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i)$ with which $\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i$ gives $\sum_{k=L_i}^{R_i^*} d(k)P_k(At_i)$. Note that, because $\tilde{w}_k^i > 0$ are scaled versions of $P_k(At_i)$, Lemma 5 implies

$\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i > 0$ and, therefore, that $(\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)) / (\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i)$ is well-defined.

Proposition 3.

$$0 < \frac{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} < \varepsilon_1 + \varepsilon_2.$$

Proof. To prove $(\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)) / (\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i) > 0$, let $\tilde{w}_k^i = cP_k(\Lambda t_i)$, where $c > 1$ because $\tilde{w}_k^i > P_k(\Lambda t_i) > 0$. We have:

$$\begin{aligned} & \frac{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} \\ &= \frac{c \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i) - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)}{c \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)} = \frac{c - 1}{c} > 0. \end{aligned}$$

We will prove next $(\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)) / (\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i) \leq \varepsilon_1 + \varepsilon_2$. Using $\tilde{w}_k^i = P_k(\Lambda t_i) / (\sum_{k=L_i}^{R_i^*} P_k(\Lambda t_i))$ we get:

$$\begin{aligned} \frac{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} &= 1 - \frac{\sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} \\ &= 1 - \frac{\sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)}{\sum_{k=L_i}^{R_i^*} d(k) \frac{P_k(\Lambda t_i)}{\sum_{k=L_i}^{R_i^*} P_k(\Lambda t_i)}} = 1 - \sum_{k=L_i}^{R_i^*} P_k(\Lambda t_i). \end{aligned}$$

Then, the result follows from Proposition 2. \square

The following theorem establishes an upper bound for the absolute relative error, $|ETRR(t_i) - \sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i| / (\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i)$ with which $\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i$ gives $ETRR(t_i)$ in terms of ε_1 and ε_2 .

Theorem 3.

$$\frac{|ETRR(t_i) - \sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i|}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} < \varepsilon_1(1 + \varepsilon_2) + \varepsilon_1 + 2\varepsilon_2.$$

Proof. Using Lemma 5 and Proposition 1, we easily obtain $ETRR_R^a(t_i) \leq (1 + \varepsilon_2) \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)$ and $\sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i) \geq ETRR_R^a(t_i) / (1 + \varepsilon_2) > 0$. Then, using $\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i > \sum_{k=L_i}^{R_i^*} d(k)$

$P_k(\Lambda t_i) > 0$, which follows from Proposition 3, $\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i > 0$ and Lemma 5, and $0 \leq |ETRR(t_i) - ETRR_R^a(t_i)|/ETRR_R^a(t_i) \leq \varepsilon_1$, by Theorem 2 and $ETRR(t_i) - ETRR_R^a(t_i) = \sum_{k=R+1}^{\infty} d(k)P_k(\Lambda t_i)$:

$$\begin{aligned} \frac{|ETRR(t_i) - ETRR_R^a(t_i)|}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} &< \frac{|ETRR(t_i) - ETRR_R^a(t_i)|}{\sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)} \\ &\leq (1 + \varepsilon_2) \frac{|ETRR(t_i) - ETRR_R^a(t_i)|}{ETRR_R^a(t_i)} \leq \varepsilon_1(1 + \varepsilon_2). \end{aligned} \tag{15}$$

From $\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i > \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i) > 0$ and Proposition 1:

$$\frac{|ETRR_R^a(t_i) - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)|}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} < \frac{|ETRR_R^a(t_i) - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)|}{\sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)} \leq \varepsilon_2. \tag{16}$$

Finally, using $\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i > 0$, Proposition 3, (15), and (16):

$$\begin{aligned} \frac{|ETRR(t_i) - \sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i|}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} &\leq \frac{|ETRR(t_i) - ETRR_R^a(t_i)|}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} \\ &+ \frac{|ETRR_R^a(t_i) - \sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i)|}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} \\ &+ \frac{|\sum_{k=L_i}^{R_i^*} d(k)P_k(\Lambda t_i) - \sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i|}{\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i} \\ &< \varepsilon_1(1 + \varepsilon_2) + \varepsilon_1 + 2\varepsilon_2. \quad \square \end{aligned}$$

Using Theorem 3, $ETRR(t_i)$ can be computed using $\sum_{k=L_i}^{R_i^*} d(k)\tilde{w}_k^i$ with absolute relative error upper bounded by ε with the selections $\varepsilon_1 = \varepsilon/4$, $\varepsilon_2 = \varepsilon/(4 + \varepsilon/2)$. Those are the selections which will be used in the implementation.

3.4. Algorithmic descriptions

Figs. 1 and 2 describe in C-like syntax the implementation of the standard randomization method for the computation of the $ETRR(t)$ measure at several time points t_1, t_2, \dots, t_n with absolute relative error upper bounded by ε discussed in the previous sections. The implementation has as inputs the CTMC X , the reward rates $r_i, i \in \Omega$, the initial probability distribution row vector $\alpha = (\alpha_i)_{i \in \Omega}$, the required relative error ε , the number of time points n at which $ETRR(t)$ has to be computed, and the time points, t_1, t_2, \dots, t_n , and has as outputs the computed values of the measure at the time points $t_i, \widehat{ETRR}(t_1), \widehat{ETRR}(t_2), \dots, \widehat{ETRR}(t_n)$. The row vector $\mathbf{q} = (q_i)_{i \in \Omega}$ holds probability row vectors of \hat{X} . The implementation has two phases. The first phase includes the determination of the truncation point R and the computation of $d(k), 0 \leq k \leq R$; the second phase (call to the function *Compute_ETRR()*) includes, for each time point t_i , the computation on the variables L and R^* of the truncation points

Inputs: $X, r_i, i \in \Omega, \alpha, \epsilon, n, t_1, t_2, \dots, t_n$
Outputs: $\widetilde{ETRR}(t_1), \widetilde{ETRR}(t_2), \dots, \widetilde{ETRR}(t_n)$
 $\nu = 10^{-6}, \kappa = 100;$
 $r_{\max} = \max_{i \in \Omega} r_i; t_{\max} = \max\{t_1, t_2, \dots, t_n\}; \Lambda = \max_{i \in \Omega} \lambda_i;$
 Obtain the transition probability matrix of $\widehat{X}, \mathbf{P};$
 $\epsilon_1 = \epsilon/4; \epsilon_2 = \epsilon/(4 + \epsilon/2);$
 $m_0 = 0;$
 if $(\Lambda t_{\max} \geq 1)$ {
 $\beta = e^{-0.5 + 1/(8\Lambda t_{\max})} \sqrt{\Lambda t_{\max} / (2\pi \lfloor \Lambda t_{\max} \rfloor)};$
 if $(P_{\lfloor \Lambda t_{\max} \rfloor}(\Lambda t_{\max}) \geq (\epsilon_1 / (1 + \epsilon_1)) e^{2/(\Lambda t_{\max})})$
 $m_0 = \max\{0, \lfloor \Lambda t_{\max} \rfloor - 2 + \sqrt{2\Lambda t_{\max} \log(P_{\lfloor \Lambda t_{\max} \rfloor}(\Lambda t_{\max}) / (\epsilon_1 / (1 + \epsilon_1)))} + 1\};$
 else if $(P_{\lfloor \Lambda t_{\max} \rfloor}(\Lambda t_{\max}) \geq \epsilon_1 / (1 + \epsilon_1))$
 $\| e^{-(\Lambda t_{\max} - \lfloor \Lambda t_{\max} \rfloor)} (1/2 + (1/3)e^{-\lfloor \Lambda t_{\max} \rfloor} \lfloor \Lambda t_{\max} \rfloor^{\lfloor \Lambda t_{\max} \rfloor} / \lfloor \Lambda t_{\max} \rfloor!)} \geq \epsilon_1 / (1 + \epsilon_1)$
 $m_0 = \lfloor \Lambda t_{\max} \rfloor;$
 else if $(\epsilon_1 / (1 + \epsilon_1) \leq 1 - \beta)$ $m_0 = \max\{0, \lfloor \Lambda t_{\max} \rfloor - 3/2 - \sqrt{\Lambda t_{\max}} + 1\};$
 else $m_0 = \max\{0, \lfloor \Lambda t_{\max} \rfloor - 3/2$
 $- \sqrt{1/4 - \Lambda t_{\max} \log(2\pi \lfloor \Lambda t_{\max} \rfloor (1 - \epsilon_1 / (1 + \epsilon_1))^2 / (\Lambda t_{\max}))} + 1\};$
 }
 $\mathbf{q} = \alpha; d(0) = \sum_{i \in \Omega} r_i q_i; trunc_sum = d(0) P_0(\Lambda t_{\max});$
 for $(k = 1; k \leq m_0; k++)$ {
 $\mathbf{nq} = \mathbf{qP}; \mathbf{q} = \mathbf{nq};$
 $d(k) = \sum_{i \in \Omega} r_i q_i; trunc_sum += d(k) P_k(\Lambda t_{\max});$
 }
 $M = \max\{\lfloor \Lambda t_{\max} - 1 \rfloor + 1, m_0 + 2\};$
 $\tilde{S} = \sum_{k=m_0+1}^{M-1} P_k(\Lambda t_{\max}); S^e = P_M(\Lambda t_{\max}) / (1 - \Lambda t_{\max} / (M + 1));$
 while $(S^e / \tilde{S} > \nu / \kappa)$ {
 $\tilde{S} += P_M(\Lambda t_{\max}); M++; S^e = P_M(\Lambda t_{\max}) / (1 - \Lambda t_{\max} / (M + 1));$
 }
 $S_{\text{last}} = \tilde{S} + S^e; S_{\text{old}} = S_{\text{last}};$
 $m = m_0;$
 while $(trunc_sum == 0 \parallel r_{\max} S_{\text{last}} / trunc_sum > \epsilon_1)$ {
 $m++; \mathbf{nq} = \mathbf{qP}; \mathbf{q} = \mathbf{nq};$
 $d(m) = \sum_{i \in \Omega} r_i q_i; trunc_sum += d(m) P_m(\Lambda t_{\max});$
 if $(m \leq M - 2)$ $S_{\text{last}} = S_{\text{last}} - P_m(\Lambda t_{\max});$
 if $(m > M - 2 \parallel S_{\text{last}} / S_{\text{old}} < 1 / \kappa)$ {
 $M = \max\{\lfloor \Lambda t_{\max} - 1 \rfloor + 1, m + 2\};$
 $\tilde{S} = \sum_{k=m+1}^{M-1} P_k(\Lambda t_{\max}); S^e = P_M(\Lambda t_{\max}) / (1 - \Lambda t_{\max} / (M + 1));$
 while $(S^e / \tilde{S} > \nu / \kappa)$ {
 $\tilde{S} += P_M(\Lambda t_{\max}); M++; S^e = P_M(\Lambda t_{\max}) / (1 - \Lambda t_{\max} / (M + 1));$
 }
 $S_{\text{last}} = \tilde{S} + S^e; S_{\text{old}} = S_{\text{last}};$
 }
 }
 $R = m;$
 Compute $\widetilde{ETRR}(t_1, t_2, \dots, t_n, \Lambda, \epsilon_2, R, d(k), 0 \leq k \leq R, \widetilde{ETRR}(t_1), \widetilde{ETRR}(t_2), \dots, \widetilde{ETRR}(t_n));$

Fig. 1. Implementation of the standard randomization method for the computation of the $ETRR(t)$ measure with control of the absolute relative error.

```

Compute_ETRR( $t_1, t_2, \dots, t_n, \Lambda, \epsilon, R, d(k), 0 \leq k \leq R, \widetilde{ETRR}(t_1), \widetilde{ETRR}(t_2), \dots, \widetilde{ETRR}(t_n)$ )
Inputs:  $t_1, t_2, \dots, t_n, \Lambda, \epsilon, R, d(k), 0 \leq k \leq R$ 
Outputs:  $\widetilde{ETRR}(t_1), \widetilde{ETRR}(t_2), \dots, \widetilde{ETRR}(t_n)$ 
 $d_{\max} = \max_{0 \leq k \leq R} d(k)$ ;
for ( $i = 1; i \leq n; i++$ ) {
     $w_{\min\{R, \lfloor \Lambda t_i \rfloor\}} = 1$ ;
     $L = \min\{R, \lfloor \Lambda t_i \rfloor\}$ ;  $trunc\_sum = d(\min\{R, \lfloor \Lambda t_i \rfloor\}) w_{\min\{R, \lfloor \Lambda t_i \rfloor\}}$ ;
    if ( $L > 0$ )  $w_{L-1} = (L / (\Lambda t_i)) w_L$ ;
    while ( $L > 0$  && ( $trunc\_sum == 0 \parallel L d_{\max} w_{L-1} / trunc\_sum > \epsilon / 2$ )) {
         $L--$ ;
         $trunc\_sum += d(L) w_L$ ; if ( $L > 0$ )  $w_{L-1} = (L / (\Lambda t_i)) w_L$ ;
    }
     $R^* = \min\{R, \lfloor \Lambda t_i \rfloor\}$ ; if ( $R^* < R$ )  $w_{R^*+1} = (\Lambda t_i / (R^* + 1)) w_{R^*}$ ;
    while ( $R^* < R$  && ( $trunc\_sum == 0 \parallel (R - R^*) d_{\max} w_{R^*+1} / trunc\_sum > \epsilon / 2$ )) {
         $R^*++$ ;
         $trunc\_sum += d(R^*) w_{R^*}$ ; if ( $R^* < R$ )  $w_{R^*+1} = (\Lambda t_i / (R^* + 1)) w_{R^*}$ ;
    }
    Compute  $S = \sum_{k=L}^{R^*} w_k$ ;
     $factor = 1 / S$ ;
    for ( $k = L; k \leq R^*; k++$ )  $\tilde{w}_k = factor * w_k$ ;
    Compute  $\widetilde{ETRR}(t_i) = \sum_{k=L}^{R^*} d(k) \tilde{w}_k$ ;
}

```

Fig. 2. Description of the procedure *Compute_ETRR* invoked in Figs. 1 and 3.

L_i and R_i^* , the computation on the variables w_k of the weights w_k^i , the computation on the variables \tilde{w}_k of the normalized weights \tilde{w}_k^i , $L_i \leq k \leq R_i^*$, and the computation on $\widetilde{ETRR}(t_i)$ of the approximate value of $ETRR(t_i)$ using $\sum_{k=L_i}^{R_i^*} d(k) \tilde{w}_k^i$. The computation of $d(k)$, $0 \leq k \leq R$ and the computation of approximate values for $S_m(At_{\max})$ for increasing m , starting at m_0 , as discussed in Section 3.1 are embedded in the pseudocode determining the truncation point R . In that pseudocode, computation of $ETRR_m^a(At_{\max})$ from $m = 0$ up to $m = R$ is done on the variable *trunc_sum*. In the call to the function *Compute_ETRR()*, the variable *trunc_sum* is used first to compute $\sum_{k=m}^{\min\{R, \lfloor \Lambda t_i \rfloor\}} d(k) w_k^i$ from $m = \min\{R, \lfloor \Lambda t_i \rfloor\}$ down to $m = L_i$ and, then, to compute $\sum_{k=L_i}^m d(k) w_k^i$ from $m = \min\{R, \lfloor \Lambda t_i \rfloor\}$ up to $m = R_i^*$. The variable may take the value 0 and that case is properly dealt with in the pseudocode.

Using (1) and (4), it follows that the $EARR(t)$ measure has a formalization in terms of $P_k(At)$ identical to that of the $ETRR(t)$ measure with $d(k)$ replaced by $d'(k) = (\sum_{l=0}^k d(l)) / (k + 1)$. This makes all the developments in the previous sections for the measure $ETRR(t)$ to carry over the $EARR(t)$ measure with the only difference that $d(k)$ has to be replaced by $d'(k)$. This allows to adapt easily to the $EARR(t)$ measure the implementation of the standard randomization method for the $ETRR(t)$ measure with control of the absolute relative error. The implementation of the method for the $EARR(t)$ measure is described in Figs. 3 and 2.

4. Analysis and discussion

In this section we will analyze the numerical stability of the implementations of the standard randomization method for the measures $ETRR(t)$ and $EARR(t)$ derived in the previous section using

Inputs: $X, r_i, i \in \Omega, \alpha, \epsilon, n, t_1, t_2, \dots, t_n$
Outputs: $\widehat{EARR}(t_1), \widehat{EARR}(t_2), \dots, \widehat{EARR}(t_n)$
 $\nu = 10^{-6}, \kappa = 100;$
 $r_{\max} = \max_{i \in \Omega} r_i; t_{\max} = \max\{t_1, t_2, \dots, t_n\}; \Lambda = \max_{i \in \Omega} \lambda_i;$
 Obtain the transition probability matrix of $\widehat{X}, \mathbf{P};$
 $\epsilon_1 = \epsilon/4; \epsilon_2 = \epsilon/(4 + \epsilon/2);$
 $m_0 = 0;$
 if $(\Lambda t_{\max} \geq 1)$ {
 $\beta = e^{-0.5+1/(8\Lambda t_{\max})} \sqrt{\Lambda t_{\max}/(2\pi \lfloor \Lambda t_{\max} \rfloor)};$
 if $(P_{\lfloor \Lambda t_{\max} \rfloor}(\Lambda t_{\max}) \geq (\epsilon_1/(1 + \epsilon_1))e^{2/(\Lambda t_{\max})})$
 $m_0 = \max\{0, \lfloor \Lambda t_{\max} \rfloor - 2 + \sqrt{2\Lambda t_{\max} \log(P_{\lfloor \Lambda t_{\max} \rfloor}(\Lambda t_{\max})/(\epsilon_1/(1 + \epsilon_1)))} + 1\};$
 else if $(P_{\lfloor \Lambda t_{\max} \rfloor}(\Lambda t_{\max}) \geq \epsilon_1/(1 + \epsilon_1))$
 $\| e^{-(\Lambda t_{\max} - \lfloor \Lambda t_{\max} \rfloor)}(1/2 + (1/3)e^{-\lfloor \Lambda t_{\max} \rfloor} \lfloor \Lambda t_{\max} \rfloor^{\lfloor \Lambda t_{\max} \rfloor} / \lfloor \Lambda t_{\max} \rfloor!)} \geq \epsilon_1/(1 + \epsilon_1)$
 $m_0 = \lfloor \Lambda t_{\max} \rfloor;$
 else if $(\epsilon_1/(1 + \epsilon_1) \leq 1 - \beta)$ $m_0 = \max\{0, \lfloor \Lambda t_{\max} \rfloor - 3/2 - \sqrt{\Lambda t_{\max}} + 1\};$
 else $m_0 = \max\{0, \lfloor \Lambda t_{\max} \rfloor - 3/2$
 $- \sqrt{1/4 - \Lambda t_{\max} \log(2\pi \lfloor \Lambda t_{\max} \rfloor (1 - \epsilon_1/(1 + \epsilon_1))^2 / (\Lambda t_{\max}))} + 1\};$
 }
 $\mathbf{q} = \alpha; d'(0) = \sum_{i \in \Omega} r_i q_i; trunc_sum = d'(0)P_0(\Lambda t_{\max});$
 for $(k = 1; k \leq m_0; k++)$ {
 $\mathbf{nq} = \mathbf{qP}; \mathbf{q} = \mathbf{nq};$
 $d'(k) = (k d'(k-1) + \sum_{i \in \Omega} r_i q_i)/(k+1); trunc_sum += d'(k)P_k(\Lambda t_{\max});$
 }
 $M = \max\{\lfloor \Lambda t_{\max} - 1 \rfloor + 1, m_0 + 2\};$
 $\tilde{S} = \sum_{k=m_0+1}^{M-1} P_k(\Lambda t_{\max}); S^e = P_M(\Lambda t_{\max})/(1 - \Lambda t_{\max}/(M+1));$
 while $(S^e/\tilde{S} > \nu/\kappa)$ {
 $\tilde{S} += P_M(\Lambda t_{\max}); M++; S^e = P_M(\Lambda t_{\max})/(1 - \Lambda t_{\max}/(M+1));$
 }
 $S_{\text{last}} = \tilde{S} + S^e; S_{\text{old}} = S_{\text{last}};$
 $m = m_0;$
 while $(trunc_sum == 0 \parallel r_{\max} S_{\text{last}}/trunc_sum > \epsilon_1)$ {
 $m++; \mathbf{nq} = \mathbf{qP}; \mathbf{q} = \mathbf{nq};$
 $d'(m) = (m d'(m-1) + \sum_{i \in \Omega} r_i q_i)/(m+1); trunc_sum += d'(m)P_m(\Lambda t_{\max});$
 if $(m \leq M-2)$ $S_{\text{last}} = S_{\text{last}} - P_m(\Lambda t_{\max});$
 if $(m > M-2 \parallel S_{\text{last}}/S_{\text{old}} < 1/\kappa)$ {
 $M = \max\{\lfloor \Lambda t_{\max} - 1 \rfloor + 1, m + 2\};$
 $\tilde{S} = \sum_{k=m+1}^{M-1} P_k(\Lambda t_{\max}); S^e = P_M(\Lambda t_{\max})/(1 - \Lambda t_{\max}/(M+1));$
 while $(S^e/\tilde{S} > \nu/\kappa)$ {
 $\tilde{S} += P_M(\Lambda t_{\max}); M++; S^e = P_M(\Lambda t_{\max})/(1 - \Lambda t_{\max}/(M+1));$
 }
 $S_{\text{last}} = \tilde{S} + S^e; S_{\text{old}} = S_{\text{last}};$
 }
 }
 $R = m;$
 Compute $\widehat{ETRR}(t_1, t_2, \dots, t_n, \Lambda, \epsilon_2, R, d'(k), 0 \leq k \leq R, \widehat{EARR}(t_1), \widehat{EARR}(t_2), \dots, \widehat{EARR}(t_n));$

Fig. 3. Implementation of the standard randomization method for the computation of the $EARR(t)$ measure with control of the absolute relative error.

a small rewarded CTMC model of a repairable fault-tolerant system. We will also discuss the computational efficiency of the implementations with respect to simpler alternatives.

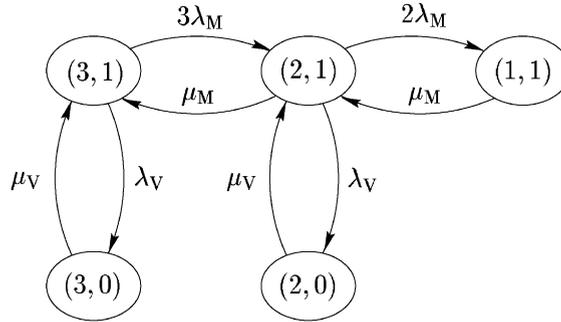


Fig. 4. State transition diagram of the CTMC describing the behavior of the example fault-tolerant system.

The rewarded CTMC model which will be used to analyze the numerical stability of the implementations corresponds to a repairable fault-tolerant computer system using the triple modular redundancy (TMR) technique including three identical processing modules and a voter. The system is up if the voter and at least two processing modules are unfailed. It is assumed that components do not fail when the system is down. Failure and repair times are assumed to have exponential distributions. The failure rate of a processing module is λ_M and the failure rate of the voter is λ_V . The repair rate of a processing module is μ_M and the repair rate of the voter is μ_V . Repairs are performed by a single repairman who gives preemptive priority to the voter. It is assumed that initially all processing modules and the voter are unfailed. Fig. 4 depicts the state transition diagram of the resulting CTMC. The states are labeled by (P_M, V) , where P_M is the number of unfailed processing modules and V has value 1 if the voter is unfailed and value 0 otherwise. The initial state of the CTMC is the state (3, 1). We will consider the reward rate structure:

$$r_i = \begin{cases} 1 & \text{for } i \in \{(3, 0), (2, 0), (1, 1)\}, \\ 0 & \text{for } i \in \{(3, 1), (2, 1)\}. \end{cases}$$

Under that reward rate structure, the $ETRR(t)$ measure is the unavailability at time t , i.e. the probability that the system is down at time t , and the $EARR(t)$ measure is the expected interval unavailability at time t , i.e. the expected value of the fraction of time that the system is down in the time interval $[0, t]$. The numerical experiments will be performed using $\lambda_M = 10^{-3} \text{ h}^{-1}$, $\lambda_V = 10^{-4} \text{ h}^{-1}$, $\mu_M = 0.5 \text{ h}^{-1}$, and $\mu_V = 1 \text{ h}^{-1}$, which yields $A = 1 \text{ h}^{-1}$. The steady-state unavailability of the system is 1.2384×10^{-4} .

The implementations used double precision arithmetic and were run in a Sun-Blade-1000 processor. The actual errors of the numerical results given by the implementations were obtained by comparing those results with an “exact” solution of the model computed using a mathematical software package with 100 digits of accuracy, which was enough to determine with good accuracy the actual errors. Fig. 5 plots the actual relative error in the numerical solution given by the implementation for the $ETRR(t)$ measure when that implementation is run with a single target time t and an absolute relative error requirement ϵ , for several values of ϵ and t . Fig. 6 plots the results obtained by the implementation for the $EARR(t)$ measure. We can note that the actual relative error, ϵ_m , is always smaller than ϵ , even for such stringent values of ϵ as 10^{-10} . For the $ETRR(t)$ measure and not small $At = t$, ϵ_m is extremely small and almost independent

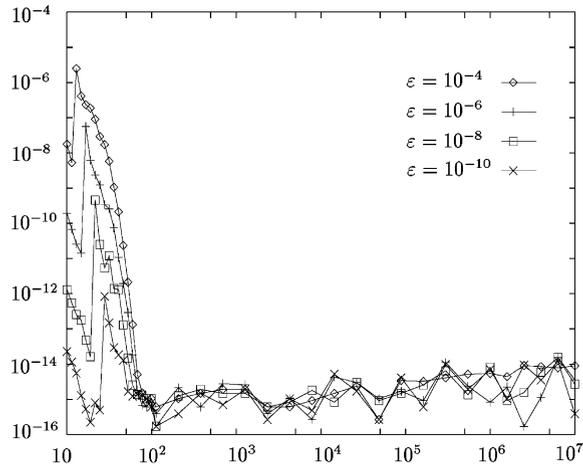


Fig. 5. Actual relative errors in the implementation for the $ETRR(t)$ measure as a function of ε and t (h).

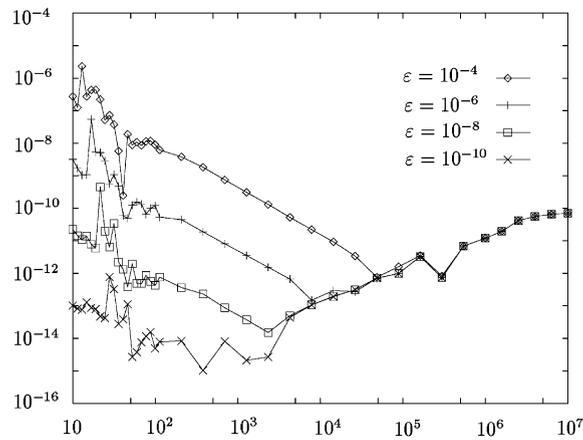


Fig. 6. Actual relative errors in the implementation for the $EARR(t)$ measure as a function of ε and t (h).

of ε . The explanation of that behavior is the following. For not small At , both $ETRR(t)$ and the $d(k)$ corresponding to non-negligible Poisson probabilities $P_k(At)$ are almost identical to the steady-state unavailability and, then, since the implementation computes $ETRR(t)$ by averaging $d(k)$ with weights proportional to the Poisson probabilities $P_k(At)$, the actual error is basically due to round-off errors and, therefore, extremely small and almost independent of the requested absolute relative error, ε . Overall, the implementations for both methods seem to have a very high numerical stability.

Simpler alternatives to the implementations of the standard randomization method developed in this paper would only use the truncation point R defined in Theorem 2 with $\varepsilon_1 = \varepsilon$, ε being the requested relative error, and would estimate the $ETRR(t)$ measure at each time point t_i using $\sum_{k=0}^R d(k)P_k(At_i)$ and the $EARR(t)$ measure at each time point t_i using $\sum_{k=0}^R d'(k)P_k(At_i)$. This would require to

Table 2

Truncation point R used by the implementations developed in the paper and the truncation point R' in simpler alternatives for the example when the implementations for both measures are run with a single target time t and $\varepsilon = 10^{-4}$

t (h)	$ETRR(t)$		$EARR(t)$	
	R	R'	R	R'
0.01	4	4	4	4
0.1	6	6	6	6
1	11	11	11	11
10	33	32	33	32
100	163	161	163	161
10^3	1,189	1,181	1,189	1,181
10^4	10,587	10,562	10,587	10,562
10^5	101,843	101,768	101,843	101,768
10^6	1,005,817	1,005,580	1,005,817	1,005,580
10^7	10,018,383	10,017,634	10,018,383	10,017,634

compute “exact” Poisson probabilities. Call R' the truncation point R in those alternatives. Then, since R decreases with ε_1 , R' would be smaller than the truncation point R of the implementations developed in the paper. This would allow a reduction on the required number of vector-matrix multiplications (3), which for large rewarded CTMC models is an important component of the computational cost of the method. However, being R a smooth function of ε_1 , the reduction will not be, in general, significant. This is illustrated in Table 2 which compares R and R' for the example when the implementations for both measures are run with a single target time t and $\varepsilon = 10^{-4}$. At the price of increasing slightly the truncation point R and the required number of vector-matrix multiplications, the implementations developed in this paper have computational advantages. One of them is a better numerical stability, since available approaches to compute “exact” Poisson probabilities, e.g. [6, pp. 1028–1029], are less stable numerically than the use of normalized weights. The implementations developed in this paper have also computational advantages when λt_{\max} is large and the number of time points n at which the measure has to be computed is large. In that case, the improvements developed in Section 3.1 to determine the truncation point R which are incorporated in the implementations reduce significantly the number of “exact” Poisson probabilities which have to be computed and the truncation points L_i and R_i^* introduced in Section 3.2 make the number of weights and normalized weights which have to be computed for each time point t_i to be significantly smaller than the number of “exact” Poisson probabilities which would have to be computed in the simpler alternative implementations.

5. Conclusions

Based on a theoretical result regarding the dependence on the parameter of the Poisson distribution of the relative error introduced when a weighted sum of Poisson probabilities is truncated by the right, we have developed implementations of the standard randomization method for the computation of the “expected transient reward rate” and the “expected averaged reward rate” measures over rewarded

CTMCs with control of the relative error. The methods seem to exhibit an excellent numerical stability and have computational advantages over simpler alternatives. The implementations developed in the paper are sometimes preferable from a practical point of view over the standard randomization method as it is usually implemented, i.e. with control of the absolute truncation error.

Appendix A.

Proof of Lemma 2. Let $m^* = \lfloor \lambda \rfloor - 2 + \sqrt{2\lambda \log(P_{\lfloor \lambda \rfloor}(\lambda)/\delta)}$. Then, for $0 \leq m \leq m^*$, which implies $m + 1 \leq \lfloor m^* \rfloor + 1$, we have

$$S_m(\lambda) = \sum_{k=m+1}^{\infty} P_k(\lambda) > P_{\lfloor m^* \rfloor + 1}(\lambda). \tag{A.1}$$

From [19, Proposition 5] it follows that, for any strictly positive integer i ,

$$P_{\lfloor \lambda \rfloor + i}(\lambda) \geq P_{\lfloor \lambda \rfloor}(\lambda) \exp\left(-\frac{(i + 1)^2}{2\lambda}\right). \tag{A.2}$$

Noting that $P_{\lfloor \lambda \rfloor}(\lambda) \geq \delta e^{2/\lambda}$ implies $m^* \geq \lfloor \lambda \rfloor$ and, therefore, $\lfloor m^* \rfloor \geq \lfloor \lambda \rfloor$, we have $\lfloor m^* \rfloor + 1 - \lfloor \lambda \rfloor \geq 1$. Then, using (A.1), using (A.2) with $i = \lfloor m^* \rfloor + 1 - \lfloor \lambda \rfloor$, noting that e^{-x^2} is decreasing, and substituting m^* , we obtain, for $0 \leq m \leq m^*$:

$$\begin{aligned} S_m(\lambda) &> P_{\lfloor m^* \rfloor + 1}(\lambda) \geq P_{\lfloor \lambda \rfloor}(\lambda) \exp\left(-\frac{(\lfloor m^* \rfloor + 2 - \lfloor \lambda \rfloor)^2}{2\lambda}\right) \\ &\geq P_{\lfloor \lambda \rfloor}(\lambda) \exp\left(-\frac{(m^* + 2 - \lfloor \lambda \rfloor)^2}{2\lambda}\right) \\ &= P_{\lfloor \lambda \rfloor}(\lambda) \exp\left(-\frac{2\lambda \log(P_{\lfloor \lambda \rfloor}(\lambda)/\delta)}{2\lambda}\right) = \delta. \end{aligned}$$

(b) Assume first $P_{\lfloor \lambda \rfloor}(\lambda) \geq \delta$. Then, for $0 \leq m \leq \lfloor \lambda \rfloor - 1$, which implies $\lfloor \lambda \rfloor \geq m + 1$,

$$S_m(\lambda) = \sum_{k=m+1}^{\infty} P_k(\lambda) > P_{\lfloor \lambda \rfloor}(\lambda) \geq \delta.$$

Assume next $e^{-\alpha}(1/2 + (1/3)e^{-\lfloor \lambda \rfloor} \lfloor \lambda \rfloor^{\lfloor \lambda \rfloor} / \lfloor \lambda \rfloor!) \geq \delta$. Let Z be a Poisson random variable with mean $\lfloor \lambda \rfloor$, and let γ such that $1/2 = P[Z < \lfloor \lambda \rfloor] + \gamma P[Z = \lfloor \lambda \rfloor]$. Using a result given in [20, p. 780], which depends on $\lfloor \lambda \rfloor \geq 1$, $\gamma > 1/3$ and, therefore,

$$\frac{1/2 - P[Z < \lfloor \lambda \rfloor]}{P[Z = \lfloor \lambda \rfloor]} = \gamma > \frac{1}{3}. \tag{A.3}$$

From (A.3) we can obtain $1 - P[Z < \lfloor \lambda \rfloor] > 1/2 + (1/3)P[Z = \lfloor \lambda \rfloor]$ and

$$P[Z \geq \lfloor \lambda \rfloor] > \frac{1}{2} + \frac{1}{3}P[Z = \lfloor \lambda \rfloor]. \tag{A.4}$$

Moreover, since $0 \leq \alpha < 1$ and $\lfloor \lambda \rfloor \geq 1$, for $k \geq 0$,

$$P_k(\lambda) = e^{-\lambda} \frac{\lambda^k}{k!} = e^{-\alpha} e^{-\lfloor \lambda \rfloor} \frac{(\lfloor \lambda \rfloor + \alpha)^k}{k!} \geq e^{-\alpha} e^{-\lfloor \lambda \rfloor} \frac{\lfloor \lambda \rfloor^k}{k!} = e^{-\alpha} P[Z = k].$$

Then, for $0 \leq m \leq \lfloor \lambda \rfloor - 1$, which implies $\lfloor \lambda \rfloor \geq m + 1$, using (A.4), and using $e^{-\alpha}(1/2 + (1/3)e^{-\lfloor \lambda \rfloor}) \lfloor \lambda \rfloor^{\lfloor \lambda \rfloor} / \lfloor \lambda \rfloor! \geq \delta$,

$$\begin{aligned} S_m(\lambda) &= \sum_{k=m+1}^{\infty} P_k(\lambda) \geq \sum_{k=\lfloor \lambda \rfloor}^{\infty} P_k(\lambda) \geq e^{-\alpha} \sum_{k=\lfloor \lambda \rfloor}^{\infty} P[Z = k] = e^{-\alpha} P[Z \geq \lfloor \lambda \rfloor] \\ &> e^{-\alpha} \left(\frac{1}{2} + \frac{1}{3} P[Z = \lfloor \lambda \rfloor] \right) = e^{-\alpha} \left(\frac{1}{2} + \frac{1}{3} e^{-\lfloor \lambda \rfloor} \frac{\lfloor \lambda \rfloor^{\lfloor \lambda \rfloor}}{\lfloor \lambda \rfloor!} \right) \geq \delta. \end{aligned}$$

(c) Let $\bar{\Phi}(x) = \int_x^{\infty} \phi(u) du$, with $\phi(u) = (1/\sqrt{2\pi})e^{-u^2/2}$. It is proved in [21, p. 13] that

$$\sum_{k=0}^m P_k(\lambda) \leq \sqrt{\frac{\lambda}{\lfloor \lambda \rfloor}} e^{1/(8\lambda)} \bar{\Phi} \left(\frac{\lfloor \lambda \rfloor - m - 3/2}{\sqrt{\lambda}} \right), \quad \lambda \geq 1, \quad 0 \leq m \leq \lfloor \lambda \rfloor - 2.$$

From [7, Formula 26.2.12],

$$\bar{\Phi}(x) = \frac{\phi(x)}{x} - \frac{\phi(x)}{x^3} - 15 \int_x^{\infty} \frac{\phi(u)}{u^6} du < \frac{\phi(x)}{x}, \quad x > 0.$$

Then, since $m \leq \lfloor \lambda \rfloor - 2$ implies $\lfloor \lambda \rfloor - m \geq 2$ and $(\lfloor \lambda \rfloor - m - 3/2)/\sqrt{\lambda} > 0$,

$$\begin{aligned} \sum_{k=0}^m P_k(\lambda) &< \sqrt{\frac{\lambda}{2\pi\lfloor \lambda \rfloor}} e^{1/(8\lambda)} \exp \left[-\frac{1}{2} \left(\frac{\lfloor \lambda \rfloor - m - 3/2}{\sqrt{\lambda}} \right)^2 \right] \\ &\times \frac{\sqrt{\lambda}}{\lfloor \lambda \rfloor - m - 3/2} = T_m(\lambda), \quad \lambda \geq 1, \quad 0 \leq m \leq \lfloor \lambda \rfloor - 2. \end{aligned} \tag{A.5}$$

Also, extending the definition of $T_m(\lambda)$ in (A.5) to m real, we have

$$\delta^* = T_{\lfloor \lambda \rfloor - 3/2 - \sqrt{\lambda}}(\lambda). \tag{A.6}$$

Let $f(x) = e^{-x^2/2}/x$. Since $f(x)$ is decreasing for $x > 0$,

$$\begin{aligned} T_m(\lambda) &= \sqrt{\frac{\lambda}{2\pi\lfloor \lambda \rfloor}} e^{1/(8\lambda)} f \left(\frac{\lfloor \lambda \rfloor - m - 3/2}{\sqrt{\lambda}} \right) \\ &\leq \sqrt{\frac{\lambda}{2\pi\lfloor \lambda \rfloor}} e^{1/(8\lambda)} f \left(\frac{\lfloor \lambda \rfloor - m' - 3/2}{\sqrt{\lambda}} \right) \\ &= T_{m'}(\lambda), \quad \lambda \geq 1, \quad 0 \leq m \leq m' \leq \lfloor \lambda \rfloor - 2. \end{aligned} \tag{A.7}$$

Since $\lambda \geq 1$ implies $\lfloor \lambda \rfloor - 3/2 - \sqrt{\lambda} \leq \lfloor \lambda \rfloor - 2$, for $0 \leq m \leq \lfloor \lambda \rfloor - 3/2 - \sqrt{\lambda}$ we can use (A.5) and use (A.7) with $m' = \lfloor \lambda \rfloor - 3/2 - \sqrt{\lambda}$, yielding

$$\sum_{k=0}^m P_k(\lambda) < T_m(\lambda) \leq T_{\lfloor \lambda \rfloor - 3/2 - \sqrt{\lambda}}(\lambda).$$

Then, using (A.6) and recalling that, by assumption, $\delta^* \leq 1 - \delta$,

$$\sum_{k=0}^m P_k(\lambda) < \delta^* \leq 1 - \delta.$$

Finally,

$$S_m(\lambda) = \sum_{k=m+1}^{\infty} P_k(\lambda) = 1 - \sum_{k=0}^m P_k(\lambda) > 1 - (1 - \delta) = \delta.$$

(d) Let $g(x) = e^{-x^2/2}$. Since $g(x)$ is decreasing for $x \geq 0$, retaking the definition of $T_m(\lambda)$ given in (A.5) extended to m real,

$$\begin{aligned} g_m(\lambda) &= T_m(\lambda) \frac{\lfloor \lambda \rfloor - m - 3/2}{\sqrt{\lambda}} = \sqrt{\frac{\lambda}{2\pi \lfloor \lambda \rfloor}} e^{1/(8\lambda)} g\left(\frac{\lfloor \lambda \rfloor - m - 3/2}{\sqrt{\lambda}}\right) \\ &\leq \sqrt{\frac{\lambda}{2\pi \lfloor \lambda \rfloor}} e^{1/(8\lambda)} g\left(\frac{\lfloor \lambda \rfloor - m' - 3/2}{\sqrt{\lambda}}\right) = g_{m'}(\lambda), \quad \lambda \geq 1, \quad 0 \leq m \leq m' \leq \lfloor \lambda \rfloor - 2. \end{aligned} \tag{A.8}$$

A solution $m(\delta)$ of $g_m(\lambda) = 1 - \delta$ is

$$m(\delta) = \lfloor \lambda \rfloor - \frac{3}{2} - \sqrt{\frac{1}{4} - \lambda \log\left(\frac{2\pi \lfloor \lambda \rfloor (1 - \delta)^2}{\lambda}\right)},$$

which is real for $\delta \geq 1 - \delta^*$. Trivially, since $m(\delta)$ is decreasing on δ for $0 < \delta < 1$, and, by assumption, $\delta \geq 1 - \delta^*$, $m(\delta) \leq m(1 - \delta^*)$. It is also easy to check that $m(1 - \delta^*) = \lfloor \lambda \rfloor - 3/2 - \sqrt{\lambda}$. Then, using $\lambda \geq 1$, for $m \leq \lfloor \lambda \rfloor - 3/2 - \sqrt{1/4 - \lambda \log(2\pi \lfloor \lambda \rfloor (1 - \delta)^2/\lambda)} = m(\delta)$,

$$m \leq m(\delta) \leq \lfloor \lambda \rfloor - \frac{3}{2} - \sqrt{\lambda} < \lfloor \lambda \rfloor - 2, \tag{A.9}$$

from which we get

$$\frac{\sqrt{\lambda}}{\lfloor \lambda \rfloor - m - 3/2} \leq 1.$$

Using, then, the definition of $g_m(\lambda)$ in (A.8) we obtain

$$T_m(\lambda) = \frac{\sqrt{\lambda}}{\lfloor \lambda \rfloor - m - 3/2} g_m(\lambda) \leq g_m(\lambda). \tag{A.10}$$

Moreover, since (A.9) $m(\delta) < \lfloor \lambda \rfloor - 2$ and $\lambda \geq 1$, for $0 \leq m \leq m(\delta)$, we have (A.8)

$$g_m(\lambda) \leq g_{m(\delta)}(\lambda) = 1 - \delta. \tag{A.11}$$

Finally, since (A.9) $m < \lfloor \lambda \rfloor - 2$, we can invoke (A.5), which together with (A.10) and (A.11) yields

$$\sum_{k=0}^m P_k(\lambda) < T_m(\lambda) \leq 1 - \delta,$$

and, therefore,

$$S_m(\lambda) = \sum_{k=m+1}^{\infty} P_k(\lambda) = 1 - \sum_{k=0}^m P_k(\lambda) > 1 - (1 - \delta) = \delta. \quad \square$$

Proof of Proposition 2. Assume $L_i > 0$. Taking into account Lemma 4, part a), that, because of Lemma 4, part a), $\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} P_k(At_i) > 0$, that, as noted in Section 3.2, $P_k(At_i)$ is increasing on k for $0 \leq k \leq L_i - 1$, that $0 \leq d(k) \leq d_{\max}$, $L_i \leq k \leq R_i^*$, and the way in which L_i is selected (13):

$$\begin{aligned} \frac{\sum_{k=0}^{L_i-1} P_k(At_i)}{\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} P_k(At_i)} &\leq \frac{L_i P_{L_i-1}(At_i)}{\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} P_k(At_i)} = \frac{L_i d_{\max} P_{L_i-1}(At_i)}{d_{\max} \sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} P_k(At_i)} \\ &\leq \frac{L_i d_{\max} P_{L_i-1}(At_i)}{\sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} d(k) P_k(At_i)} \leq \frac{\varepsilon_2}{2}, \end{aligned}$$

implying (because $0 < \sum_{k=L_i}^{\min\{R, \lfloor At_i \rfloor\}} P_k(At_i) < 1$)

$$\sum_{k=0}^{L_i-1} P_k(At_i) \leq \frac{\varepsilon_2}{2}. \tag{A.12}$$

The previous inequality can be trivially extended to the case $L_i = 0$.

Assume $R_i^* < R$. Taking into account Lemma 5, that, because of Lemma 5, $\sum_{k=L_i}^{R_i^*} P_k(At_i) > 0$, that, as noted in Section 3.2, $P_k(At_i)$ is decreasing on k for $k \geq R_i^* + 1$, that $0 \leq d(k) \leq d_{\max}$, $L_i \leq k \leq R_i^*$, and the way in which R_i^* is selected (14):

$$\begin{aligned} \frac{\sum_{k=R_i^*+1}^R P_k(At_i)}{\sum_{k=L_i}^{R_i^*} P_k(At_i)} &\leq \frac{(R - R_i^*) P_{R_i^*+1}(At_i)}{\sum_{k=L_i}^{R_i^*} P_k(At_i)} = \frac{(R - R_i^*) d_{\max} P_{R_i^*+1}(At_i)}{d_{\max} \sum_{k=L_i}^{R_i^*} P_k(At_i)} \\ &\leq \frac{(R - R_i^*) d_{\max} P_{R_i^*+1}(At_i)}{\sum_{k=L_i}^{R_i^*} d(k) P_k(At_i)} \leq \frac{\varepsilon_2}{2}, \end{aligned}$$

implying (again, because $0 < \sum_{k=L_i}^{R_i^*} P_k(At_i) < 1$)

$$\sum_{k=R_i^*+1}^R P_k(At_i) \leq \frac{\varepsilon_2}{2}. \tag{A.13}$$

The previous inequality can be trivially extended to the case $R_i^* = R$.

Finally, since $\sum_{k=R+1}^{\infty} P_k(At)$ increases with t , because it is the probability that the number of arrivals in a Poisson process with arrival rate λ in the time interval $[0, t]$ is $\geq R + 1$, taking into account the way in which R is selected (10) and using $0 < ETRR_R^a(t_{\max}) \leq r_{\max}$,

$$\sum_{k=R+1}^{\infty} P_k(At_i) \leq \sum_{k=R+1}^{\infty} P_k(At_{\max}) \leq \frac{\varepsilon_1}{r_{\max}} ETRR_R^a(t_{\max}) \leq \varepsilon_1. \tag{A.14}$$

Finally, combining (A.12), (A.13) and (A.14) and using $\sum_{k=0}^{\infty} P_k(At) = 1$:

$$\sum_{k=L_i}^{R_i^*} P_k(At_i) = 1 - \sum_{k=0}^{L_i-1} P_k(At_i) - \sum_{k=R_i^*+1}^R P_k(At_i) - \sum_{k=R+1}^{\infty} P_k(At_i) \geq 1 - (\varepsilon_1 + \varepsilon_2). \quad \square$$

References

- [1] Grassman WK. Transient solutions in Markovian queuing systems. *Computers and Operations Research* 1977;4: 47–53.
- [2] Gross D, Miller DR. The randomization technique as a modelling tool and solution procedure for transient Markov processes. *Operations Research* 1984;32:343–61.
- [3] Kijima M. *Markov processes for stochastic modeling*. London: Chapman & Hall; 1997.
- [4] Spiegel MR. *Mathematical handbook of formulas and tables*. New York: McGraw-Hill; 1970.
- [5] Reibman A, Trivedi K. Transient analysis of cumulative measures of Markov model behavior. *Communications in Statistics—Stochastic Models* 1989;5(4):683–710.
- [6] Knüsel L. Computation of the chi-square and Poisson distribution. *SIAM Journal of Scientific and Statistical Computing* 1986;7(3):1,022–36.
- [7] Abramowitz M, Stegun IA, editors. *Handbook of mathematical functions*. New York: Dover; 1964.
- [8] Ross SM. *Stochastic processes*. New York: Wiley; 1983.
- [9] Melamed B, Yadin M. Randomization procedures in the computation of cumulative-time distributions over discrete state Markov processes. *Operations Research* 1984;32(4):926–44.
- [10] Miller DR. Reliability calculation using randomization for Markovian fault-tolerant computing systems. In: *Proceedings of the 13th IEEE International Symposium on Fault-Tolerant Computing, Milano, Italy, FTCS-13, 1983*. p. 284–9.
- [11] Reibman A, Trivedi KS. Numerical transient analysis of Markov models. *Computers and Operations Research* 1988;15:19–36.
- [12] van Moorsel AP, Sanders WH. Adaptive uniformization. *Communications in Statistics—Stochastic Models* 1994;10(3):619–47.
- [13] van Moorsel AP, Sanders WH. Transient solution of Markov models by combining adaptive & standard uniformization. *IEEE Transactions on Reliability* 1997;46(3):430–40.
- [14] Malhotra M, Muppala JK, Trivedi KS. Stiffness-tolerant methods for transient analysis of stiff Markov chains. *Microelectronics and Reliability* 1994;34(11):1,825–41.
- [15] Sericola B. Availability analysis of repairable computer systems and stationarity detection. *IEEE Transactions on Computers* 1999;48(11):1166–72.
- [16] Carrasco JA. Transient analysis of large Markov models with absorbing states using regenerative randomization. Technical Report DMSD_99_2, Universitat Politècnica de Catalunya, February 1999, available at <ftp://ftp-eel.upc.es/techreports>.
- [17] Carrasco JA. Computation of bounds for transient measures of large rewarded Markov models using regenerative randomization. *Computers and Operations Research* 2003;30(7):1005–35.
- [18] Carrasco JA. Computationally efficient and numerically stable reliability bounds for repairable fault-tolerant systems. *IEEE Transactions on Computers* 2002;51(3):254–68.
- [19] Fox BL, Glynn PW. Computing Poisson probabilities. *Communications of the ACM* 1988;31(4):440–5.
- [20] Siegel A. Median bounds and their application. *ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, Maryland, USA, 1999. p. 776–85.
- [21] Glynn PW. Upper bounds on Poisson tail probabilities. *Operations Research Letters* 1987;6(1):9–14.