

Transient Analysis of Large Markov Models with Absorbing States using Regenerative Randomization

Juan A. Carrasco
Departament d'Enginyeria Electrònica
Universitat Politècnica de Catalunya
Diagonal 647, plta. 9
08028 Barcelona, Spain
carrasco@eel.upc.edu

Technical report DMSD_99_2

Last revision: April 30, 2005

appeared in reduced form in *Communications in Statistics—Simulation and Computation*

Abstract

In this paper, we develop a new method, called regenerative randomization, for the transient analysis of continuous time Markov models with absorbing states. The method has the same good properties as standard randomization: numerical stability, well-controlled computation error, and ability to specify the computation error in advance. The method has a benign behavior for large t and is significantly less costly than standard randomization for large enough models and large enough t . For a class of models, class C, including typical failure/repair reliability models with exponential failure and repair time distributions and repair in every state with failed components, stronger theoretical results are available assessing the efficiency of the method in terms of “visible” model characteristics. A large example belonging to that class is used to illustrate the performance of the method and to show that it can indeed be much faster than standard randomization.

Index terms: Continuous time Markov chains, transient analysis, randomization, reliability, fault-tolerant systems.

1 Introduction

Homogeneous continuous time Markov chains (CTMCs) are frequently used for performance, dependability and performability modeling. The transient analysis of these models is usually significantly more costly than the steady-state analysis, and very costly in absolute terms when the CTMC is large. This makes the development of efficient transient analysis techniques for CTMCs a research topic of great interest. Commonly used methods are ODE (ordinary differential equation) solvers and randomization. Good recent reviews of these methods with new results can be found in [17, 18, 26]. The randomization method (also called uniformization) is attractive because of its excellent numerical stability and the facts that the computation error is well-controlled and can be specified in advance¹. It was first proposed by Grassman [11] and has been further developed by Gross and Miller [13]. The method is offered by well-known performance, dependability and performability modeling packages [2, 7, 8, 10]. The randomization method is based on the following result [15, Theorem 4.19]. Let $X = \{X(t); t \geq 0\}$ be a CTMC with finite state space Ω ; let $\lambda_{i,j}$, $i, j \in \Omega$, $i \neq j$ be the transition rate of X from state i to state j , and let $\lambda_i = \sum_{j \in \Omega - \{i\}} \lambda_{i,j}$, $i \in \Omega$ be the output rate of state i . Consider any $\Lambda \geq \max_{i \in \Omega} \lambda_i$ and define the homogeneous 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 probabilities $P[\hat{X}_{k+1} = j | \hat{X}_k = i] = P_{i,j} = \lambda_{i,j}/\Lambda$, $i \neq j$, $P[\hat{X}_{k+1} = i | \hat{X}_k = i] = P_{i,i} = 1 - \lambda_i/\Lambda$. Let $Q = \{Q(t); t \geq 0\}$ be a Poisson process with arrival rate Λ ($P[Q(t) = k] = e^{-\Lambda t}(\Lambda t)^k/k!$) independent of \hat{X} . Then, $X = \{X(t); t \geq 0\}$ is probabilistically identical to $\{\hat{X}_{Q(t)}; t \geq 0\}$ (we call this the *randomization result*). The DTMC \hat{X} is called the randomized DTMC of X with rate Λ . The CTMC X is called the derandomized CTMC of \hat{X} with rate Λ .

Assume that a reward rate structure, $r_i \geq 0$, $i \in \Omega$ is defined over the state space of X . The quantity r_i has the meaning of ‘‘rate’’ at which reward is earned while X is in state i . Then, a useful measure to consider is the expected transient reward rate $ETRR(t) = E[r_{X(t)}]$. Using the facts that $X = \{X(t); t \geq 0\}$ and $\{\hat{X}_{Q(t)}; t \geq 0\}$ are probabilistically identical and that \hat{X} and Q are independent, we can express $ETRR(t)$ in terms of the transient regime of \hat{X} as

$$\begin{aligned} ETRR(t) &= \sum_{i \in \Omega} r_i P[X(t) = i] = \sum_{i \in \Omega} r_i \sum_{k=0}^{\infty} P[\hat{X}_k = i] P[Q(t) = k] \\ &= \sum_{k=0}^{\infty} \sum_{i \in \Omega} r_i P[\hat{X}_k = i] e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} = \sum_{k=0}^{\infty} d(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}, \end{aligned} \quad (1)$$

with $d(k) = \sum_{i \in \Omega} r_i P[\hat{X}_k = i]$. Denoting by $\mathbf{q}(k) = (P[\hat{X}_k = i])_{i \in \Omega}$ the probability row vector of \hat{X} at step k , $\mathbf{q}(k)$, $k > 0$ can be obtained from $\mathbf{q}(0)$ using

$$\mathbf{q}(k+1) = \mathbf{q}(k)\mathbf{P}, \quad (2)$$

where $\mathbf{P} = (P_{i,j})_{i,j \in \Omega}$ is the transition probability matrix of \hat{X} .

¹The computation error has two components: truncation error and round-off error; the truncation error can be made arbitrarily small, the round-off error will have a very small relative value due to the numerical stability of the method if double precision is used. Rigorous bounds for the round-off error have been obtained in [12] under certain conditions concerning the values that transition rates can have and assuming a special method for computing Poisson probabilities.

In a practical implementation of the randomization method, an approximate value for $ETRR(t)$, $ETRR_N^a(t)$, is obtained by truncating the summatory (1) so that N steps have to be given to \hat{X} :

$$ETRR_N^a(t) = \sum_{k=0}^N d(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!},$$

and, taking into account that $0 \leq d(k) \leq r_{\max} = \max_{i \in \Omega} r_i$, the truncation error verifies

$$0 \leq ETRR(t) - ETRR_N^a(t) \leq r_{\max} \sum_{k=N+1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}.$$

A usual accuracy requirement is to limit the truncation error to a value $\leq \varepsilon$. Then, N is chosen as

$$N = \min \left\{ m \geq 0 : r_{\max} \sum_{k=m+1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} \leq \varepsilon \right\}.$$

Stable and efficient computation of the Poisson probabilities $e^{-\Lambda t} (\Lambda t)^k / k!$ avoiding overflows and intermediate underflows is a delicate issue and several alternatives have been proposed [4, 9, 16, 22]. Our implementation of both standard randomization and regenerative randomization use the method described in [16, pp. 1028–1029] (see also [1]), which has good numerical stability.

For large models, the computational cost of the randomization method is roughly due to the N vector-matrix multiplications (2). The truncation parameter N increases with Λt and, for that reason, Λ is usually taken equal to $\max_{i \in \Omega} \lambda_i$. Using the well-known result [27] that $Q(t)$ has for $\Lambda t \rightarrow \infty$ an asymptotic normal distribution with mean and variance Λt , it is easy to realize that, for large Λt and $\varepsilon \ll 1$, the required N will be $\approx \Lambda t$. If one is interested in solving the model for values of t for which Λt is very large, randomization will be highly inefficient. Consider, for instance, a CTMC dependability model of a fault-tolerant system with hot restarts having an exponential duration with mean 1 minute so that Λ is of the order of 1 min^{-1} . For a time $t = 1$ year, $\Lambda t \approx 525,600$, making randomization very inefficient if X is large.

The randomization result can also be exploited to develop methods to compute more complex measures such as the distribution of the interval availability [28, 29, 33] and the performability [23, 25, 34, 35]. The performance of those methods also degrades as Λt increases.

Several variants of the (standard) randomization method have been proposed to improve its efficiency. Miller has used selective randomization to solve reliability models with detailed representation of error handling activities [20]. The idea behind selective randomization [19] is to randomize the model only in a subset of the state space. Reibman and Trivedi [26] have proposed an approach based on the multistep concept. The idea is to compute \mathbf{P}^M explicitly, where M is the length of the multistep, and use the recurrence $\mathbf{q}(k+M) = \mathbf{q}(k)\mathbf{P}^M$ to advance \hat{X} faster for steps which have negligible contributions to the transient solution of X . Since, for large Λt , the number of $\mathbf{q}(k)$'s with significant contributions is of the order of $\sqrt{\Lambda t}$, the multistep concept allows a significant reduction of the required number of vector-matrix multiplications. However, when computing \mathbf{P}^M , significant fill-in can occur if \mathbf{P} is sparse. Adaptive uniformization [21] is a recent method in which the randomization rate is adapted depending on the states in which the randomized DTMC can be at a given step. Numerical experiments have shown that adaptive uniformization can be faster than standard randomization for short to medium mission times. In addition, it can be used to solve

models with infinite state spaces and not uniformly bounded output rates. Recently, it has been proposed to combine adaptive and standard uniformization to obtain a method which outperforms both for most models [22]. Another recent proposal to speed up the randomization method is steady-state detection [17]. A method performing steady-state detection with error bounds has been developed [32]. Steady-state detection is useful for models which reach their steady-state before the largest time at which the measure has to be computed.

In this paper, we focus on CTMC models with absorbing states and develop a method called *regenerative randomization* for their transient analysis. Specifically, we will consider CTMCs X with finite state space $\Omega = S \cup \{f_1, f_2, \dots, f_A\}$, $|S| \geq 2$, $A \geq 1$, where f_i are absorbing states and will consider the particular instance of the *ETRR*(t) measure, $m(t) = \sum_{i=0}^A r_{f_i} P[X(t) = f_i]$, where all reward rates $r_{f_i} \geq 0$ are different. The method will require the selection of a “regenerative” state $r \in S$. It will be assumed that either a) all states in S are transient, or b) S has a single recurrent class of states and the selected regenerative state r belongs to that class. It will be also assumed that all states are reachable from some state with nonnull initial probability and, to simplify the description of the method, that X has a nonnull transition rate from r to $S' = S - \{r\}$. The latter condition can, however, be circumvented in practice by adding, if X has no transition rate from r to S' , a tiny transition rate $\lambda \leq 10^{-10}\varepsilon/(2r_{\max}t_{\max})$ from r to some state in S' , where ε is the allowed error, $r_{\max} = \max_{1 \leq i \leq A} r_{f_i}$ and t_{\max} is the largest time at which $m(t)$ has to be computed, with a negligible impact on $m(t) \leq 10^{-10}\varepsilon$, $t \leq t_{\max}^2$.

The measure $m(t)$ for CTMC models with the assumed structure has important applications. Thus, S could include the operational states of a fault-tolerant system and entry into a single absorbing state f_1 could model the failure of the system. Then, with $r_{f_1} = 1$ and $P[X(0) = f_1]$ equal to the probability that initially the system is failed, the $m(t)$ measure would be the unreliability of the system at time t . As another example, S could include a proper subset of the set of operational states of a fault-tolerant system, entry into an absorbing state f_1 could model system failure, and entry into another absorbing state f_2 could model entry into an operational state not in S . Then, with $r_{f_1} = 1$, $r_{f_2} = 0$, $P[X(0) = f_1]$ equal to the probability that initially the system is failed, and $P[X(0) = f_2]$ equal to the probability that initially the system is in an operational state outside S , $m(t)$ would be a lower bound for the unreliability of the system at time t . As a third example, S could include a proper subset of the set of operational states of a fault-tolerant system and entry into a single absorbing state f_1 could model either entry into an operational state outside S or system failure. Then, with $r_{f_1} = 1$ and $P[X(0) = f_1]$ equal to the probability that initially the system is in an operational state outside S or failed, $m(t)$ would be an upper bound for the unreliability of the system at time t . Applications also exist in the performance domain. Thus, the states in S could be states visited by a system while completing a task and entry into a single absorbing state f_1 could model task completion. Then, $m(t)$ would be the probability distribution function of the task completion time.

Let r be the selected regenerative state. The basic idea in regenerative randomization is to obtain a transformed CTMC model, of potentially smaller size than the original CTMC model, by

²Let $\mathbf{p}(\lambda, t)$ be the probability distribution column vector of X at time t as a function of the added transition rate λ from r to S' . Using Lemma 1 in the Appendix, $\|\mathbf{p}(\lambda, t) - \mathbf{p}(0, t)\|_1 \leq \|\mathbf{A}_\lambda^T\|_1 t$, where \mathbf{A}_λ^T is a matrix with all elements null except an element with value λ and an element with value $-\lambda$, both in the column corresponding to state r . Then, $\|\mathbf{A}_\lambda^T\|_1 = 2\lambda$ and $\|\mathbf{p}(\lambda, t) - \mathbf{p}(0, t)\|_1 \leq 2\lambda t$, implying that the absolute error in $m(t)$ due to the addition of the transition rate λ is upper bounded by $2r_{\max}\lambda t$.

characterizing with enough accuracy the behavior of the original model from S' up to state r or a state f_i and from r until next hit of r or a state f_i , and solve the transformed CTMC model by standard randomization. The performance of the method depends, of course, on the selection for r . The method offers the same good properties as standard randomization: numerical stability, well-controlled computation error, and ability to specify the computation error in advance, and can be much faster than standard randomization. The rest of the paper is organized as follows. In Section 2, we develop and describe the method. In Section 3, we state the so-called benign behavior of the method, which implies that for large enough models and large enough t , the method will be significantly less costly than standard randomization. Also, for a class of models, class C, including typical failure/repair reliability models with exponential failure and repair time distributions and repair in every state with failed components, we obtain stronger theoretical results assessing the computational cost of the method in terms of “visible” model characteristics. In Section 4, using a large reliability model belonging to class C, we illustrate the performance of the method and show that it can indeed be much faster than standard randomization. Section 5 discusses preliminary related work. Finally, Section 6 concludes the paper. The Appendix includes long proofs and a technical result (Lemma 1).

2 Regenerative Randomization

As previously said, the regenerative randomization method combines a model transformation step with the solution of the transformed model by standard randomization. Such decomposition is conceptually clear and furthermore allows the future development of variants of the method by simply using alternative methods to solve the transformed model. The model transformation step can, conceptually, be further decomposed into two steps. In the first step, a CTMC model, V , with infinite state space is obtained such that the $m(t)$ measure can be expressed exactly in terms of the transient regime of V . Intuitively, such a model transformation is done by characterizing, using states, the behavior of X from S' up to state r or a state f_i and from r until next hit of r or a state f_i . In the second step, V is truncated to obtain a CTMC model with finite state space whose transient regime gives with some upper bounded, arbitrarily small error the $m(t)$ measure. An important aspect of the method is the use of computationally inexpensive upper bounds for the model truncation error.

In this section, we will develop and describe algorithmically the method. We will also show that the method has the same good properties as standard randomization and will analyze its memory overhead with respect to standard randomization. Theoretical properties of the method regarding its computational cost will be investigated in Section 3. The rest of the section is organized as follows. Section 2.1 will derive and describe the transformed CTMC model V with infinite state space. Section 2.2 will deal with the truncation of V . Finally, Section 2.3 will give an algorithmic description of the method, will show that the method has the same good properties as standard randomization, and will analyze its memory overhead with respect to standard randomization.

As in the previous section, we will denote by $\lambda_{i,j}$, $i, j \in \Omega$, $j \neq i$ the transition rates of X , by λ_i , $i \in \Omega$ the output rates of X , by $\hat{X} = \{\hat{X}_k; k = 0, 1, 2 \dots\}$ the randomized DTMC of X with randomization rate Λ , by $P_{i,j}$, $i, j \in \Omega$ the transition probabilities of \hat{X} , and by $\mathbf{P} = (P_{i,j})_{i,j \in \Omega}$ the transition probability matrix of \hat{X} . We will also use the notation $\lambda_{i,B} = \sum_{j \in B} \lambda_{i,j}$, $B \subset \Omega - \{i\}$,

$P_{i,B} = \sum_{j \in B} P_{i,j}$, $B \subset \Omega$, $\alpha_i = P[X(0) = i]$, and $\alpha_B = \sum_{i \in B} \alpha_i$, $B \subset \Omega$. Given a DTMC Y , we will denote by $Y_{m:n}c$ the predicate which is true when Y_k satisfies condition c for all k , $m \leq k \leq n$, (by convention $Y_{m:n}c$ will be true for $m > n$) and by $\#(Y_{m:n}c)$ the number of indices k , $m \leq k \leq n$, for which Y_k satisfies condition c . To simplify the method, we will assume Λ slightly larger than $\max_{i \in \Omega} \lambda_i = \max_{i \in S} \lambda_i$ (i.e. $\Lambda = (1 + \theta) \max_{i \in S} \lambda_i$, where θ is a small quantity > 0). This implies $P_{i,i} > 0$, $i \in S$. Also, since it has been assumed $\lambda_{r,S'} > 0$, we have $P_{r,S'} > 0$.

2.1 Transformed model with infinite state space

We will start by introducing two DTMCs Z, Z' . It will turn out that the transition rates of V can be expressed in terms of the transient regimes of Z and Z' . Furthermore, the benign behavior of the method will depend on the transient nature of some states of Z and Z' .

The DTMC $Z = \{Z_k; k = 0, 1, 2, \dots\}$ follows \widehat{X} from r till re-entry in r . Formally, considering a version of $\widehat{X}, \widehat{X}'$, with initial probability distribution concentrated in state r ,

$$\begin{aligned} Z_0 &= r, \\ Z_k &= \begin{cases} i \in S' \cup \{f_1, f_2, \dots, f_A\} & \text{if } \widehat{X}'_{1:k} \neq r \wedge \widehat{X}'_k = i \\ a & \text{if } \#(\widehat{X}'_{1:k} = r) > 0 \end{cases}, \quad k > 0. \end{aligned} \quad (3)$$

The DTMC Z has state space $S \cup \{f_1, f_2, \dots, f_A, a\}$, and its (possibly) nonnull transition probabilities are:

$$P[Z_{k+1} = j \mid Z_k = i] = P_{i,j}, \quad i \in S, j \in S' \cup \{f_1, f_2, \dots, f_A\}, \quad (4)$$

$$P[Z_{k+1} = a \mid Z_k = i] = P_{i,r}, \quad i \in S, \quad (5)$$

$$P[Z_{k+1} = f_i \mid Z_k = f_i] = P[Z_{k+1} = a \mid Z_k = a] = 1, \quad 1 \leq i \leq A. \quad (6)$$

States f_1, \dots, f_A, a are absorbing in Z . Also, because of the assumed properties for X , it can be easily checked that there is a path in Z from every state $i \in S$ to an absorbing state and, therefore, that all states in S are transient in Z .

The second transient DTMC, $Z' = \{Z'_k; k = 0, 1, 2, \dots\}$, follows \widehat{X} until its first visit to state r . Formally, Z' is defined as

$$Z'_k = \begin{cases} i \in S' \cup \{f_1, f_2, \dots, f_A\} & \text{if } \widehat{X}_{0:k} \neq r \wedge \widehat{X}_k = i \\ a & \text{if } \#(\widehat{X}_{0:k} = r) > 0 \end{cases}. \quad (7)$$

The DTMC Z' has state space $S' \cup \{f_1, f_2, \dots, f_A, a\}$, initial probability distribution $P[Z'_0 = i] = \alpha_i$, $i \in S' \cup \{f_1, f_2, \dots, f_A\}$, $P[Z'_0 = a] = \alpha_r$, and (possibly) nonnull transition probabilities:

$$P[Z'_{k+1} = j \mid Z'_k = i] = P_{i,j}, \quad i \in S', j \in S' \cup \{f_1, f_2, \dots, f_A\}, \quad (8)$$

$$P[Z'_{k+1} = a \mid Z'_k = i] = P_{i,r}, \quad i \in S', \quad (9)$$

$$P[Z'_{k+1} = f_i \mid Z'_k = f_i] = P[Z'_{k+1} = a \mid Z'_k = a] = 1, \quad 1 \leq i \leq A. \quad (10)$$

States f_1, \dots, f_A, a are absorbing in Z' . Also, because of the assumed properties for X , it can be easily checked that there is a path in Z' from every state $i \in S'$ to an absorbing state and, therefore, that all states in S' are transient in Z' .

Let $\pi_i(k) = P[Z_k = i]$, $\pi'_i(k) = P[Z'_k = i]$ and consider the row vectors $\boldsymbol{\pi}(k) = (\pi_i(k))_{i \in S}$ and $\boldsymbol{\pi}'(k) = (\pi'_i(k))_{i \in S'}$. Let \mathbf{P}_Z be the transition probability matrix of Z restricted to $S \times S$. Let $\mathbf{P}_{Z'}$ be the transition probability matrix of Z' restricted to $S' \times S'$. Vector $\boldsymbol{\pi}(0)$ has components $\pi_r(0) = 1$, $\pi_i(0) = 0$, $i \in S'$. From $\boldsymbol{\pi}(0)$, $\boldsymbol{\pi}(k)$, $k > 0$ can be obtained using

$$\boldsymbol{\pi}(k+1) = \boldsymbol{\pi}(k)\mathbf{P}_Z. \quad (11)$$

Vector $\boldsymbol{\pi}'(0)$ has components $\pi'_i(0) = \alpha_i$, $i \in S'$. From $\boldsymbol{\pi}'(0)$, $\boldsymbol{\pi}'(k)$, $k > 0$ can be obtained using

$$\boldsymbol{\pi}'(k+1) = \boldsymbol{\pi}'(k)\mathbf{P}_{Z'}. \quad (12)$$

The transformed CTMC model is the derandomized CTMC $V = \{V(t); t \geq 0\}$ with randomization rate Λ of the DTMC $\widehat{V} = \{\widehat{V}_k; k = 0, 1, 2, \dots\}$ defined from \widehat{X} as

$$\widehat{V}_k = \begin{cases} s_l & \text{if } 0 \leq l \leq k \wedge \widehat{X}_{k-l} = r \wedge \widehat{X}_{k-l+1:k} \in S' \\ s'_k & \text{if } \widehat{X}_{0:k} \in S' \\ f_i & \text{if } \widehat{X}_k = f_i \end{cases}. \quad (13)$$

In words, $\widehat{V}_k = s_l$ if, by step k , \widehat{X} has not left S , has made some visit to r , and the last visit to r was at step $k-l$; $\widehat{V}_k = s'_k$ if, by step k , \widehat{X} has not left S' ; and $\widehat{V}_k = f_i$ if, by step k , \widehat{X} has been absorbed into state f_i . Note that $\widehat{V}_k = s_0$ if and only if $\widehat{X}_k = r$. Informally, states s_l , $l \geq 0$ characterize the behavior of \widehat{X} from r until next hit of r or a state f_i and states s'_l , $l \geq 0$ characterize the behavior of \widehat{X} from S' up to state r or a state f_i .

Let $a(l) = \sum_{i \in S} \pi_i(l)$ and let $a'(l) = \sum_{i \in S'} \pi'_i(l)$. Note that, being $P_{r,S'} > 0$ and $P_{i,i} > 0$, $i \in S'$, $a(l) > 0$, $l \geq 0$ and, for $\alpha_{S'} > 0$, $a'(l) > 0$, $l \geq 0$. The following proposition states formally that \widehat{V} is a DTMC and gives its initial probability distribution and transition probabilities for the case $\alpha_{S'} > 0$. Note that, since $a(l) > 0$, $l \geq 0$ and, for $\alpha_{S'} > 0$, $a'(l) > 0$, $l \geq 0$, there are not divisions by 0.

Proposition 1. Assume $\alpha_{S'} > 0$. Let $v_l^j = \sum_{i \in S} \pi_i(l) P_{i,f_j} / a(l)$, $q_l = \sum_{i \in S} \pi_i(l) P_{i,r} / a(l)$, $w_l = \sum_{i \in S} \pi_i(l) P_{i,S'} / a(l)$, $v_l^{j,j} = \sum_{i \in S'} \pi'_i(l) P_{i,f_j} / a'(l)$, $q'_l = \sum_{i \in S'} \pi'_i(l) P_{i,r} / a'(l)$, $w'_l = \sum_{i \in S'} \pi'_i(l) P_{i,S'} / a'(l)$. Then, \widehat{V} is a DTMC with state space $\{s_0, s_1, \dots\} \cup \{s'_0, s'_1, \dots\} \cup \{f_1, f_2, \dots, f_A\}$, initial probability distribution $P[\widehat{V}_0 = s_0] = \alpha_r$, $P[\widehat{V}_0 = s'_0] = \alpha_{S'}$, $P[\widehat{V}_0 = f_i] = \alpha_{f_i}$, $P[\widehat{V}_0 = i] = 0$, $i \notin \{s_0, s'_0, f_1, f_2, \dots, f_A\}$, and (possibly) nonnull transition probabilities $P[\widehat{V}_{k+1} = f_j \mid \widehat{V}_k = s_l] = v_l^j$, $P[\widehat{V}_{k+1} = s_0 \mid \widehat{V}_k = s_l] = q_l$, $P[\widehat{V}_{k+1} = s_{l+1} \mid \widehat{V}_k = s_l] = w_l$, $P[\widehat{V}_{k+1} = f_j \mid \widehat{V}_k = s'_l] = v_l^{j,j}$, $P[\widehat{V}_{k+1} = s_0 \mid \widehat{V}_k = s'_l] = q'_l$, $P[\widehat{V}_{k+1} = s'_{l+1} \mid \widehat{V}_k = s'_l] = w'_l$, and $P[\widehat{V}_{k+1} = f_i \mid \widehat{V}_k = f_i] = 1$ (the state transition diagram of \widehat{V} is illustrated in Figure 1 for the case $A = 1$).

Proof. See the Appendix. □

In the case $\alpha_{S'} = 0$, \widehat{V} has state space $\{s_0, s_1, \dots\} \cup \{f_1, f_2, \dots, f_A\}$, initial probability distribution $P[\widehat{V}_0 = s_0] = \alpha_r = \alpha_S$, $P[\widehat{V}_0 = f_i] = \alpha_{f_i}$, $P[\widehat{V}_0 = i] = 0$, $i \notin \{s_0, f_1, f_2, \dots, f_A\}$ and identical state transition diagram as for the case $\alpha_{S'} > 0$ except for the absence of states s'_0, s'_1, \dots

The CTMC V has same state space and initial probability distribution as \widehat{V} . Its state transition diagram is illustrated in Figure 2 for the case $\alpha_{S'} > 0$, $A = 1$. In the case $\alpha_{S'} = 0$, states s'_0, s'_1, \dots

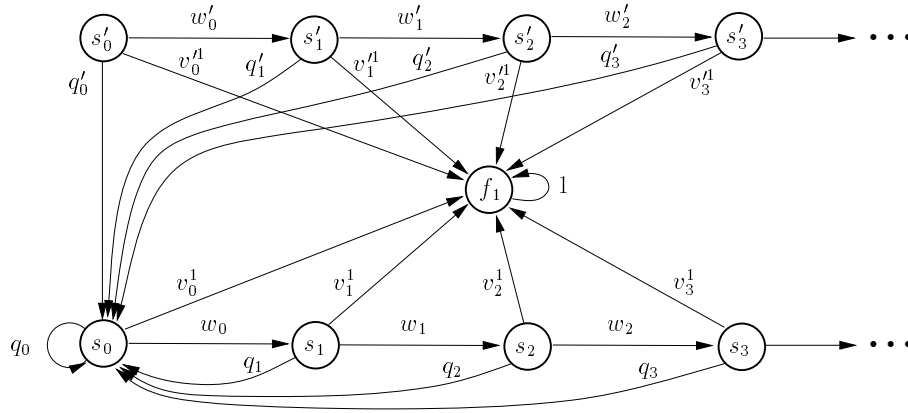


Figure 1: State transition diagram of the DTMC \widehat{V} for the case $\alpha_{S'} > 0, A = 1$.

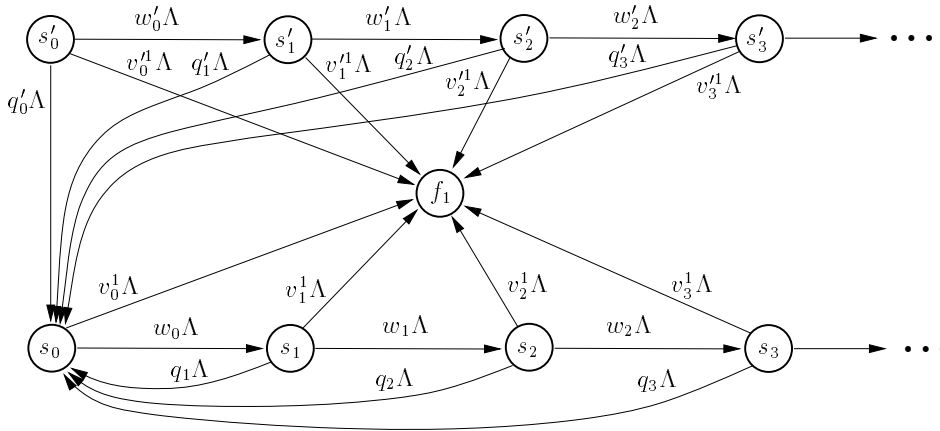


Figure 2: State transition diagram of the CTMC V for the case $\alpha_{S'} > 0, A = 1$.

would be absent. The following theorem establishes that $m(t)$ can be expressed exactly in terms of the transient regime of V .

Theorem 1. $m(t) = \sum_{i=1}^A r_{f_i} P[V(t) = f_i]$.

Proof. Note that (13) $\widehat{V}_k = f_i$ if and only if $\widehat{X}_k = f_i$. Then, using the probabilistic identity of $X = \{X(t); t \geq 0\}$ and $\{\widehat{X}_{Q(t)}; t \geq 0\}$ on one hand and of $V = \{V(t); t \geq 0\}$ and $\{\widehat{V}_{Q(t)}; t \geq 0\}$ on the other hand, where Q is a Poisson process with arrival rate Λ independent of both \widehat{X} and \widehat{V} ,

$$\begin{aligned} m(t) &= \sum_{i=1}^A r_{f_i} P[X(t) = f_i] = \sum_{i=1}^A r_{f_i} \sum_{k=0}^{\infty} P[\widehat{X}_k = f_i] P[Q(t) = k] \\ &= \sum_{i=1}^A r_{f_i} \sum_{k=0}^{\infty} P[\widehat{V}_k = f_i] P[Q(t) = k] = \sum_{i=1}^A r_{f_i} P[V(t) = f_i]. \quad \square \end{aligned}$$

2.2 Truncation of the transformed model

In this section we will show how V can be truncated to obtain a CTMC model with finite state space yielding the measure $m(t)$ with some arbitrarily small error and will derive computationally inexpensive upper bounds for the resulting model truncation error.

For the case $\alpha_{S'} > 0$, the truncated CTMC is called $V_{K,L}$ and is obtained from V by keeping the states s_k up to s_K , $K \geq 1$ and the states s'_k up to s'_L , $L \geq 1$ and directing to an absorbing state a the transitions rates from states s_K and s'_L . The initial probability distribution of $V_{K,L}$ is the same as that of V and its state transition diagram is illustrated in Figure 3 for the case $A = 1$. Formally, $V_{K,L}$ can be defined from V as

$$V_{K,L}(t) = \begin{cases} V(t) & \text{if, by time } t, V \text{ has not exited state } s_K \text{ or state } s'_L, \\ a & \text{otherwise.} \end{cases} \quad (14)$$

For the case $\alpha_{S'} = 0$, the truncated CTMC is called V_K and is obtained from V by keeping the states s_k up to s_K , $K \geq 1$ and directing to an absorbing state a the transition rates from s_K . The initial probability distribution of V_K is the same as that of V and its state transition diagram is the same as that of $V_{K,L}$ without the upper part, corresponding to the states s'_k , $0 \leq k \leq L$. Formally, V_K can be defined from V as

$$V_K(t) = \begin{cases} V(t) & \text{if, by time } t, V \text{ has not exited state } s_K, \\ a & \text{otherwise.} \end{cases}$$

For the case $\alpha_{S'} > 0$, an approximate value for $m(t)$ can be obtained in terms of the transient regime of $V_{K,L}$ as:

$$m_{K,L}^a(t) = \sum_{i=1}^A r_{f_i} P[V_{K,L}(t) = f_i]. \quad (15)$$

For the case $\alpha_{S'} = 0$, an approximate value for $m(t)$ is given by:

$$m_K^a(t) = \sum_{i=1}^A r_{f_i} P[V_K(t) = f_i]. \quad (16)$$

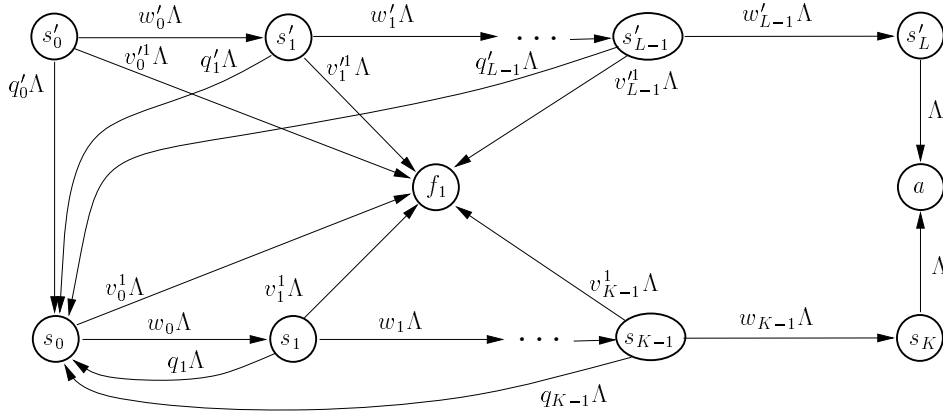


Figure 3: State transition diagram of the CTMC $V_{K,L}$ for the case $A = 1$.

The following proposition gives upper bounds for the model truncation error in terms of the transient regimes of $V_{K,L}$ and V_K ($r_{\max} = \max_{1 \leq i \leq A} r_{f_i}$).

Proposition 2. For $\alpha_{S'} > 0$, $0 \leq m(t) - m_{K,L}^a(t) \leq r_{\max} P[V_{K,L}(t) = a] = m_{K,L}^e(t)$. For the case $\alpha_{S'} = 0$, $0 \leq m(t) - m_K^a(t) \leq r_{\max} P[V_K(t) = a] = m_K^e(t)$.

Proof. Consider the case $\alpha_{S'} > 0$. Using (14), letting $A(t)$ the proposition “by time t , V has not exited state s_K or state s'_L ”, and denoting by $\overline{A}(t)$ the negated proposition of $A(t)$, we have:

$$P[V(t) = f_i] - P[V_{K,L}(t) = f_i] = P[V(t) = f_i] - P[V(t) = f_i \wedge A(t)] = P[V(t) = f_i \wedge \overline{A}(t)],$$

implying

$$P[V(t) = f_i] - P[V_{K,L}(t) = f_i] \geq 0$$

and

$$\sum_{i=1}^A (P[V(t) = f_i] - P[V_{K,L}(t) = f_i]) \leq P[\overline{A}(t)] = P[V_{K,L}(t) = a].$$

Since

$$\begin{aligned} m(t) - m_{K,L}^a(t) &= \sum_{i=1}^A r_{f_i} P[V(t) = f_i] - \sum_{i=1}^A r_{f_i} P[V_{K,L}(t) = f_i] \\ &= \sum_{i=1}^A r_{f_i} (P[V(t) = f_i] - P[V_{K,L}(t) = f_i]) \end{aligned}$$

and $0 \leq r_{f_i} \leq r_{\max}$, $1 \leq i \leq A$, we have:

$$\begin{aligned} 0 &\leq m(t) - m_{K,L}^a(t) \leq r_{\max} \sum_{i=1}^A (P[V(t) = f_i] - P[V_{K,L}(t) = f_i]) \\ &\leq r_{\max} P[V_{K,L}(t) = a] = m_{K,L}^e(t). \end{aligned}$$

The result for the case $\alpha_{S'} = 0$ can be proved similarly. \square

There does not seem to exist any specially efficient way of computing the probabilities $P[V_{K,L}(t) = a]$ and $P[V_K(t) = a]$ and, then, use of the upper bounds for the model truncation error given by Proposition 2 could be relatively expensive. In the remaining of this section, we will obtain upper bounds for $m_{K,L}^e(t)$ and $m_K^e(t)$ which can be computed quite inexpensively. The regenerative randomization method will use those upper bounds to control the model truncation error. Use of those looser upper bounds may result in an increase of the model truncation parameter K . However, as we shall illustrate in Section 4, for class C models, the increase seems to be very small.

We will start by obtaining some simple relationships. Using Proposition 1, taking into account that, according to (4)–(6), Z can only enter S' from states $i \in S$ and, therefore (4), $\sum_{i \in S'} \pi_i(k+1) = \sum_{i \in S} \pi_i(k) P_{i,S'}$, and using the fact that $\pi_r(k) = 0$ for $k > 0$, we have, for $k \geq 0$,

$$w_k = \frac{\sum_{i \in S} \pi_i(k) P_{i,S'}}{a(k)} = \frac{\sum_{i \in S'} \pi_i(k+1)}{a(k)} = \frac{\sum_{i \in S} \pi_i(k+1)}{a(k)} = \frac{a(k+1)}{a(k)}.$$

and, using $a(0) = 1$,

$$\prod_{i=0}^{k-1} w_i = \prod_{i=0}^{k-1} \frac{a(i+1)}{a(i)} = \frac{a(k)}{a(0)} = a(k). \quad (17)$$

Similarly, assuming $\alpha_{S'} > 0$, which implies $a'(k) > 0$, $k \geq 0$, using Proposition 1, taking into account that, according to (8)–(10), Z' can only enter S' from states $i \in S'$ and, therefore (8), $\sum_{i \in S'} \pi'_i(k+1) = \sum_{i \in S'} \pi'_i(k) P_{i,S'}$, for $k \geq 0$,

$$w'_k = \frac{\sum_{i \in S'} \pi'_i(k) P_{i,S'}}{a'(k)} = \frac{\sum_{i \in S'} \pi'_i(k+1)}{a'(k)} = \frac{a'(k+1)}{a'(k)},$$

and, using $a'(0) = \alpha_{S'}$,

$$\prod_{i=0}^{k-1} w'_i = \prod_{i=0}^{k-1} \frac{a'(i+1)}{a'(i)} = \frac{a'(k)}{a'(0)} = \frac{a'(k)}{\alpha_{S'}}. \quad (18)$$

Next, we will consider the randomized DTMCs with randomization rate Λ of the truncated models $V_{K,L}$ and V_K . For the case $\alpha_{S'} > 0$, let $\widehat{V}_{K,L} = \{(\widehat{V}_{K,L})_k; k = 0, 1, \dots\}$ be the randomized DTMC of $V_{K,L}$ (its state transition diagram is illustrated in Figure 4 for the case $A = 1$). For the case $\alpha_{S'} = 0$, let $\widehat{V}_K = \{(\widehat{V}_K)_k; k = 0, 1, \dots\}$ be the randomized DTMC of V_K (its state transition diagram is the same as that of $\widehat{V}_{K,L}$ but without the states s'_k , $0 \leq k \leq L$).

The quantity $m_{K,L}^e(t)$ can be decomposed as $m_L^{e'}(t) + m_{K,L}^{e''}(t)$, where $m_L^{e'}(t)$ is r_{\max} times the probability that, by time t , $V_{K,L}$ has entered a through s'_L and $m_{K,L}^{e''}(t)$ is r_{\max} times the probability that, by time t , $V_{K,L}$ has entered a through s_K . The term $m_L^{e'}(t)$ can be easily computed using the probabilistic identity of $V_{K,L} = \{V_{K,L}(t); t \geq 0\}$ and $\{(\widehat{V}_{K,L})_Q(t); t \geq 0\}$, where Q is a Poisson process with arrival rate Λ independent of $\widehat{V}_{K,L}$, by noting that the only path to a of $\widehat{V}_{K,L}$ with nonnull probability which enters a through s'_L is $(s'_0, s'_1, \dots, s'_L, a)$ and that path has probability $\alpha_{S'} (\prod_{i=0}^{L-1} w'_i)$. Using (18):

$$m_L^{e'}(t) = r_{\max} \alpha_{S'} \left(\prod_{i=0}^{L-1} w'_i \right) P[Q(t) \geq L+1] = r_{\max} a'(L) \sum_{k=L+1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}. \quad (19)$$

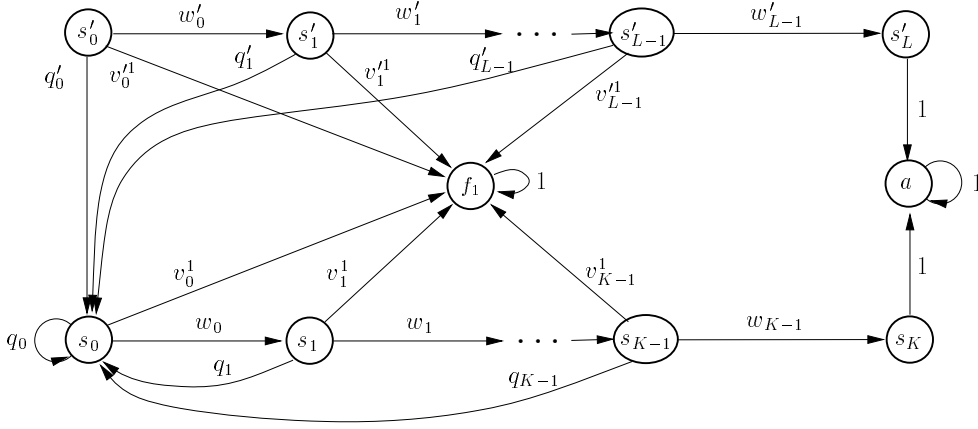


Figure 4: State transition diagram of the DTMC $\widehat{V}_{K,L}$ for the case $A = 1$.

Let $c_{K,L}(k)$ be the probability that, by step k , $\widehat{V}_{K,L}$ has entered a through s_K and let $c_K(k) = P[(\widehat{V}_K)_k = a]$. Note that, using the probabilistic identity of $V_{K,L}$ and $\{(\widehat{V}_{K,L})_{Q(t)}; t \geq 0\}$, where Q is a Poisson process with arrival rate Λ independent of $\widehat{V}_{K,L}$,

$$m_{K,L}^{e''}(t) = r_{\max} \sum_{k=0}^{\infty} c_{K,L}(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}, \quad (20)$$

and, using the probabilistic identity of V_K and $\{(\widehat{V}_K)_{Q(t)}; t \geq 0\}$, where Q is a Poisson process with arrival rate Λ independent of \widehat{V}_K ,

$$m_K^e(t) = r_{\max} \sum_{k=0}^{\infty} c_K(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}. \quad (21)$$

The exact values of $c_{K,L}(k)$ and $c_K(k)$ are difficult to compute. The following proposition gives inexpensive upper bounds for those quantities (I_c denotes the indicator function returning value 1 when condition 1 is satisfied and value 0 otherwise).

Proposition 3. For the case $\alpha_{S'} > 0$, $c_{K,L}(k) \leq I_{k>K} \alpha_S (k - K) a(K)$. For the case $\alpha_{S'} = 0$, $c_K(k) \leq I_{k>K} \alpha_S (k - K) a(K)$.

Proof. See the Appendix. □

Using (19)-(21) and Proposition 3, it is relatively simple to obtain computationally inexpensive upper bounds for $m_{K,L}^e(t)$ and $m_K^e(t)$ in terms of $a'(L)$ and $a(K)$:

Theorem 2. For the case $\alpha_{S'} > 0$, $m_{K,L}^e(t) \leq r_{\max} a'(L) \sum_{k=L+1}^{\infty} e^{-\Lambda t} (\Lambda t)^k / k! + r_{\max} \alpha_S a(K) \sum_{k=K+1}^{\infty} (k - K) e^{-\Lambda t} (\Lambda t)^k / k!$. For the case $\alpha_{S'} = 0$, $m_K^e(t) \leq r_{\max} \alpha_S a(K) \sum_{k=K+1}^{\infty} (k - K) e^{-\Lambda t} (\Lambda t)^k / k!$.

Proof. We consider first the case $\alpha_{S'} > 0$. Using (20) and Proposition 3,

$$\begin{aligned} m_{K,L}^{e''}(t) &= r_{\max} \sum_{k=0}^{\infty} c_{K,L}(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} \leq r_{\max} \sum_{k=K+1}^{\infty} \alpha_S(k-K) a(K) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} \\ &= r_{\max} \alpha_S a(K) \sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}, \end{aligned}$$

and the result follows using $m_{K,L}^e(t) = m_L^e(t) + m_{K,L}^{e''}(t)$ and (19). For the case $\alpha_{S'} = 0$, using (21) and Proposition 3,

$$\begin{aligned} m_K^e(t) &= r_{\max} \sum_{k=0}^{\infty} c_K(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} \leq r_{\max} \sum_{k=K+1}^{\infty} \alpha_S(k-K) a(K) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} \\ &= r_{\max} \alpha_S a(K) \sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}. \quad \square \end{aligned}$$

Being the states in S transient in Z , $\pi_i(k)$, $i \in S$ decrease geometrically fast with k [31, Theorem 4.3], implying that $a(k) = \sum_{i \in S} \pi_i(k)$ also decreases geometrically fast with k . Similarly, being the states in S' transient in Z' , $\pi'_i(k)$, $i \in S'$ decrease geometrically fast with k , implying that $a'(k)$ also decreases geometrically fast with k . Then, the upper bounds for $m_{K,L}^e(t)$ and $m_K^e(t)$ given by Theorem 2 can be made arbitrarily small by choosing large enough values of K and L (see proof of Theorem 3 for a rigorous justification), and the model truncation error can be made arbitrarily small.

Notice that $\sum_{k=L+1}^{\infty} e^{-\Lambda t} (\Lambda t)^k / k!$ is the probability that in the time interval $[0, t]$ there have been more than L arrivals in a Poisson process with arrival rate Λ and, therefore, $\sum_{k=L+1}^{\infty} e^{-\Lambda t} (\Lambda t)^k / k!$ is increasing with t . Regarding $\sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t} (\Lambda t)^k / k!$, we can write

$$\sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} = \sum_{k=K+1}^{\infty} \sum_{i=K+1}^k e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} = \sum_{i=K+1}^{\infty} \sum_{k=i}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^k}{k!},$$

and, since each term $\sum_{k=i}^{\infty} e^{-\Lambda t} (\Lambda t)^k / k!$ is increasing with t , $\sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t} (\Lambda t)^k / k!$ is also increasing with t . Then, the upper bounds for $m_{K,L}^e(t)$ and $m_K^e(t)$ given by Theorem 2 are both increasing with t .

2.3 Algorithmic Description and Discussion

An algorithmic description of the regenerative randomization method is given in Figure 5. The algorithm has as inputs the CTMC X , the number A of absorbing states f_i , the reward rates $r_{f_i} \geq 0$, $1 \leq i \leq A$, an initial probability distribution row vector $\alpha = (\alpha_i)_{i \in \Omega}$, the regenerative state r , the allowed error ε , the number of time points n at which $m(t)$ has to be computed, and the time points t_1, t_2, \dots, t_n . The algorithm has as outputs the estimates for $m(t)$, $\tilde{m}(t_1), \tilde{m}(t_2), \dots, \tilde{m}(t_n)$. The algorithm uses $\theta = 10^{-4}$ and, therefore, Λ is taken equal to $(1 + 10^{-4}) \max_{i \in S} \lambda_i$. Of the allowed error ε , a portion $\varepsilon/2$ is allocated for the error associated with the truncation of the transformed model and a portion $\varepsilon/2$ is allocated for the error associated with the solution of the truncated transformed model by standard randomization. Since the upper bounds for $m_{K,L}^e(t)$ and $m_K^e(t)$ given by Theorem 2 increase with t , the error associated with the truncation of V is controlled

for $t_{\max} = \max\{t_1, t_2, \dots, t_n\}$. For the case $\alpha_{S'} > 0$, the error allocated for the truncation of V , $\varepsilon/2$, is divided equally between the contributions $r_{\max} a'(L) \sum_{k=L+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k!$ and $r_{\max} \alpha_{S'} a(K) \sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k!$ to the upper bound for that error given by Theorem 2. The error upper bound associated with the solution of the truncated transformed model by standard randomization, $r_{\max} \sum_{k=N+1}^{\infty} e^{-\Lambda t} (\Lambda t)^k / k!$, where N is the truncation point, also increases with t , and, therefore, that error is also controlled for t_{\max} . Using the probabilistic identity between $V_{K,L}$ and $\{(\widehat{V}_{K,L})_{Q(t)}; t \geq 0\}$, where $Q = \{Q(t); t \geq 0\}$ is a Poisson process with arrival rate Λ independent of $\widehat{V}_{K,L}$, and (15), for the case $\alpha_{S'} > 0$, the estimates for $m(t)$, $\tilde{m}(t)$, are computed using

$$\tilde{m}(t) = \sum_{i=1}^A r_{f_i} \sum_{k=0}^N P[(\widehat{V}_{K,L})_k = f_i] e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} = \sum_{k=0}^N d(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!},$$

with $d(k) = \sum_{i=1}^A r_{f_i} P[(\widehat{V}_{K,L})_k = f_i]$. Similarly, using the probabilistic identity between V_K and $\{(\widehat{V}_K)_{Q(t)}; t \geq 0\}$, where $Q = \{Q(t); t \geq 0\}$ is a Poisson process with arrival rate Λ independent of \widehat{V}_K , and (16), for the case $\alpha_{S'} = 0$, the estimates for $m(t)$, $\tilde{m}(t)$, are computed using

$$\tilde{m}(t) = \sum_{k=0}^N d(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!},$$

with $d(k) = \sum_{i=1}^A r_{f_i} P[(\widehat{V}_K)_k = f_i]$.

We deal next with some low-level implementation details not explicitly shown in Figure 5. The method requires the computation of $S(m) = \sum_{k=m+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k!$ for increasing values of m . This could be done using

$$S(m) = 1 - \sum_{k=0}^m e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!},$$

but this is numerically unstable when the requested error ε is very small. In our implementation of regenerative randomization we use, instead, the algorithm described next, which is much more efficient and numerically stable.

Assuming $M + 1 > \Lambda t_{\max}$, which implies $0 < \Lambda t_{\max} / (M + 1) < 1$, and $M \geq m + 2$, we have

$$\begin{aligned} S(m) &= \sum_{k=m+1}^{M-1} e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!} + \sum_{k=M}^{\infty} e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!} \\ &< \sum_{k=m+1}^{M-1} e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!} + e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^M}{M!} \sum_{k=0}^{\infty} \left(\frac{\Lambda t_{\max}}{M+1} \right)^k \\ &= \tilde{S}(m, M) + S^e(M), \end{aligned}$$

with

$$\tilde{S}(m, M) = \sum_{k=m+1}^{M-1} e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!},$$

Inputs: $X, A, r_{f_1}, r_{f_2}, \dots, r_{f_A}, \boldsymbol{\alpha}, r, \varepsilon, n, t_1, t_2, \dots, t_n$
Outputs: $\tilde{m}(t_1), \tilde{m}(t_2), \dots, \tilde{m}(t_n)$
 $r_{\max} = \max\{r_{f_1}, r_{f_2}, \dots, r_{f_A}\};$
 $t_{\max} = \max\{t_1, t_2, \dots, t_n\};$
 $\Lambda = (1 + 10^{-4}) \max_{i \in S} \lambda_i;$
 Obtain \mathbf{P} ;
 for $(i \in S)$ $P_{i,S'} = \sum_{j \in S', P_{i,j} \neq 0} P_{i,j}; \alpha_{S'} = \sum_{i \in S'} \alpha_i; \alpha_S = \alpha_r + \alpha_{S'};$
 if $(\alpha_{S'} > 0)$ $tol_K = \varepsilon/4;$ else $tol_K = \varepsilon/2;$
 $\boldsymbol{\pi} = (I_{i=r})_{i \in S}; a = 1; K = 0;$
 do {
 for $(j = 1; j \leq A; j++)$ $v_K^j = \sum_{i \in S, P_{i,j} \neq 0} \pi_i P_{i,j} / a;$
 $q_K = \sum_{i \in S, P_{i,r} \neq 0} \pi_i P_{i,r} / a; w_K = \sum_{i \in S} \pi_i P_{i,S'} / a;$
 $\mathbf{n}\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}_Z; \boldsymbol{\pi} = \mathbf{n}\boldsymbol{\pi};$
 $K++;$
 $a = \sum_{i \in S} \pi_i;$
 }
 until $(r_{\max} \alpha_S a \sum_{k=K+1}^{\infty} (k - K) e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k! \leq tol_K);$
 if $(\alpha_{S'} > 0)$ {
 $\boldsymbol{\pi}' = (\alpha_i)_{i \in S'}; a' = \alpha_{S'}; L = 0;$
 do {
 for $(j = 1; j \leq A; j++)$ $v_L^j = \sum_{i \in S', P_{i,j} \neq 0} \pi'_i P_{i,j} / a';$
 $q_L = \sum_{i \in S', P_{i,r} \neq 0} \pi'_i P_{i,r} / a'; w_L = \sum_{i \in S'} \pi'_i P_{i,S'} / a';$
 $\mathbf{n}\boldsymbol{\pi}' = \boldsymbol{\pi}'\mathbf{P}_{Z'}; \boldsymbol{\pi}' = \mathbf{n}\boldsymbol{\pi}';$
 $L++;$
 $a' = \sum_{i \in S'} \pi'_i;$
 }
 until $(r_{\max} a' \sum_{k=L+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k! \leq \varepsilon/4);$
 }
 $N = \min\{m \geq 0 : r_{\max} \sum_{l=m+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^l / l! \leq \varepsilon/2\};$
 if $(\alpha_{S'} > 0)$
 Give N steps to $\widehat{V}_{K,L}$ and compute $d(k) = \sum_{i=1}^A r_{f_i} P[(\widehat{V}_{K,L})_k = f_i], k = 0, 1, \dots, N;$
 else
 Give N steps to \widehat{V}_K and compute $d(k) = \sum_{i=1}^A r_{f_i} P[(\widehat{V}_K)_k = f_i], k = 0, 1, \dots, N;$
 for $(i = 1; i \leq n; i++)$
 for $(k = 0, \tilde{m}(t_i) = 0; k \leq N; k++)$ $\tilde{m}(t_i) += d(k) e^{-\Lambda t_i} (\Lambda t_i)^k / k!;$

Figure 5: Algorithmic description of regenerative randomization.

$$\begin{aligned}
S^e(M) &= e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^M}{M!} \sum_{k=0}^{\infty} \left(\frac{\Lambda t_{\max}}{M+1} \right)^k \\
&= e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^M}{M!} \frac{1}{1 - \frac{\Lambda t_{\max}}{M+1}}.
\end{aligned}$$

The quantity $\tilde{S}(m, M) + S^e(M)$ can be taken as a pessimistic (larger) approximation for $S(m)$ with error upper bounded by $S^e(M)$. Then, being ν a small quantity representing the desired relative error for the computation of the $S(m)$'s (for instance, $\nu = 10^{-6}$), the algorithm starts by selecting for the initial m for which $S(m)$ has to be computed, m_0 , the smallest integer M with $M + 1 > \Lambda t_{\max}$, $M \geq m_0 + 2$ satisfying $S^e(M)/\tilde{S}(m_0, M) \leq \nu/10$, sets $S_{\text{last}} = \tilde{S}(m_0, M) + S^e(M)$, $S_{\text{old}} = S_{\text{last}}$, and approximates $S(m_0)$ with S_{last} . Then, for increasing m , if $m \leq M - 2$, the algorithm computes $S_{\text{last}} = S_{\text{last}} - e^{-\Lambda t_{\max}} (\Lambda t_{\max})^m / m!$ and, if $S_{\text{last}}/S_{\text{old}} \geq 0.1$, approximates $S(m)$ with S_{last} . If m becomes $> M - 2$ or, being $m \leq M - 2$, $S_{\text{last}}/S_{\text{old}}$ becomes < 0.1 , the algorithm obtains a new M as the smallest integer with $M + 1 > \Lambda t_{\max}$, $M \geq m + 2$ satisfying $S^e(M)/\tilde{S}(m, M) \leq \nu/10$, sets $S_{\text{last}} = \tilde{S}(m, M) + S^e(M)$, $S_{\text{old}} = S_{\text{last}}$, approximates $S(m)$ with S_{last} , and continues.

The computation of $S'(m) = \sum_{k=m+1}^{\infty} (k-m)e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k!$ for increasing values of m is also required. Our implementation of regenerative randomization uses a reasonably efficient and numerically stable algorithm similar to the algorithm used for the computation of the $S(m)$'s. Assuming $M + 1 > \Lambda t_{\max}$, which implies $0 < \Lambda t_{\max}/(M + 1) < 1$, and $M \geq m + 2$, we have:

$$\begin{aligned}
S'(m) &= \sum_{k=m+1}^{M-1} (k-m)e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!} + \sum_{k=M}^{\infty} (k-m)e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!} \\
&< \sum_{k=m+1}^{M-1} (k-m)e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!} + e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^M}{M!} \sum_{k=M}^{\infty} (k-m) \left(\frac{\Lambda t_{\max}}{M+1} \right)^{k-M} \\
&= \tilde{S}'(m, M) + S'^e(m, M),
\end{aligned}$$

with

$$\begin{aligned}
\tilde{S}'(m, M) &= \sum_{k=m+1}^{M-1} (k-m)e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^k}{k!}, \\
S'^e(m, M) &= e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^M}{M!} \sum_{k=M}^{\infty} (k-m) \left(\frac{\Lambda t_{\max}}{M+1} \right)^{k-M} \\
&= e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^M}{M!} \left[(M-m) \sum_{k=0}^{\infty} \left(\frac{\Lambda t_{\max}}{M+1} \right)^k + \sum_{k=0}^{\infty} k \left(\frac{\Lambda t_{\max}}{M+1} \right)^k \right] \\
&= e^{-\Lambda t_{\max}} \frac{(\Lambda t_{\max})^M}{M!} \left[\frac{M-m}{1 - \frac{\Lambda t_{\max}}{M+1}} + \frac{\frac{\Lambda t_{\max}}{M+1}}{\left(1 - \frac{\Lambda t_{\max}}{M+1}\right)^2} \right],
\end{aligned}$$

where it has been used $\sum_{k=0}^{\infty} ka^k = a/(1-a)^2$, $0 < a < 1$ (see, for instance, [36, Formula 19.7]). The quantity $\tilde{S}'(m, M) + S'^e(m, M)$ can be taken as a pessimistic approximation for $S'(m)$ with error upper bounded by $S'^e(m, M)$. It is easy to check that $\tilde{S}'(m, M) = \tilde{S}'(m-1, M) - \tilde{S}(m-1, M)$. Also, $S'^e(m, M)$ decreases with m . Those observations justify the following algorithm,

where ν is the desired relative error for the computation of $S'(m)$. For the initial m for which $S'(m)$ has to be computed, m_0 , the algorithm obtains the smallest integer M with $M + 1 > \Lambda t_{\max}$, $M \geq m_0 + 2$ satisfying $S'^e(m_0, M)/\tilde{S}'(m_0, M) \leq \nu/10$, sets $S'_{\text{last}} = \tilde{S}'(m_0, M) + S'^e(m_0, M)$, $S'_{\text{old}} = S'_{\text{last}}$, $S_{\text{last}} = \tilde{S}(m_0, M)$, $S_{\text{old}} = S_{\text{last}}$, and approximates $S'(m_0)$ with S'_{last} . Then, each time m is incremented, if $m \leq M - 2$, the algorithm computes $S'_{\text{last}} = S'_{\text{last}} - S_{\text{last}}$ and, if $S'_{\text{last}}/S'_{\text{old}} \geq 0.1$, approximates $S'(m)$ with S'_{last} , and computes $S_{\text{last}} = S_{\text{last}} - e^{-\Lambda t_{\max}}(\Lambda t_{\max})^m/m!$ (so that it be $\tilde{S}(m, M)$), unless $S_{\text{last}}/S_{\text{old}}$ becomes < 0.1 , in which case it computes $S_{\text{last}} = \tilde{S}(m, M)$ and sets $S_{\text{old}} = S_{\text{last}}$. If m becomes $> M - 2$ or, being $m \leq M - 2$, $S'_{\text{last}}/S'_{\text{old}}$ becomes < 0.1 , the algorithm obtains a new M as the smallest integer with $M + 1 > \Lambda t_{\max}$, $M \geq m + 2$ satisfying $S'^e(m, M)/\tilde{S}'(m, M) \leq \nu/10$, sets $S'_{\text{last}} = \tilde{S}'(m, M) + S'^e(m, M)$, $S'_{\text{old}} = S'_{\text{last}}$, $S_{\text{last}} = \tilde{S}(m, M)$, $S_{\text{old}} = S_{\text{last}}$, approximates $S'(m)$ with S'_{last} , and continues.

We note that $P[(\widehat{V}_{K,L})_k = f_i]$ ($P[(\widehat{V}_K)_k = f_i]$) are determined, once \mathbf{P} has been computed, by adding always positive numbers smaller than 1 and, therefore, regenerative randomization has the same excellent numerical stability as standard randomization. In addition, the computation error is well-controlled and can be specified in advance. Thus, regenerative randomization has the same good properties as standard randomization.

We analyze next the memory overhead of regenerative randomization with respect to standard randomization. Given the relationships (4)–(6), (8)–(10) between the transition probabilities of, respectively, Z and Z' and the transition probabilities of \widehat{X} , it is not necessary to store \mathbf{P}_Z and $\mathbf{P}_{Z'}$ explicitly. In addition, vectors $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ and vectors $\mathbf{n}\boldsymbol{\pi}$ and $\mathbf{n}\boldsymbol{\pi}'$ can share the same storage, and a similar storage is required by standard randomization. The memory overhead of regenerative randomization with respect to standard randomization is, then, basically restricted to the space needed to store the vector of size $|S|$, $(P_{i,S'})_{i \in S}$ and the transition probabilities of $\widehat{V}_{K,L}$ (\widehat{V}_K) v_k^i , q_k , w_k , $0 \leq k \leq K - 1$, $1 \leq i \leq A$ and, if $\alpha_{S'} > 0$, v_k^i , q_k' , w_k' , $0 \leq k \leq L - 1$, $1 \leq i \leq A$.

3 Theoretical properties

As discussed in Section 1, standard randomization requires a number of steps on \widehat{X} which, for large Λt and $\varepsilon \ll 1$, is approximately equal to Λt . Regarding regenerative randomization, using the facts that all states in S are transient in Z and that all states in S' are transient in Z' , it can be shown that:

Theorem 3. *For the case $\alpha_{S'} > 0$, the number of steps K on Z and the number of steps L on Z' required in regenerative randomization are, respectively, $O(\log(\Lambda t/\varepsilon))$ and $O(\log(1/\varepsilon))$. For the case $\alpha_{S'} = 0$, the number of steps K on Z required in regenerative randomization is $O(\log(\Lambda t/\varepsilon))$.*

Proof. See the Appendix. □

Theorem 3 asserts that, contrary to standard randomization, the model truncation parameters K and L are smooth functions of Λt . That property is called “benign behavior”. A consequence of Theorem 3 is that, for large enough Λt , the number of steps on Z and Z' required in regenerative randomization will be significantly smaller than the number of steps on \widehat{X} required in standard randomization, implying that the computational cost of the first phase of regenerative randomization

(generation of the truncated transformed model) will be significantly smaller than the computational cost of standard randomization. In addition, for large enough X , the truncated transformed model will be significantly smaller than X , and, since the maximum output rate of the truncated transformed model is only slightly larger than $\max_{i \in \Omega} \lambda_i$ and, then, for large t , the truncation point N of the standard randomization method applied to the solution of the truncated transformed model would be almost identical to the truncation point N of standard randomization applied to X , the second phase of regenerative randomization (solution of the truncated transformed model by standard randomization) will have significantly smaller computational cost than standard randomization. In summary, for large enough X and Λt , the computational cost of regenerative randomization will be significantly smaller than the computational cost of standard randomization.

The computational cost of regenerative randomization depends, of course, on the selection of the regenerative state r , since that selection influences the behavior of $a(k)$ and $a'(k)$ and the required values for the truncation parameters K and L . Ideally, the state r should be chosen so that $a(k)$ and $a'(k)$ decrease as fast as possible. For as wide class of models as covered by regenerative randomization, automatic selection of r does not seem to be easy in general, and, then, the method relies on the user's intuition to select an appropriate state r . We will consider, however, a class of models, class C, for which a natural selection for the regenerative state exists, and, for models in that class and that natural selection, will obtain stronger theoretical results than the benign behavior asserted by Theorem 3 assessing the performance of regenerative randomization in terms of "visible" model characteristics.

The model class C includes all CTMCs X with the properties described in Section 1 for which a partition $S_0 \cup S_1 \cup \dots \cup S_{N_C}$ for S exists satisfying the following two properties:

P1. $S_0 = \{o\}$ (i.e. $|S_0| = 1$).

P2. $\max_{0 \leq k \leq N_C} \max_{i \in S_k} \lambda_{i, S_k - \{i\} \cup S_{k+1} \cup \dots \cup S_{N_C}}$ is significantly smaller than $\min_{0 < k \leq N_C} \min_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}} > 0$.

Class C covers failure/repair reliability models with exponential failure and repair time distributions and repair in every state with failed components when failure rates are significantly smaller than repair rates (the typical case). For those models, a partition for which properties P1 and P2 are satisfied is $S_k = \{\text{states in } S \text{ with } k \text{ failed components}\}$. To illustrate those models, Figure 6 shows a small failure/repair reliability model of a fault-tolerant system using the pair-and-spare technique [14], in which active modules have failure rate λ_M , the spare module does not fail, the failure of an active module is "soft" with probability S_M and "hard" with probability $1 - S_M$, and, whether soft or hard, the failure of an active module is covered with probability C_M . Modules in soft failure are independently recovered at rate μ_S and modules in hard failure are repaired by a single repairman at rate μ_H . The system is initially in the state with no module failed. With $r_{f_1} = 1$, the measure $m(t)$ would be the unreliability of the fault-tolerant system. Assuming λ_M much smaller than both μ_H and μ_S , a partition for $S = \{1, 2, 3, 4, 5, 6\}$ showing that the model is in class C is $S_0 = \{1\}$, $S_1 = \{2, 3\}$, $S_2 = \{4, 5, 6\}$. Class C also covers failure/repair models with exponential failure time distributions, repair times with acyclic phase-type distributions [24] (which can be used to fit distributions of non-exponential positive random variables [3]), and repair in every state with

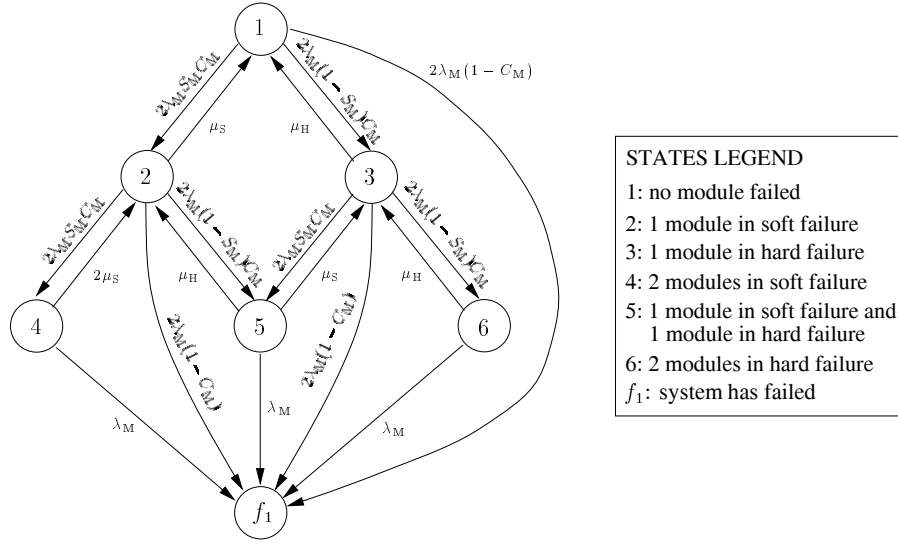


Figure 6: CTMC reliability model of a repairable fault-tolerant system using the pair-and-spare technique.

failed components, provided that the transition rates of the transient CTMCs defining the phase-type distributions are sufficiently large compared with failure rates.

Since, for class C models, X moves “fast” to either state o or an absorbing state f_i , a natural selection for the regenerative state for those models is $r = o$. That selection is possible. Consider a class C model and a partition $S_0 \cup \dots \cup S_{N_C}$ for S satisfying properties P1 and P2, and let

$$\delta = \frac{\max_{0 \leq k \leq N_C} \max_{i \in S_k} \lambda_{i, S_k - \{i\} \cup S_{k+1} \cup \dots \cup S_{N_C}}}{\min_{0 < k \leq N_C} \min_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}}}.$$

The parameter δ can be seen as a “rarity” parameter measuring how small the transition rates $\lambda_{i,j}$, $i \in S_k$, $j \in S_k - \{i\} \cup S_{k+1} \cup \dots \cup S_{N_C}$, $0 \leq k \leq N_C$ are compared to $\min_{0 < k \leq N_C} \min_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}}$. In terms of the rarity parameter δ , we can model the transition rates $\lambda_{i,j}$, $i \in S_k$, $j \in S_k - \{i\} \cup S_{k+1} \cup \dots \cup S_{N_C}$, $0 \leq k \leq N_C$ as $\lambda_{i,j} = \Lambda_{i,j} \delta$, where $\Lambda_{i,j}$ are constants satisfying $\sum_{j \in S_k - \{i\} \cup S_{k+1} \cup \dots \cup S_{N_C}} (\Lambda_{i,j} / \min_{0 < k \leq N_C} \min_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}}) \leq 1$ and study the behavior of $a(k)$ and $a'(k)$ with the selection $r = o$ as $\delta \rightarrow 0$. Since δ is small, the actual behavior should be close to that limit behavior. Let $P_{i,j}(\delta)$ denote the transition probabilities of \hat{X} as a function of the rarity parameter δ and let $\mathbf{P}_Z(\delta)$ and $\mathbf{P}_{Z'}(\delta)$ denote, respectively, the matrices \mathbf{P}_Z and $\mathbf{P}_{Z'}$ as a function of δ . Note that, for $i \in S_k$, $j \in S_k - \{i\} \cup S_{k+1} \cup \dots \cup S_{N_C}$, $0 \leq k \leq N_C$, $\lim_{\delta \rightarrow 0} P_{i,j}(\delta) = 0$ and that

$$\lim_{\delta \rightarrow 0} P_{i,i}(\delta) = 1 - \frac{\lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}}}{(1 + \theta) \max_{0 \leq k \leq N_C} \max_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}}}, i \in S_k, 0 < k \leq N_C.$$

We have the following result: ³

Theorem 4. For class C models and the selection $r = o$, $a(k) \leq h(k)$ and $a'(k) \leq \alpha_S h'(k)$, where, for $k \rightarrow \infty$, $h(k) \sim B(\delta) \binom{k}{p(\delta)-1} \rho(\mathbf{P}_Z(\delta)^T)^k$, $B(\delta) > 0$, $p(\delta)$ integer ≥ 1 and

³ $x(k) \sim y(k)$ for $k \rightarrow \infty$ denotes $\lim_{k \rightarrow \infty} x(k)/y(k) = 1$.

$h'(k) \sim B'(\delta) \binom{k}{p'(\delta)-1} \rho(\mathbf{P}_{Z'}(\delta)^T)^k$, $B'(\delta) > 0$, $p'(\delta)$ integer ≥ 1 , with $\lim_{\delta \rightarrow 0} \rho(\mathbf{P}_Z(\delta)^T) = \lim_{\delta \rightarrow 0} \rho(\mathbf{P}_{Z'}(\delta)^T) = q$,

$$q = 1 - \frac{\min_{0 < k \leq N_C} \min_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}}}{(1 + \theta) \max_{0 \leq k \leq N_C} \max_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}}}.$$

Proof. From (11), $\boldsymbol{\pi}(k)^T = \mathbf{P}_Z(\delta)^{Tk} \boldsymbol{\pi}(0)^T$. Then, since $\pi_i(k) \geq 0$, $i \in S$, and $\sum_{i \in S} \pi_i(0) = 1$,

$$\begin{aligned} a(k) &= \sum_{i \in S} \pi_i(k) = \|\boldsymbol{\pi}(k)^T\|_1 \leq \|\mathbf{P}_Z(\delta)^{Tk}\|_1 \|\boldsymbol{\pi}(0)^T\|_1 = \|\mathbf{P}_Z(\delta)^{Tk}\|_1 \sum_{i \in S} \pi_i(0) \\ &= \|\mathbf{P}_Z(\delta)^{Tk}\|_1. \end{aligned}$$

Let $\rho(\mathbf{P}_Z(\delta)^T)$ denote the spectral radius of $\mathbf{P}_Z(\delta)^T$. We have [37, Theorem 3.1] that, for $k \rightarrow \infty$, $\|\mathbf{P}_Z(\delta)^{Tk}\|_1 \sim B(\delta) \binom{k}{p(\delta)-1} \rho(\mathbf{P}_Z(\delta)^T)^k$, $B(\delta) > 0$, $p(\delta)$ integer ≥ 1 ⁴. Also, since the eigenvalues of a matrix are continuous functions of the elements of the matrix [30, Theorem 3.13], we have $\lim_{\delta \rightarrow 0} \rho(\mathbf{P}_Z(\delta)^T) = \rho(\mathbf{P}_Z(0)^T)$. But, with the ordering of states S_0, S_1, \dots, S_{N_C} , the elements in the lower triangular portion of $\mathbf{P}_Z(0)^T$ are 0 and the diagonal elements have values 0 and values $1 - \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}} / ((1 + \theta) \max_{0 \leq k \leq N_C} \max_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}})$, $i \in S_k$, $0 < k \leq N_C$ and, then, $\rho(\mathbf{P}_Z(0)^T) = q$.

From (12), $\boldsymbol{\pi}'(k)^T = \mathbf{P}_{Z'}(\delta)^{Tk} \boldsymbol{\pi}'(0)^T$. Then, since $\pi'_i(k) \geq 0$, and $\sum_{i \in S'} \pi'_i(0) = \alpha_{S'}$,

$$\begin{aligned} a'(k) &= \sum_{i \in S'} \pi'_i(k) = \|\boldsymbol{\pi}'(k)^T\|_1 \leq \|\mathbf{P}_{Z'}(\delta)^{Tk}\|_1 \|\boldsymbol{\pi}'(0)^T\|_1 = \|\mathbf{P}_{Z'}(\delta)^{Tk}\|_1 \sum_{i \in S'} \pi'_i(0) \\ &= \alpha_{S'} \|\mathbf{P}_{Z'}(\delta)^{Tk}\|_1. \end{aligned}$$

As before, we have that, for $k \rightarrow \infty$, $\|\mathbf{P}_{Z'}(\delta)^{Tk}\|_1 \sim B'(\delta) \binom{k}{p'(\delta)-1} \rho(\mathbf{P}_{Z'}(\delta)^T)^k$, $B'(\delta) > 0$, $p'(\delta)$ integer ≥ 1 and $\lim_{\delta \rightarrow 0} \rho(\mathbf{P}_{Z'}(\delta)^T) = \rho(\mathbf{P}_{Z'}(0)^T)$. But, with the ordering of states S_1, \dots, S_{N_C} , the elements in the lower triangular portion of $\mathbf{P}_{Z'}(0)^T$ are 0 and the diagonal elements have values $1 - \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}} / ((1 + \theta) \max_{0 \leq k \leq N_C} \max_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}})$, $i \in S_k$, $0 < k \leq N_C$ and, then, $\rho(\mathbf{P}_{Z'}(0)^T) = q$. \square

Theorem 4 tells that, for class C models with the selection $r = o$, both $a(k)$ and $a'(k)$ are upper bounded by something which decays asymptotically by a factor with value q for $\delta \rightarrow 0$. Let $R = \max_{i \in S} \lambda_i / \min_{i \in S - \{o\}} \lambda_i \geq 1$. For small δ , $\min_{0 < k \leq N_C} \min_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}} \approx \min_{i \in S - \{o\}} \lambda_i$ and $\max_{0 \leq k \leq N_C} \max_{i \in S_k} \lambda_{i, S_0 \cup \dots \cup S_{k-1} \cup \{f_1, \dots, f_A\}} \approx \max_{i \in S} \lambda_i$ and, since θ is a small quantity > 0 , $q \approx 1 - 1/R$. Then, the closer R to 1, the smaller q would be, the faster should $a(k)$ and $a'(k)$ decrease, and the smaller the model truncation parameters should be. Note that R is a ‘‘visible’’ model characteristic, i.e. one that can be easily estimated.

Theorem 4 also suggests that the K and L required by regenerative randomization could be roughly upper bounded assuming $a(k) = a'(k) = q^k$. In order to asses the accuracy in practice of that approximation, we will consider the paradigmatic CTMC with initial state s_2 and the state transition diagram of Figure 7. That example falls in the model class C if $\beta \ll \mu + \gamma$ (the subsets

⁴Strictly speaking, Theorem 3.1 of [37] asserts the result for the Euclidean norm, but the result easily extends to the considered 1-norm.

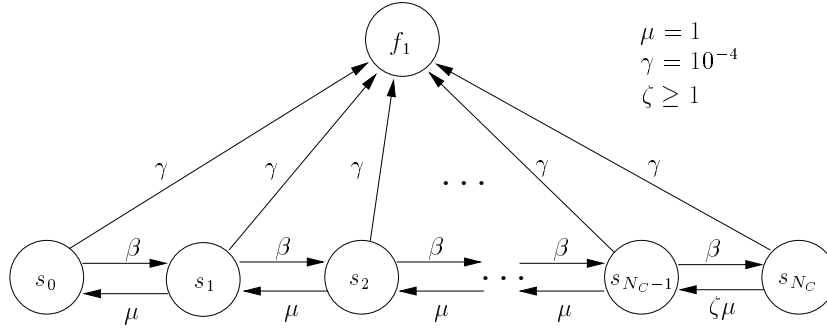


Figure 7: State transition diagram of the paradigmatic CTMC.

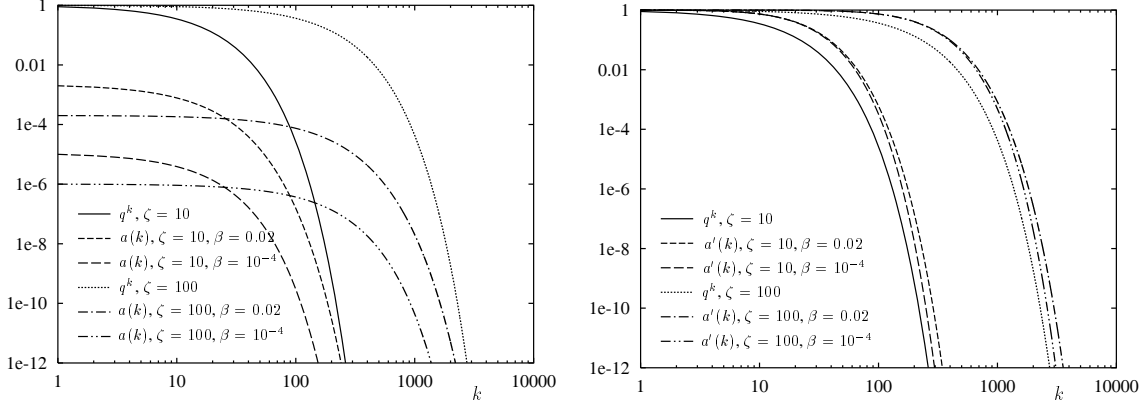


Figure 8: Behavior of $a(k)$ (left) and $a'(k)$ (right) compared with q^k for the paradigmatic CTMC for $N_C = 5$.

of a partition satisfying properties P1 and P2 are $S_k = \{s_k\}$, $0 \leq k \leq N_C$). The value of q for the example is

$$q = 1 - \frac{\mu + \gamma}{(1 + \theta)(\zeta\mu + \gamma)}.$$

We will consider the values $\zeta = 10$ and $\zeta = 100$ and will take $\theta = 10^{-4}$. This yields $q \approx 0.9$ for $\zeta = 10$ and $q \approx 0.99$ for $\zeta = 100$. We will consider two values for β : $\beta = 0.02$ and $\beta = 10^{-4}$, yielding $\delta \approx 0.02$ and $\delta \approx 10^{-4}$, respectively. Figure 8, left shows the behavior of q^k and $a(k)$ for $N_C = 5$. Figure 8, right shows the behavior of q^k and $a'(k)$, also for $N_C = 5$. Although $a(k)$ are obviously significantly different from q^k , q^k seems to be a reasonable pessimistic approximation for $a(k)$. The behavior of $a'(k)$ is very well matched by q^k , but this depends on the selection of the initial state of the CTMC. We found that as the initial state is farther apart from s_0 the difference between $a'(k)$ and q^k increases, but, nevertheless, q^k is still a reasonable approximation for $a'(k)$. Using further the approximation $q \approx 1 - 1/R$, we can try to approximate the K and L required by regenerative randomization taking $a(k) = a'(k) = (1 - 1/R)^k$. It is easy to prove that, for not small R , fixed, small ε and fixed, large Δt , that approximation yields required K and L which are proportional to R . For the paradigmatic example, $R \approx \zeta$ and, in the light of the results shown in Figure 8, we can propose the rule of thumb that, for class C models with the selection $r = o$, the required K and L can be roughly upper bounded by $30R$.

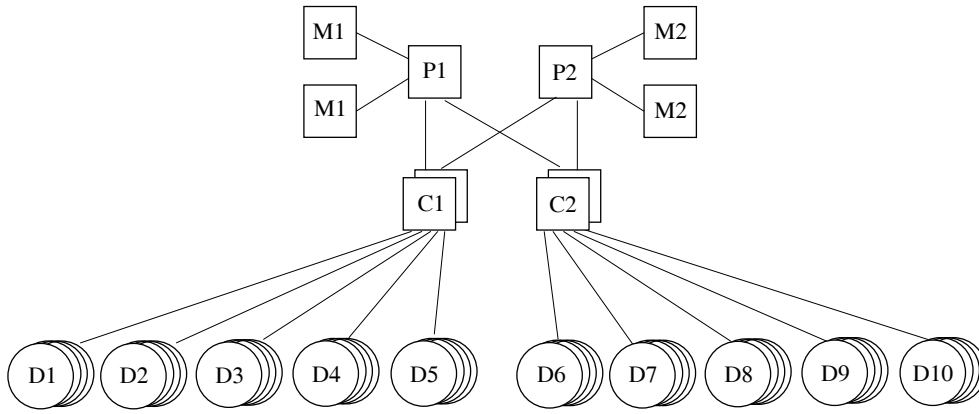


Figure 9: Block diagram of the example.

4 Analysis

In this section, using a large reliability example belonging to the class C described in Section 3, we illustrate the performance of regenerative randomization and show that it can indeed be much faster than standard randomization. The example is the fault-tolerant system whose block diagram is given in Figure 9. The system is made up of two processing subsystems, each including one processor P and two memories M, two sets of controllers with two controllers per set, and 10 sets of disks, each with four disks. Each set of controllers controls 5 sets of disks. The system is operational if at least one processor and one memory connected to it are unfailed, one controller of each set is unfailed, and three disks of each set are unfailed. Processors fail with rate $2 \times 10^{-5} \text{ h}^{-1}$; a processor failure is soft with probability 0.8 and hard with probability 0.2. Memories fail with rate 10^{-4} h^{-1} . Controllers fail with rate $2 \times 10^{-5} \text{ h}^{-1}$. Disks fail with rates 10^{-5} h^{-1} . A failure of a controller is uncovered and is propagated to two disks of a randomly chosen set of disks controlled by it with probability 0.01. There are two repairmen who repair processors in soft failure with rate μ_{PS} . The other repair actions are performed by another repairman, with first priority given to disks, next to controllers, next to processors, and last to memories. Components with the same repair priority are chosen at random. The repair rates are 0.2 h^{-1} for processors in hard failure mode, 0.2 h^{-1} for memories, 0.5 h^{-1} for controllers, and 0.5 h^{-1} for disks. For μ_{PS} we will consider two values: $\mu_{PS} = 1 \text{ h}^{-1}$ and $\mu_{PS} = 10 \text{ h}^{-1}$. The measure of interest is the unreliability at time t , $ur(t)$, a particular case of the measure $m(t)$ considered in this paper. The number of states and transitions of the corresponding CTMC X are 131,073 and 1,876,132, respectively. As regenerative state r we choose, of course, the single state without failed components. Under the partition $S_k = \{\text{operational states with } k \text{ failed components}\}$, $\delta = 0.004596$, and the analysis for the performance of regenerative randomization for class C models made in Section 3 should apply. The standard randomization method is implemented with $\Lambda = \max_{i \in S} \lambda_i$. The methods are run with a single target time t .

We illustrate first the dependence of the required K and L in regenerative randomization on t and ε and compare K and L with the number of steps N required in standard randomization. Table 1 gives results for $\varepsilon = 10^{-8}, 10^{-10}$ for the example with $\mu_{PS} = 1 \text{ h}^{-1}$ and initial state the state in which one controller C1 is failed. The unreliability $ur(t)$ varied from 1.220×10^{-4} for $t = 100 \text{ h}$ to 4.062×10^{-2} for $t = 50,000 \text{ h}$. Table 2 gives results for the model with same

Table 1: Required K and L in regenerative randomization and required number of steps N in standard randomization, for increasing t and $\varepsilon = 10^{-8}, 10^{-10}$ for the example with initial state the state in which one controller C1 is failed and $\mu_{PS} = 1 \text{ h}^{-1}$.

t (h)	$\varepsilon = 10^{-8}$			$\varepsilon = 10^{-10}$		
	K	L	N	K	L	N
100	110	98	224	136	130	234
200	121	98	402	153	131	417
500	130	98	909	163	131	931
1,000	135	98	1,723	169	131	1,754
2,000	141	98	3,314	174	131	3,357
5,000	148	98	7,996	181	131	8,062
10,000	153	98	15,701	186	131	15,795
20,000	158	98	30,995	191	131	31,126
50,000	164	98	76,587	198	131	76,794

initial state but $\mu_{PS} = 10 \text{ h}^{-1}$. In that case, $ur(t)$ varied from 1.220×10^{-4} for $t = 100 \text{ h}$ to 4.059×10^{-2} for $t = 50,000 \text{ h}$. The required K increases logarithmically with t while the required L is almost independent on t . The results also show a moderate increase of the required K and L when ε decreases. All this is in agreement with the fact that the required K is $O(\log(\Delta t/\varepsilon))$ and the required L is $O(\log(1/\varepsilon))$. Also, the rule of thumb given in Section 3 that the required K and L are roughly upper bounded by $30R$ seems to work well ($R = 7.48$ for $\mu_{PS} = 1 \text{ h}^{-1}$ and $R = 52.3$ for $\mu_{PS} = 10 \text{ h}^{-1}$). The number of steps on Z, Z' in regenerative randomization, $K + L$, is smaller than N in all cases except for $\mu_{PS} = 1 \text{ h}^{-1}, \varepsilon = 10^{-10}$ and $t = 100 \text{ h}$, and for $\mu_{PS} = 10 \text{ h}^{-1}$ and $t = 100 \text{ h}$. For large t , $K + L$ is much smaller than N .

We compare next the computational costs (measured in terms of CPU times) of regenerative randomization and standard randomization. Figure 10 gives measured CPU times for regenerative randomization and measured/estimated CPU times for standard randomization (when the CPU times required by standard randomization were very large we estimated them from the number of required steps) for the example with initial state the state without failed components, t varying from 100 to 50,000 h, $\mu_{PS} = 1 \text{ h}^{-1}$ (left) and $\mu_{PS} = 10 \text{ h}^{-1}$ (right). In both cases we took $\varepsilon = 10^{-8}$. The measure $ur(t)$ varied from 8.280×10^{-5} to 4.058×10^{-2} for $\mu_{PS} = 1 \text{ h}^{-1}$ and from 8.274×10^{-5} to 4.056×10^{-2} for $\mu_{PS} = 10 \text{ h}^{-1}$. All CPU times were measured/estimated on an 167 MHz, 128 MB UltraSPARC 1 workstation. Memory usage of both methods was about 70 MB. For regenerative randomization we decompose the CPU time in CPU time required to obtain the truncated transformed model (trans) and CPU time required to solve it by standard randomization (sol). We can note that regenerative randomization is always faster than standard randomization. The total CPU time required by regenerative randomization increases smoothly with t , whereas the CPU time required by standard randomization increases approximately linearly with t . For $\mu_{PS} = 1 \text{ h}^{-1}$, the total CPU time required by regenerative randomization goes from 76.93 s for $t = 100 \text{ h}$ to 119.4 s for $t = 50,000 \text{ h}$, whereas the CPU time required by standard randomization goes from 141.5 s for $t = 100 \text{ h}$ to $4.835 \times 10^4 \text{ s}$ (about 13 hours) for $t = 50,000 \text{ h}$; for the largest t ,

Table 2: Required K and L in regenerative randomization and required number of steps N in standard randomization, for increasing t and $\varepsilon = 10^{-8}, 10^{-10}$ for the example with initial state the state in which one controller C1 is failed and $\mu_{PS} = 10 \text{ h}^{-1}$.

t (h)	$\varepsilon = 10^{-8}$			$\varepsilon = 10^{-10}$		
	K	L	N	K	L	N
100	801	722	1,237	980	967	1,263
200	886	722	2,362	1,120	967	2,398
500	953	722	5,662	1,197	967	5,718
1,000	995	722	11,081	1,241	967	11,159
2,000	1,035	722	21,820	1,282	967	21,930
5,000	1,086	722	53,795	1,333	967	53,969
10,000	1,124	722	106,833	1,371	967	107,077
20,000	1,161	722	212,595	1,409	967	212,940
50,000	1,210	722	529,116	1,458	967	529,660

regenerative randomization is 405 times faster than standard randomization. For $\mu_{PS} = 10 \text{ h}^{-1}$, the total CPU time required by regenerative randomization goes from 545.4 s for $t = 100 \text{ h}$ to 1,035 s for $t = 50,000 \text{ h}$, whereas the CPU time required by standard randomization goes from 777.1 s for $t = 100 \text{ h}$ to $3.363 \times 10^5 \text{ s}$ (about 4 days) for $t = 50,000 \text{ h}$; for the largest t , regenerative randomization is 325 times faster than standard randomization. Also interesting is the distribution of the CPU time required by regenerative randomization. For $\mu_{PS} = 1 \text{ h}^{-1}$, the time spent solving the truncated transformed model by standard randomization is negligible compared to the time spent obtaining that model. However, for $\mu_{PS} = 10 \text{ h}^{-1}$, that time is significant for large t . This is because, being the maximum output rate of the truncated transformed model, equal to $(1 + \theta) \max_{i \in S} \lambda_i$, larger for $\mu_{PS} = 10 \text{ h}^{-1}$, standard randomization becomes a relatively less efficient method to solve the truncated transformed model. Figure 11 compares the CPU times of regenerative randomization and standard randomization for the example with initial state the state with one controller C1 failed and $\varepsilon = 10^{-8}$ for $\mu_{PS} = 1 \text{ h}^{-1}$ (left) and $\mu_{PS} = 10 \text{ h}^{-1}$ (right). In those cases, there is a crosspoint time t below which standard randomization is faster than standard randomization. For $\mu_{PS} = 1 \text{ h}^{-1}$, the CPU time required by regenerative randomization goes from 148.4 s for $t = 100 \text{ h}$ to 192.3 s for $t = 50,000 \text{ h}$, whereas the CPU time required by standard randomization goes from 140.9 s for $t = 100 \text{ h}$ to $4.765 \times 10^4 \text{ s}$ for $t = 50,000 \text{ h}$, making regenerative randomization 248 times faster than standard randomization for the largest t . For $\mu_{PS} = 10 \text{ h}^{-1}$, the CPU time required by regenerative randomization goes from 1,088 s for $t = 100 \text{ h}$ to 1,585 s for $t = 50,000 \text{ h}$, whereas the CPU time required by standard randomization goes from 775.6 s for $t = 100 \text{ h}$ to $3.292 \times 10^5 \text{ s}$ for $t = 50,000$, making regenerative randomization 208 times faster than standard randomization for the largest t .

Table 3 compares the K required by the regenerative randomization method and the K, K' , which would be required were the error associated with the truncation of the transformed model controlled by computing $m_K^e(t)$ exactly, for the example with initial state the state without failed components, $\varepsilon = 10^{-8}$, and $\mu_{PS} = 1 \text{ h}^{-1}, 10 \text{ h}^{-1}$. We can note that K is very close to K' (in most

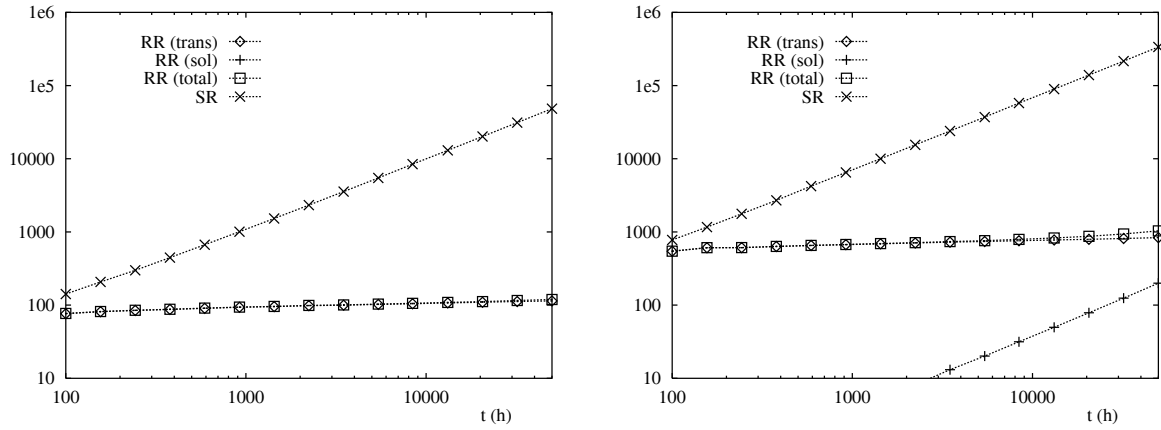


Figure 10: CPU times in seconds required by regenerative randomization (RR) and standard randomization (SR) to solve the example with initial state the state without failed components and $\varepsilon = 10^{-8}$ for $\mu_{PS} = 1 \text{ h}^{-1}$ (left) and $\mu_{PS} = 10 \text{ h}^{-1}$ (right).

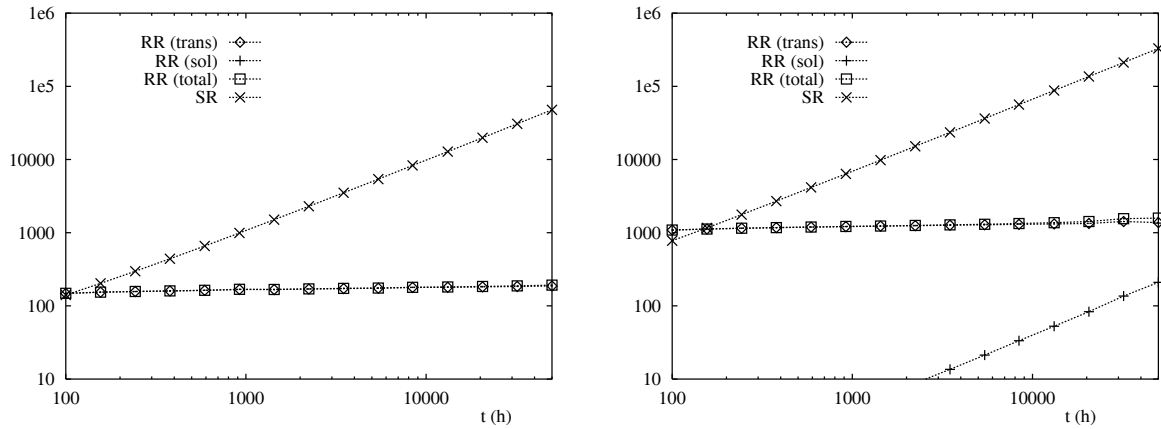


Figure 11: CPU times in seconds required by regenerative randomization (RR) and standard randomization (SR) to solve the example with initial state the state with one controller C1 failed and $\varepsilon = 10^{-8}$ for $\mu_{PS} = 1 \text{ h}^{-1}$ (left) and $\mu_{PS} = 10 \text{ h}^{-1}$ (right).

Table 3: Required K and K' for increasing t and $\varepsilon = 10^{-8}$ for the example with initial state the state without failed components and $\mu_{PS} = 1 \text{ h}^{-1}, 10 \text{ h}^{-1}$.

t (h)	$\mu_{PS} = 1 \text{ h}^{-1}$		$\mu_{PS} = 10 \text{ h}^{-1}$	
	K	K'	K	K'
100	106	106	770	770
200	116	116	850	850
500	125	125	916	916
1,000	130	130	959	958
2,000	136	136	998	998
5,000	143	143	1,049	1,048
10,000	148	148	1,086	1,086
20,000	153	153	1,124	1,123
50,000	159	159	1,173	1,172

cases equal and in some cases greater by one). Thus, the upper bounds for $m_{K,L}^{e''}(t)$ and $m_{K,L}^e(t)$ given in Section 2 seem to be, for class C models, quite tight.

5 Related work

Reference [5], which is based on [6], describes preliminary related work. In [5], it is considered the particular case in which $A = 1$ and all states in S are transient, the error associated with the truncation of the transformed model is controlled by computing $m_{K,L}^{e''}(t)$, $m_{K,L}^e(t)$ using a numerical Laplace inversion algorithm, and the approximated model solution $m_{K,L}^a(t)$, $m_{K,L}^a(t)$ is computed using also a numerical Laplace inversion algorithm. That strategy for controlling the error associated with the truncation of the transformed model is expensive when the required K is large. Also, computing $m_{K,L}^a(t)$, $m_{K,L}^a(t)$ using a numerical Laplace inversion algorithm results in a method in which the computation error is less well-controlled. Finally, [5] does not analyze the performance of the method for class C models with the selection $r = o$.

6 Conclusions

We have developed a new method called regenerative randomization for the transient analysis of continuous time Markov chain models with absorbing states. The method has the same good properties (numerical stability, well-controlled computation error, and ability to specify the computation error in advance) as the well-known standard randomization method and can be significantly less costly than that method for large models and large t . The method requires the selection of a regenerative state, on which the performance of the method depends. Automatic selection of the regenerative state seems to be difficult, in general, and the method relies on the user's intuition to select a good regenerative state. However, a class of models, class C, including typical failure/repair models with

exponential failure and repair time distributions and repair in every state with failed components, has been identified for which a natural selection for the regenerative state exists and, for those models, theoretical results assessing approximately the performance of the method with that natural selection in terms of “visible” model characteristics have been obtained. Those theoretical results can be used to anticipate, for class C models and that natural selection for the regenerative state, when regenerative randomization can be expected to be significantly less costly than standard randomization. Using an example belonging to that class, we have illustrated the performance of regenerative randomization and have shown that it can indeed be much faster than standard randomization, allowing a numerically stable transient analysis, with well-controlled and specifiable-in-advance computation error, of very large CTMC models with absorbing states in affordable CPU times.

Acknowledgments

This work was supported by the “Comisión Interministerial de Ciencia y Tecnología” (CICYT) of the Ministry of Science and Technology of Spain under the research grants TIC95-0707-C02-02 and TAP1999-0443-C05-05.

Appendix

Lemma 1. *Let X and Y be finite CTMCs with identical state spaces and initial probability distributions and transition rate matrices \mathbf{A} and \mathbf{B} , respectively. Let $\mathbf{p}_X(t)$ and $\mathbf{p}_Y(t)$ be the probability distribution column vectors of, respectively, X and Y at time t . Then*

$$\|\mathbf{p}_Y(t) - \mathbf{p}_X(t)\|_1 \leq \|\mathbf{B}^T - \mathbf{A}^T\|_1 t.$$

Proof. Vector $\mathbf{p}_Y(t)$ is the solution of

$$\frac{d\mathbf{p}_Y}{dt} = \mathbf{B}^T \mathbf{p}_Y(t),$$

which can be rewritten as

$$\frac{d\mathbf{p}_Y}{dt} = \mathbf{A}^T \mathbf{p}_Y(t) + (\mathbf{B}^T - \mathbf{A}^T) \mathbf{p}_Y(t),$$

from which, since $\mathbf{p}_X(t) = e^{\mathbf{A}^T t} \mathbf{p}_X(0)$ and $\mathbf{p}_X(0) = \mathbf{p}_Y(0)$,

$$\begin{aligned} \mathbf{p}_Y(t) &= e^{\mathbf{A}^T t} \mathbf{p}_Y(0) + \int_0^t e^{\mathbf{A}^T(t-s)} (\mathbf{B}^T - \mathbf{A}^T) \mathbf{p}_Y(s) ds \\ &= \mathbf{p}_X(t) + \int_0^t e^{\mathbf{A}^T(t-s)} (\mathbf{B}^T - \mathbf{A}^T) \mathbf{p}_Y(s) ds, \end{aligned}$$

which implies

$$\begin{aligned} \|\mathbf{p}_Y(t) - \mathbf{p}_X(t)\|_1 &= \left\| \int_0^t e^{\mathbf{A}^T(t-s)} (\mathbf{B}^T - \mathbf{A}^T) \mathbf{p}_Y(s) ds \right\|_1 \\ &\leq \int_0^t \|e^{\mathbf{A}^T(t-s)}\|_1 \|\mathbf{B}^T - \mathbf{A}^T\|_1 \|\mathbf{p}_Y(s)\|_1 ds. \end{aligned}$$

But $\|\mathbf{p}_Y(s)\|_1 = 1$ and $\|e^{\mathbf{A}^T(t-s)}\|_1 = 1$, since column i of $e^{\mathbf{A}^T(t-s)}$ is the probability distribution vector of X at time $t - s$ conditioned to $X(0) = i$, and the result follows. \square

Proof of Proposition 1 The initial probability distribution of \widehat{V} follows immediately from its definition (13). It is also clear that: 1) $\widehat{V}_k \in \{s_0, s_1, \dots, s_k, s'_k, f_1, f_2, \dots, f_A\}$, 2) from a state s_l , \widehat{V} can only jump to states f_i , $1 \leq i \leq A$, s_0 and s_{l+1} , and 3) from a state s'_k , \widehat{V} can only jump to states f_i , $1 \leq i \leq A$, s_0 and s'_{k+1} . Since $\widehat{V}_k = f_i$ if and only if $\widehat{X}_k = f_i$ and f_i is absorbing in \widehat{X} , from state f_i , \widehat{V} can only jump to f_i and $P[\widehat{V}_{k+1} = f_i | \widehat{V}_k = f_i] = P[\widehat{V}_{k+1} = f_i | \widehat{V}_k = f_i \wedge \widehat{V}_{k-1} = x_{k-1} \wedge \dots \wedge \widehat{V}_0 = x_0] = 1$. To completely verify that \widehat{V} is a DTMC we have to show that $P[\widehat{V}_{k+1} = y | \widehat{V}_k = s_l \wedge \widehat{V}_{k-1} = x_{k-1} \wedge \dots \wedge \widehat{V}_0 = x_0] = P[\widehat{V}_{k+1} = y | \widehat{V}_k = s_l]$ for all feasible paths $(x_0, x_1, \dots, x_{k-1}, s_l)$ of \widehat{V} and that $P[\widehat{V}_{k+1} = y | \widehat{V}_k = s'_k \wedge \widehat{V}_{k-1} = x_{k-1} \wedge \dots \wedge \widehat{V}_0 = x_0] = P[\widehat{V}_{k+1} = y | \widehat{V}_k = s'_k]$ for all feasible paths $(x_0, x_1, \dots, x_{k-1}, s'_k)$ of \widehat{V} . But, $\widehat{V}_k = s_l$ implies $\widehat{V}_{k-l} = s_0, \widehat{V}_{k-l+1} = s_1, \dots, \widehat{V}_{k-1} = s_{l-1}$ and, given the definition of \widehat{V} (13), the behavior of \widehat{V} after the steps m at which \widehat{V} hits state s_0 is independent of $\widehat{X}_0, \widehat{X}_1, \dots, \widehat{X}_{m-1}$ and, therefore, of $\widehat{V}_0, \widehat{V}_1, \dots, \widehat{V}_{m-1}$, and, then, $P[\widehat{V}_{k+1} = y | \widehat{V}_k = s_l \wedge \widehat{V}_{k-1} = s_{l-1} \wedge \dots \wedge \widehat{V}_{k-l} = s_0 \wedge \widehat{V}_{k-l-1} = x_{k-l-1} \wedge \dots \wedge \widehat{V}_0 = x_0] = P[\widehat{V}_{k+1} = y | \widehat{V}_k = s_l \wedge \widehat{V}_{k-1} = s_{l-1} \wedge \dots \wedge \widehat{V}_{k-l} = s_0] = P[\widehat{V}_{k+1} = y | \widehat{V}_k = s_l]$. Similarly, $\widehat{V}_k = s'_k$ implies $\widehat{V}_0 = s'_0, \widehat{V}_1 = s'_1, \dots, \widehat{V}_{k-1} = s'_{k-1}$ and $P[\widehat{V}_{k+1} = y | \widehat{V}_k = s'_k \wedge \widehat{V}_{k-1} = s'_{k-1} \wedge \dots \wedge \widehat{V}_0 = s'_0] = P[\widehat{V}_{k+1} = s'_{k+1} | \widehat{V}_k = s'_k]$. It remains to verify the values of the transition probabilities from states s'_l and s_l . We will use $\widehat{V}_k \in \{s_0, s_1, \dots, s_k, s'_k, f_1, f_2, \dots, f_A\}$.

Case a ($P[\widehat{V}_k = s_0] > 0$, which implies (13) $P[\widehat{X}_k = r] > 0$): Taking into account the definition of \widehat{V} (13), that $\pi_r(0) = 1$ and $\pi_i(0) = 0$, $i \in S'$, and the definition of $a(l)$:

$$\begin{aligned} P[\widehat{V}_{k+1} = f_j | \widehat{V}_k = s_0] &= \frac{P[\widehat{V}_k = s_0 \wedge \widehat{V}_{k+1} = f_j]}{P[\widehat{V}_k = s_0]} = \frac{P[\widehat{X}_k = r \wedge \widehat{X}_{k+1} = f_j]}{P[\widehat{X}_k = r]} \\ &= P[\widehat{X}_{k+1} = f_j | \widehat{X}_k = r] = P_{r,f_j} = \frac{\sum_{i \in S} \pi_i(0) P_{i,f_j}}{\sum_{i \in S} \pi_i(0)} = \frac{\sum_{i \in S} \pi_i(0) P_{i,f_j}}{a(0)} = v_0^j. \end{aligned}$$

Similarly, we have:

$$\begin{aligned} P[\widehat{V}_{k+1} = s_0 | \widehat{V}_k = s_0] &= \frac{P[\widehat{V}_k = s_0 \wedge \widehat{V}_{k+1} = s_0]}{P[\widehat{V}_k = s_0]} = \frac{P[\widehat{X}_k = r \wedge \widehat{X}_{k+1} = r]}{P[\widehat{X}_k = r]} \\ &= P[\widehat{X}_{k+1} = r | \widehat{X}_k = r] = P_{r,r} = \frac{\sum_{i \in S} \pi_i(0) P_{i,r}}{\sum_{i \in S} \pi_i(0)} = \frac{\sum_{i \in S} \pi_i(0) P_{i,r}}{a(0)} = q_0, \\ P[\widehat{V}_{k+1} = s_1 | \widehat{V}_k = s_0] &= \frac{P[\widehat{V}_k = s_0 \wedge \widehat{V}_{k+1} = s_1]}{P[\widehat{V}_k = s_0]} = \frac{P[\widehat{X}_k = r \wedge \widehat{X}_{k+1} \in S']}{P[\widehat{X}_k = r]} \\ &= P[\widehat{X}_{k+1} \in S' | \widehat{X}_k = r] = P_{r,S'} = \frac{\sum_{i \in S} \pi_i(0) P_{i,S'}}{\sum_{i \in S} \pi_i(0)} = \frac{\sum_{i \in S} \pi_i(0) P_{i,S'}}{a(0)} = w_0. \end{aligned}$$

Case b ($P[\widehat{V}_k = s_l] > 0$, $1 \leq l \leq k$, which implies $P[\widehat{X}_{k-l} = r] > 0$ (13)): Taking into account the definition of \widehat{V} (13), that the DTMC $\{\widehat{X}_{m+k}; k = 0, 1, 2, \dots\}$ conditioned to $\widehat{X}_m = r$ is probabilistically identical to $\widehat{X}' = \{\widehat{X}'_k; k = 0, 1, 2, \dots\}$, the definition of Z (3), that $Z_l \neq r$ for $l \geq 1$ and, therefore, for $l \geq 1$, $Z_l \in S'$ if and only if $Z_l \in S$, that (4) $P[Z_{l+1} = f_j | Z_l = i] = P_{i,f_j}$, $i \in S$, and the definition of $a(l)$:

$$P[\widehat{V}_{k+1} = f_j | \widehat{V}_k = s_l] = \frac{P[\widehat{V}_k = s_l \wedge \widehat{V}_{k+1} = f_j]}{P[\widehat{V}_k = s_l]}$$

$$\begin{aligned}
&= \frac{P[\widehat{X}_{k-l} = r \wedge \widehat{X}_{k-l+1:k} \in S' \wedge \widehat{X}_{k+1} = f_j]}{P[\widehat{X}_{k-l} = r \wedge \widehat{X}_{k-l+1:k} \in S']} \\
&= \frac{P[\widehat{X}_{k-l+1:k} \in S' \wedge \widehat{X}_{k+1} = f_j \mid \widehat{X}_{k-l} = r]}{P[\widehat{X}_{k-l+1:k} \in S' \mid \widehat{X}_{k-l} = r]} \\
&= \frac{P[\widehat{X}'_{1:l} \in S' \wedge \widehat{X}'_{l+1} = f_j]}{P[\widehat{X}'_{1:l} \in S']} = \frac{P[Z_l \in S' \wedge Z_{l+1} = f_j]}{P[Z_l \in S']} = \frac{P[Z_l \in S \wedge Z_{l+1} = f_j]}{P[Z_l \in S]} \\
&= \frac{\sum_{i \in S} \pi_i(l) P_{i,f_j}}{\sum_{i \in S} \pi_i(l)} = \frac{\sum_{i \in S} \pi_i(l) P_{i,f_j}}{a(l)} = v_l^j.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
P[\widehat{V}_{k+1} = s_0 \mid \widehat{V}_k = s_l] &= \frac{P[\widehat{V}_k = s_l \wedge \widehat{V}_{k+1} = s_0]}{P[\widehat{V}_k = s_l]} \\
&= \frac{P[\widehat{X}_{k-l} = r \wedge \widehat{X}_{k-l+1:k} \in S' \wedge \widehat{X}_{k+1} = r]}{P[\widehat{X}_{k-l} = r \wedge \widehat{X}_{k-l+1:k} \in S']} = \frac{P[\widehat{X}_{k-l+1:k} \in S' \wedge \widehat{X}_{k+1} = r \mid \widehat{X}_{k-l} = r]}{P[\widehat{X}_{k-l+1:k} \in S' \mid \widehat{X}_{k-l} = r]} \\
&= \frac{P[\widehat{X}'_{1:l} \in S' \wedge \widehat{X}'_{l+1} = r]}{P[\widehat{X}'_{1:l} \in S']} = \frac{P[Z_l \in S' \wedge Z_{l+1} = a]}{P[Z_l \in S']} = \frac{P[Z_l \in S \wedge Z_{l+1} = a]}{P[Z_l \in S]} \\
&= \frac{\sum_{i \in S} \pi_i(l) P_{i,r}}{\sum_{i \in S} \pi_i(l)} = \frac{\sum_{i \in S} \pi_i(l) P_{i,r}}{a(l)} = q_l,
\end{aligned}$$

$$\begin{aligned}
P[\widehat{V}_{k+1} = s_{l+1} \mid \widehat{V}_k = s_l] &= \frac{P[\widehat{V}_k = s_l \wedge \widehat{V}_{k+1} = s_{l+1}]}{P[\widehat{V}_k = s_l]} \\
&= \frac{P[\widehat{X}_{k-l} = r \wedge \widehat{X}_{k-l+1:k+1} \in S']}{P[\widehat{X}_{k-l} = r \wedge \widehat{X}_{k-l+1:k} \in S']} = \frac{P[\widehat{X}_{k-l+1:k+1} \in S' \mid \widehat{X}_{k-l} = r]}{P[\widehat{X}_{k-l+1:k} \in S' \mid \widehat{X}_{k-l} = r]} \\
&= \frac{P[\widehat{X}'_{1:l+1} \in S']}{P[\widehat{X}'_{1:l} \in S']} = \frac{P[Z_{l+1} \in S']}{P[Z_l \in S']} = \frac{P[Z_l \in S \wedge Z_{l+1} \in S']}{P[Z_l \in S]} \\
&= \frac{\sum_{i \in S} \pi_i(l) P_{i,S'}}{\sum_{i \in S} \pi_i(l)} = \frac{\sum_{i \in S} \pi_i(l) P_{i,S'}}{a(l)} = w_l.
\end{aligned}$$

Case c ($P[\widehat{V}_k = s'_k] > 0$): Taking into account the definition of \widehat{V} (13), the definition of Z' (7), that (8) $P[Z'_{k+1} = f_j \mid Z'_k = i] = P_{i,f_j}$, $i \in S'$, and the definition of $a'(k)$:

$$\begin{aligned}
P[\widehat{V}_{k+1} = f_j \mid \widehat{V}_k = s'_k] &= \frac{P[\widehat{V}_k = s'_k \wedge \widehat{V}_{k+1} = f_j]}{P[\widehat{V}_k = s'_k]} = \frac{P[\widehat{X}_{0:k} \in S' \wedge \widehat{X}_{k+1} = f_j]}{P[\widehat{X}_{0:k} \in S']} \\
&= \frac{P[Z'_k \in S' \wedge Z'_{k+1} = f_j]}{P[Z'_k \in S']} = \frac{\sum_{i \in S'} \pi'_i(k) P_{i,f_j}}{\sum_{i \in S'} \pi'_i(k)} = \frac{\sum_{i \in S'} \pi'_i(k) P_{i,f_j}}{a'(k)} = v_k^j.
\end{aligned}$$

Similarly, we have:

$$\begin{aligned}
P[\widehat{V}_{k+1} = s_0 \mid \widehat{V}_k = s'_k] &= \frac{P[\widehat{V}_k = s'_k \wedge \widehat{V}_{k+1} = s_0]}{P[\widehat{V}_k = s'_k]} = \frac{P[\widehat{X}_{0:k} \in S' \wedge \widehat{X}_{k+1} = r]}{P[\widehat{X}_{0:k} \in S']} \\
&= \frac{P[Z'_k \in S' \wedge Z'_{k+1} = a]}{P[Z'_k \in S']} = \frac{\sum_{i \in S'} \pi'_i(k) P_{i,r}}{\sum_{i \in S'} \pi'_i(k)} = \frac{\sum_{i \in S'} \pi'_i(k) P_{i,r}}{a'(k)} = q_k,
\end{aligned}$$

$$\begin{aligned}
P[\widehat{V}_{k+1} = s'_{k+1} \mid \widehat{V}_k = s'_k] &= \frac{P[\widehat{V}_k = s'_k \wedge \widehat{V}_{k+1} = s'_{k+1}]}{P[\widehat{V}_k = s'_k]} = \frac{P[\widehat{X}_{0:k+1} \in S']}{P[\widehat{X}_{0:k} \in S']} \\
&= \frac{P[Z'_{k+1} \in S']}{P[Z'_k \in S']} = \frac{P[Z'_k \in S' \wedge Z'_{k+1} \in S']}{P[Z'_k \in S']} = \frac{\sum_{i \in S'} \pi'_i(k) P_{i,S'}}{\sum_{i \in S'} \pi'_i(k)} \\
&= \frac{\sum_{i \in S'} \pi'_i(k) P_{i,S'}}{a'(k)} = w'_k. \quad \square
\end{aligned}$$

Proof of Proposition 3 We start by considering the case $\alpha_{S'} > 0$. Assuming $P[(\widehat{V}_{K,L})_0 = s_0] > 0$, let $\mu(k) = P[(\widehat{V}_{K,L})_k = a \mid (\widehat{V}_{K,L})_0 = s_0]$. We start by showing that $\mu(k) \leq I_{k > K}(k - K)a(K)$. To that end define

$$\nu(k) = P[(\widehat{V}_{K,L})_k = s_0 \wedge (\widehat{V}_{K,L})_{1:k-1} \neq s_0 \mid (\widehat{V}_{K,L})_0 = s_0], \quad k \geq 1.$$

In words, $\nu(k)$ is the probability that, starting at s_0 , $\widehat{V}_{K,L}$ hits for first time s_0 after k steps. Note that, from the state transition diagram of $\widehat{V}_{K,L}$ (Figure 4), $\nu(k) = 0$ for $k > K$. For $1 \leq k \leq K$ using (17), Proposition 1, and $a(k) = a(k-1) - \sum_{i \in S} \pi_i(k-1)(P_{i,r} + \sum_{j=1}^A P_{i,f_j})$,

$$\begin{aligned}
\nu(k) &= \left(\prod_{i=0}^{k-2} w_i \right) q_{k-1} = a(k-1) \frac{\sum_{i \in S} \pi_i(k-1) P_{i,r}}{a(k-1)} \\
&= a(k-1) \frac{a(k-1) - a(k) - \sum_{i \in S} \pi_i(k-1) \sum_{j=1}^A P_{i,f_j}}{a(k-1)} \\
&\leq a(k-1) \frac{a(k-1) - a(k)}{a(k-1)} = a(k-1) - a(k). \tag{22}
\end{aligned}$$

Since at least $K+1$ steps are required for $\widehat{V}_{K,L}$ to arrive to a from s_0 , $\mu(k) = 0$ for $k \leq K$. For $k \geq K+1$, the result $\mu(k) \leq (k-K)a(K)$ is shown by complete induction on k . The base case is $k = K+1$. For $k = K+1$ the only path from s_0 to a with $K+1$ steps is $(s_0, s_1, \dots, s_K, a)$ and (17)

$$\mu(K+1) = \prod_{i=0}^{K-1} w_i = a(K).$$

Let us proceed with the induction step. Assume $\mu(l) \leq (l-K)a(K)$ for $K+1 \leq l < k$ and let us prove the result for $k > K+1$. Let $i'_{\max} = \min\{K, k-K-1\}$. Classifying the paths of $\widehat{V}_{K,L}$ with k steps which start at s_0 and end in a into those which do so without revisiting s_0 and those which do so by re-entering s_0 for first time after i steps, $1 \leq i \leq i'_{\max}$, using (17), (22), the induction hypothesis, and $a(0) = 1$, noting that $i'_{\max} \geq 1$ (because $K \geq 1$ and $k > K+1$), we have

$$\begin{aligned}
\mu(k) &= \prod_{i=0}^{K-1} w_i + \sum_{i=1}^{i'_{\max}} \nu(i) \mu(k-i) = a(K) + \sum_{i=1}^{i'_{\max}} \nu(i) \mu(k-i) \\
&\leq a(K) + \sum_{i=1}^{i'_{\max}} \left(a(i-1) - a(i) \right) (k-i-K) a(K) \\
&= \left[1 + \sum_{i=1}^{i'_{\max}} \left(a(i-1) - a(i) \right) (k-K-i) \right] a(K) \\
&= \left[1 + a(0)(k-K-1) - \sum_{i=1}^{i'_{\max}-1} a(i) - a(i'_{\max})(k-K-i'_{\max}) \right] a(K)
\end{aligned}$$

$$< [1 + k - K - 1]a(K) = (k - K)a(K),$$

completing the induction step.

Let

$$\psi(k) = P[(\widehat{V}_{K,L})_k = s_0 \wedge (\widehat{V}_{K,L})_{1:k-1} \neq s_0 \mid (\widehat{V}_{K,L})_0 = s'_0], \quad k \geq 1.$$

In words, $\psi(k)$ is the probability that, starting at s'_0 , $\widehat{V}_{K,L}$ hits s_0 for first time after k steps. From the state transition diagram of $\widehat{V}_{K,L}$ (Figure 4), it is clear that $\psi(k) = 0$ for $k > L$. For $1 \leq k \leq L$, using (18), Proposition 1, and $a'(k) = a'(k-1) - \sum_{i \in S'} \pi'_i(k-1)(P_{i,r} + \sum_{j=1}^A P_{i,f_j})$:

$$\begin{aligned} \psi(k) &= \left(\prod_{i=0}^{k-2} w'_i \right) q'_{k-1} = \frac{a'(k-1)}{\alpha_{S'}} \frac{\sum_{i \in S'} \pi'_i(k-1) P_{i,r}}{a'(k-1)} \\ &= \frac{a'(k-1)}{\alpha_{S'}} \frac{a'(k-1) - a'(k) - \sum_{i \in S'} \pi'_i(k-1) \sum_{j=1}^A P_{i,f_j}}{a'(k-1)} \\ &\leq \frac{a'(k-1)}{\alpha_{S'}} \frac{a'(k-1) - a'(k)}{a'(k-1)} = \frac{a'(k-1) - a'(k)}{\alpha_{S'}}. \end{aligned} \quad (23)$$

Let $i''_{\max} = \min\{L, k - K - 1\}$. Classifying the paths of $\widehat{V}_{K,L}$ which hit a through s_K into those which do so starting at s'_0 and hitting s_0 for first time after i steps, $1 \leq i \leq i''_{\max}$, and those which start at s_0 , and using $\mu(k) \leq I_{k>K}(k - K)a(K)$ and (23):

$$\begin{aligned} c_{K,L}(k) &= \alpha_{S'} \sum_{i=1}^{i''_{\max}} \psi(i) \mu(k-i) + \alpha_r \mu(k) \\ &\leq \sum_{i=1}^{i''_{\max}} \left(a'(i-1) - a'(i) \right) (k-i-K)a(K) + \alpha_r I_{k>K}(k-K)a(K). \end{aligned} \quad (24)$$

For $k \leq K+1$, $i''_{\max} \leq 0$ and $\sum_{i=1}^{i''_{\max}} (a'(i-1) - a'(i))(k-i-K)a(K) = 0$. For $k > K+1$, since $L \geq 1$, $i''_{\max} \geq 1$ and, noting that $a'(0) = \alpha_{S'}$,

$$\begin{aligned} &\sum_{i=1}^{i''_{\max}} \left(a'(i-1) - a'(i) \right) (k-i-K)a(K) \\ &= \left[a'(0)(k-K-1) - \sum_{i=1}^{i''_{\max}-1} a'(i) - a'(i''_{\max})(k-K-i''_{\max}) \right] a(K) \\ &< \alpha_{S'}(k-K-1)a(K). \end{aligned}$$

In summary, we have

$$\sum_{i=1}^{i''_{\max}} \left(a'(i-1) - a'(i) \right) (k-i-K)a(K) \leq I_{k>K+1} \alpha_{S'}(k-K-1)a(K).$$

Finally, substituting in (24),

$$\begin{aligned} c_{K,L}(k) &\leq I_{k>K+1} \alpha_{S'}(k-K-1)a(K) + I_{k>K} \alpha_r(k-K)a(K) \\ &\leq I_{k>K} \alpha_{S'}(k-K)a(K) + I_{k>K} \alpha_r(k-K)a(K) = I_{k>K} \alpha_S(k-K)a(K). \end{aligned}$$

It remains to deal with the particular case $\alpha_{S'} = 0$. In that case, $c_K(k) = \alpha_S \mu(k)$ and, as it has been shown, $\mu(k) \leq I_{k>K}(k-K)a(K)$. \square

Proof of Theorem 3 We start by upper bounding the terms in the upper bounds for the error associated with the truncation of the transformed model used in regenerative randomization. Trivially,

$$r_{\max} a'(L) \sum_{k=L+1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} < r_{\max} a'(L). \quad (25)$$

Also,

$$\begin{aligned} r_{\max} \alpha_S a(K) \sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} &= r_{\max} \alpha_S a(K) \sum_{k=1}^{\infty} k e^{-\Lambda t} \frac{(\Lambda t)^{K+k}}{(K+k)!} \\ &< r_{\max} \alpha_S a(K) \sum_{k=1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^{K+k}}{(K+k-1)!} = r_{\max} \alpha_S a(K) \Lambda t \sum_{k=1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^{K+k-1}}{(K+k-1)!} \\ &= r_{\max} \alpha_S a(K) \Lambda t \sum_{k=K}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} < r_{\max} \alpha_S a(K) \Lambda t. \end{aligned} \quad (26)$$

Being all states $i \in S$ of Z transient, $\pi_i(k)$, $i \in S$ decrease geometrically fast [31, Theorem 4.3] and, since $a(k) = \sum_{i \in S} \pi_i(k)$, so does $a(k)$. This implies that there exist $k_0 \geq 0$, $B > 0$ and b , $0 < b < 1$ such that

$$a(k) \leq B b^k, \quad k \geq k_0. \quad (27)$$

Similarly, for $\alpha_{S'} > 0$, using the fact that all states $i \in S'$ of Z' are transient, there exists $k'_0 \geq 0$, $B' > 0$, and b' , $0 < b' < 1$ such that

$$a'(k) \leq B' b'^k, \quad k \geq k'_0. \quad (28)$$

We start by considering the case $\alpha_{S'} > 0$. Using (26), the required K is upper bounded by the minimum integer $k \geq 1$ satisfying $r_{\max} \alpha_S a(k) \Lambda t \leq \varepsilon/4$. Using (27), that integer is not greater than $\max\{k_0, l\}$, where l is the minimum integer ≥ 1 satisfying $r_{\max} \alpha_S B b^l \Lambda t \leq \varepsilon/4$. Then, to prove that K is $O(\log(\Lambda t/\varepsilon))$, it suffices to prove that l is $O(\log(\Lambda t/\varepsilon))$. But l is not greater than $\max\{1, \lfloor x \rfloor + 1\}$, where x satisfies

$$r_{\max} \alpha_S B b^x \Lambda t = \frac{\varepsilon}{4}.$$

Taking logarithms:

$$\log r_{\max} + \log \alpha_S + \log B + x \log b + \log(\Lambda t) = \log\left(\frac{\varepsilon}{4}\right),$$

$$x = \frac{\log(4\Lambda t/\varepsilon) + \log r_{\max} + \log \alpha_S + \log B}{\log(1/b)}.$$

Being $0 < b < 1$, $\log(1/b) > 0$, showing that x , l and K are $O(\log(\Lambda t/\varepsilon))$.

For the case $\alpha_{S'} > 0$, using (25), the required L is upper bounded by the minimum integer $k \geq 1$ satisfying $r_{\max} a'(k) \leq \varepsilon/4$. Using (28), that integer is not greater than $\max\{k'_0, l\}$, where

l is the minimum integer ≥ 1 satisfying $r_{\max} B' b^l \leq \varepsilon/4$. Then, to prove that L is $O(\log(1/\varepsilon))$, it suffices to prove that l is $O(\log(1/\varepsilon))$. But l is not greater than $\max\{1, \lfloor x \rfloor + 1\}$, where x satisfies

$$r_{\max} B' b^x = \frac{\varepsilon}{4}.$$

Taking logarithms:

$$\begin{aligned} \log r_{\max} + \log B' + x \log b' &= \log \left(\frac{\varepsilon}{4} \right), \\ x &= \frac{\log(4/\varepsilon) + \log r_{\max} + \log B'}{\log(1/b')}, \end{aligned}$$

and, being $0 < b' < 1$, $\log(1/b') > 0$, showing that x , l and L are $O(\log(1/\varepsilon))$.

The proof that, for the case $\alpha_{S'} = 0$, the required K is $O(\log(\Lambda t/\varepsilon))$ follows the proof that, for the case $\alpha_{S'} > 0$, the required K is $O(\log(\Lambda t/\varepsilon))$ with ε replaced by 2ε (because the allocated model truncation error is $\varepsilon/2$ instead of $\varepsilon/4$). \square

References

- [1] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions*, Dover, 1964.
- [2] C. Béounes, M. Aguéra, J. Arlat, S. Bachman, C. Bourdeu, J. E. Doucet, K. Kanoun, J. C. Laprie, S. Metge, J. Moreira de Souza, D. Powell and P. Spiesser, “SURF-2: A Program for Dependability Evaluation of Complex Hardware and Software Systems,” in *Proc. 23rd IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-23)*, Toulouse, June 1993, pp. 142–150.
- [3] A. Bobbio and M. Telek, “A Benchmark for PH Estimation Algorithms: Results for Acyclic-PH,” *Communications in Statistics—Stochastic Models*, vol. 10, no. 3, 1994, pp. 661–677.
- [4] P. N. Bowerman, R. G. Nolty and E. M. Schener, “Calculation of the Poisson Cumulative Distribution Function,” *IEEE Trans. on Reliability*, vol. 39, no. 2, 1990, pp. 158–161.
- [5] A. Calderón and J. A. Carrasco, “Computation of Absorption Probability Distributions of continuous-time Markov chains using Regenerative Randomization,” in *Proc. IEEE Int. Computer Performance and Dependability Symposium*, 1995, Erlangen, April 1995, pp. 92–101.
- [6] J. A. Carrasco and A. Calderón, “Regenerative Randomization: Theory and application examples,” *Performance Evaluation Review*, vol. 23, no. 1, May 1995, pp. 241–252.
- [7] G. Ciardo, J. Muppala and K. Trivedi, “SPNP: Stochastic Petri Net Package,” in *Proc. 3rd IEEE Int. Workshop on Petri Nets and Performance Models, PNPM89*, Kyoto, December 1989, pp. 142–151.
- [8] J. Couvillon, R. Freire, R. Johnson, W. Obal II, A. Qureshi, M. Rai, W. Sanders and J. Tvedt, “Performability modelling with UltraSAN,” *IEEE Software*, September 1991, pp. 69–80.
- [9] B. L. Fox and P. W. Glynn, “Computing Poisson probabilities,” *Communications of the ACM*, vol. 31, 1988, pp. 440–445.
- [10] A. Goyal, W. C. Carter, E. de Souza e Silva and S. S. Lavenberg, “The System Availability Estimator,” in *Proc. 16th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-16)*, July 1986, pp. 84–89.
- [11] W. K. Grassman, “Transient solutions in Markovian queuing systems,” *Computers and Operations Research*, vol. 4, 1977, pp. 47–53.

- [12] W. K. Grassman, "Rounding Errors in Certain Algorithms Involving Markov Chains," *ACM Transactions on Mathematical Software*, vol. 19, no. 4, December 1993, pp. 496–508.
- [13] D. Gross and D. R. Miller, "The randomization technique as a modelling tool and solution procedure for transient Markov processes," *Operations Research*, vol. 32, 1984, pp. 343–361.
- [14] B. W. Johnson, *Design and Analysis of Fault Tolerant Digital Systems*, Addison-Wesley, 1989.
- [15] M. Kijima, *Markov Processes for Stochastic Modeling*, University Press, Cambridge, 1997.
- [16] L. Knüsel, "Computation of the Chi-square and Poisson Distribution," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, July 1986, pp. 1022–1036.
- [17] M. Malhotra, J. K. Muppala and K. S. Trivedi, "Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains," *Microelectronics and Reliability*, vol. 34, no. 11, 1994, pp. 1825–1841.
- [18] M. Malhotra, "A computationally efficient technique for transient analysis of repairable Markovian systems," *Performance Evaluation*, November 1995, no. 1–2, pp. 311–331.
- [19] B. Melamed and M. Yadin, "Randomization Procedures in the Computation of Cumulative-Time Distributions over Discrete State Markov Processes," *Operations Research*, vol. 32, 1984, pp. 926–944.
- [20] D. R. Miller, "Reliability Calculation using Randomization for Markovian Fault-Tolerant Computing Systems," in *Proc. 13th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-13)*, June 1983, pp 284–289.
- [21] A. P. Moorsel and W. H. Sanders, "Adaptive Uniformization", *Communications in Statistics–Stochastic Models*, vol. 10, no. 3, 1994, pp. 619–647.
- [22] A. P. Moorsel and W. H. Sanders, "Transient Solution of Markov Models by Combining Adaptive and Standard Uniformization," *IEEE Trans. on Reliability*, vol. 46, no. 3, 1997, pp. 430–440.
- [23] H. Nabli and B. Sericola, "Performability Analysis: A New Algorithm," *IEEE Trans. on Computers*, vol. 45, no. 4, April 1996, pp. 491–494.
- [24] F. Neuts, *Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach*, Dover Publications Inc., New York, 1994.
- [25] M. A. Qureshi and W. H. Sanders, "Reward Model Solution Methods with Impulse and Rate Rewards: An Algorithm and Numerical Results," *Performance Evaluation*, vol. 20, 1994, pp. 413–436.
- [26] A. Reibman and K. S. Trivedi, "Numerical Transient Analysis of Markov Models," *Computers and Operations Research*, vol. 15, 1988, pp. 19–36.
- [27] S. M. Ross, *Stochastic Processes*, John Wiley & Sons, 1983.
- [28] G. Rubino and B. Sericola, "Interval Availability Distribution Computation," in *Proc. 23th Int. Symp. on Fault-Tolerant Computing (FTCS-23)*, Toulouse, June 1993, pp. 48–55.
- [29] G. Rubino and B. Sericola, "Interval Availability Analysis Using Denumerable Markov Processes: Application to Multiprocessor Subject to Breakdowns and Repair," *IEEE Trans. on Computers*, vol. 44, no. 2, February 1995, pp. 286–291.
- [30] J. R. Schott, *Matrix Analysis for Statistics*, John Wiley & Sons, Inc., 1997.
- [31] E. Seneta, *Non-negative Matrices and Markov Chains*, Springer Series in Statistics, Springer-Verlag, 2nd edition, 1981.

- [32] B. Sericola, "Availability Analysis of Repairable Computer Systems and Stationarity Detection," *IEEE Trans. on Computers*, vol. 48, no. 11, November 1999, pp. 1166–1172.
- [33] E. de Souza e Silva and H. R. Gail, "Calculating Cumulative Operational Time Distributions of Repairable Computer Systems," *IEEE Trans. on Computers*, vol. 35, 1986, pp. 322–332.
- [34] E. de Souza e Silva and H. R. Gail, "Calculating Availability and Performability Measures of Repairable Computer Systems using Randomization," *Journal of the ACM*, vol. 34, 1989, pp. 179–199.
- [35] E. de Souza e Silva and H. R. Gail, "An Algorithm to Calculate Transient Distributions of Cumulative Rate and Impulse Based Reward," *Communications in Statistics–Stochastic Models*, vol. 14, no. 3, 1998, pp. 509–536.
- [36] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*, McGraw-Hill, 1970.
- [37] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Inc., 1962.