

Computationally Efficient and Numerically Stable Reliability Bounds for Repairable Fault-Tolerant Systems

Juan A. Carrasco, *Member, IEEE*

Abstract—The transient analysis of large continuous time Markov reliability models of repairable fault-tolerant systems is computationally expensive due to model stiffness. In this paper, we develop and analyze a method to compute bounds for a measure defined on a particular, but quite wide, class of continuous time Markov models, encompassing both exact and bounding continuous time Markov reliability models of fault-tolerant systems. The method is numerically stable and computes the bounds with well-controlled and specifiable-in-advance error. Computational effort can be traded off with bounds accuracy. For a class of continuous time Markov models, class C'' , including typical failure/repair reliability models with exponential failure and repair time distributions and repair in every state with failed components, the method can yield reasonably tight bounds at a very small computational cost. The method builds upon a recently proposed numerical method for the transient analysis of continuous time Markov models called regenerative randomization.

Index Terms—Fault-tolerant systems, repairable systems, reliability, continuous time Markov models, bounds, randomization.

1 INTRODUCTION

INCREASING demand for system dependability has created great interest in fault-tolerant systems. In many applications, e.g., critical applications, an appropriate measure to quantify a system's dependability is the reliability, defined as the probability that the system has not failed by time t , or, alternatively, the complementary unreliability measure, $wr(t)$, defined as the probability that the system has failed by time t . Homogeneous continuous time Markov chain (CTMC) models are commonly used to predict the unreliability of fault-tolerant systems, particularly when the system is repairable. Computation of the unreliability then requires the transient analysis of the CTMC model. Available numerical methods to perform that transient analysis include ODE (ordinary differential equation) solvers and randomization (also called uniformization) [12], [13], [19]. The randomization method is attractive because it is numerically stable and the computation error is well controlled and can be specified in advance. However, the performance of randomization is seriously affected by model stiffness. For CTMC models, a practical measure of stiffness is Λt [19], where Λ is the maximum output rate of the model. For large Λt , randomization requires a number of steps $\approx \Lambda t$ and will be highly inefficient if the model is large.

CTMC reliability models of repairable fault-tolerant systems tend to be very stiff when the mission time of interest is large. To illustrate the point, Fig. 1 shows a small CTMC reliability model $X = \{X(t); t \geq 0\}$ of a repairable fault-tolerant system using the pair-and-spare technique [9] 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 unreliability of the system is $wr(t) = P[X(t) = f]$. For the model, $\Lambda = 2\mu_S + \lambda_M \approx 120 \text{ h}^{-1}$ and, for a mission time $t = 1 \text{ year} = 8,760 \text{ h}$, $\Lambda t \approx 1,051,200$.

Several variants of the (standard) randomization method have been proposed to improve its efficiency: selective randomization [14], [15], multisteping [19, Section 3.1.2], adaptive uniformization [16], adaptive/standard uniformization [17], uniformization with steady-state detection [12], [21], and regenerative randomization [5], [6]. For large CTMC reliability models of repairable fault-tolerant systems and long mission times, regenerative randomization seems to be the best of them. The method has the same good properties as the standard randomization method (numerical stability, well-controlled computation error, and ability to specify the computation error in advance) and can be much faster than standard randomization.

The regenerative randomization method covers CTMC models $X = \{X(t); t \geq 0\}$ with state space

$$\Omega = S \cup \{f_1, f_2, \dots, f_A\}, \quad |S| \geq 2, \quad A \geq 0,$$

where f_i are absorbing states and either 1) all states in S are transient or 2) S has a single trapping component¹ and the chosen regenerative state $r \in S$ belongs to that component, and all states are reachable from some state with nonnull initial probability. It is also assumed that X

• The author is with the Department d'Enginyeria Electrònica, Universitat Politècnica de Catalunya, Diagonal 647, plta. 9, 08028 Barcelona, Spain. E-mail: carrasco@eel.upc.es.

Manuscript received 1 Apr. 2000; revised 21 Feb. 2001; accepted 7 Apr. 2001. For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 111155.

1. Two states i, j of a CTMC are strongly connected if there are paths in the state transition diagram of the CTMC from i to j and from j to i ; a state i is strongly connected with itself; a component is a maximal subset of strongly connected states; a component is trapping if no state of the component has transition rates to states outside the component.

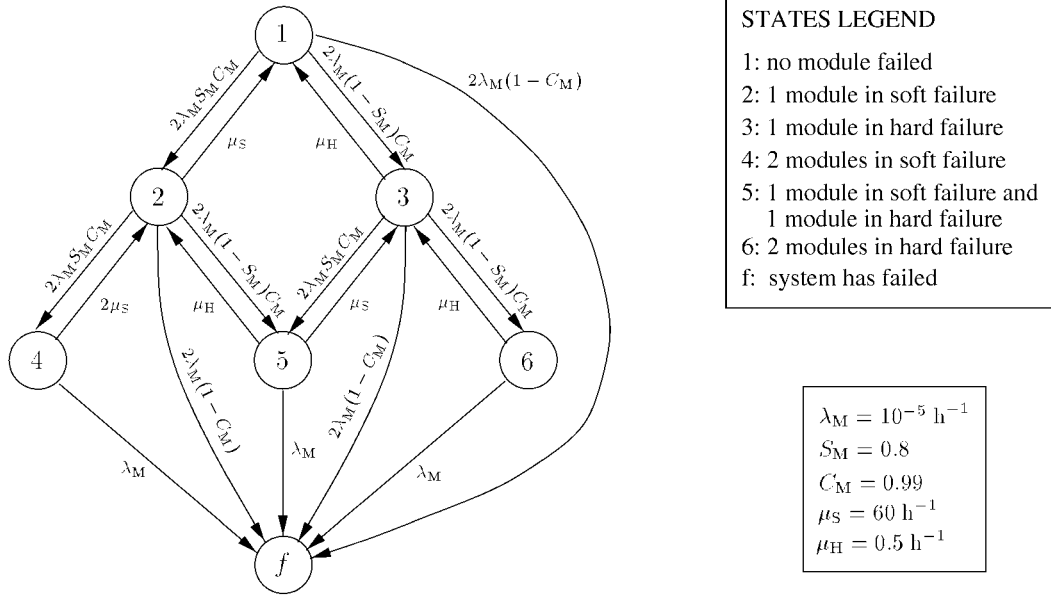


Fig. 1. CTMC reliability model of a repairable fault-tolerant system using the pair-and-spares technique.

has some transition rate from r to $S' - \{r\}$, although that condition can be easily circumvented in practice [5]. The generic measure considered in [5] is

$$m(t) = \sum_{i=1}^A r_{f_i} P[X(t) = f_i],$$

$A \geq 1$, where r_{f_i} are different reward rates ≥ 0 (in [6], more general measures are considered and $A \geq 0$ is allowed). Those models with $A \geq 1$ and the generic measure $m(t)$ cover both exact and bounding CTMC reliability models of fault-tolerant systems (bounding models are useful when an exact model would have an unmanageable size). In an exact reliability model, A would be equal to 1, S would include all operational states, entry in f_1 would represent the failure of the system, r_{f_1} would be equal to 1, the initial probability of f_1 would be equal to the probability of the system being initially failed, and $m(t)$ would be the unreliability $ur(t)$ (an example of such an exact reliability model is the model given in Fig. 1 with $S = \{1, 2, 3, 4, 5, 6\}$ and $f_1 = f$). In a lower bounding reliability model, A would be equal to 2, S would be a proper subset of the set of operational states, entry in f_1 would represent the failure of the system from a state in S , entry in f_2 would represent entry in an operational state outside S , r_{f_1} would be equal to 1, r_{f_2} would be equal to 0, the initial probability of f_1 would be equal to the probability of the system being initially failed, the initial probability of f_2 would be the probability of the system being initially in an operational state outside S , and $m(t)$ would be a lower bound for $ur(t)$. Finally, in an upper bounding reliability model, A would be equal to 1, S would be a proper subset of the set of operational states, entry in f_1 would represent exit from S , r_{f_1} would be equal to 1, the initial probability of f_1 would be the probability of the system being initially either failed or in an operational state outside S , and $m(t)$ would be an upper bound for $ur(t)$. The regenerative randomization method requires the selection of a *regenerative state* $r \in S$. The performance of the method depends on that selection.

In this paper, we consider CTMC models with the same structure and properties as the models considered in regenerative randomization with $A \geq 1$ and develop a method called *bounding regenerative randomization* to obtain bounds for the generic measure $m(t)$. The method yields a lower bound for $m(t)$, an upper bound for $m(t)$, or both. The lower bound is obtained by solving, by regenerative randomization, a lower bounding CTMC, X^{lb} . The upper bound is obtained by solving, by regenerative randomization, an upper bounding CTMC, X^{ub} . Both X^{lb} and X^{ub} are obtained from X by scaling some of its transition rates. The method has the same good properties as standard randomization.

Although not restricted to them, the bounding regenerative randomization method is intended to be used for a class of models C'' . Let $\lambda_{i,j}$ denote the transition rate of X from state i to state j , $i \neq j$, let $\lambda_i = \sum_{j \in \Omega - \{i\}} \lambda_{i,j}$ denote the output rate from state i , and let $\lambda_{i,B} = \sum_{j \in B} \lambda_{i,j}$, $B \subset \Omega - \{i\}$. Class C'' includes the models X with the properties assumed in the regenerative randomization method with $A \geq 1$ for which there exists a partition $S_0 \cup S_1 \cup \dots \cup S_{N_C}$ for S satisfying the following three 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

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

- P3. $\lambda_o \leq \min_{i \in S - \{o\}} \lambda_i$.

The class 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), such as the model given in Fig. 1. For those models, a partition for which properties P1, P2, and P3 are satisfied is $S_k = \{\text{states in } S \text{ with } k \text{ failed components}\}$. The class also

covers failure/repair reliability models with exponential failure time distributions, repair times with acyclic phase-type distributions [18] (which can be used to fit distributions of nonexponential positive random variables [3]), and repair in every state with failed components, provided that the transition rates of the transient CTMCs defining the phase-type distributions are sufficiently large compared with failure rates. For those models, the proposed method can be extremely efficient and, yet, provide quite tight bounds. Tighter bounds can be obtained at the cost of increased computational effort.

An approach to deal with stiffness is the aggregation technique proposed in [2]. For class C'' models, that technique could be used to aggregate the states in $S - \{o\}$, yielding an aggregated CTMC model with a transient state and A absorbing states with symbolic solution. The aggregation would be done by replacing each state in $S - \{o\}$ by a switch and is equivalent to scale the transition rates $\lambda_{i,j}$, $i \in S - \{o\}$ with $\lambda_i \rightarrow \infty$, keeping the relative values of the transition rates from a given state. The aggregated model would give an upper bound for the measure $m(t)$ looser than the upper bounds that are computed by the bounding regenerative randomization method proposed in this paper.

The rest of the paper is organized as follows: Section 2 presents a brief review of both the standard randomization and the regenerative randomization methods (the latter particularized to the computation of the measure $m(t)$), including algorithmic descriptions for both methods. Section 3 describes the proposed bounding regenerative randomization method, proves that it yields bounds for the measure $m(t)$, and gives theoretical results assessing the efficiency of the method for class C'' models. Section 4 analyzes the performance of the bounding regenerative randomization method using a large reliability model belonging to class C'' and compares the computational cost of the method with that of regenerative randomization and standard randomization. Finally, Section 5 concludes the paper.

2 REVIEW OF STANDARD AND REGENERATIVE RANDOMIZATION

The review of the standard randomization method will be made for arbitrary rewarded CTMC models $X = \{X(t); t \geq 0\}$ with finite state space Ω and for the expected transient reward rate measure

$$ETRR(t) = E[r_{X(t)}] = \sum_{i \in \Omega} r_i P[X(t) = i],$$

where $r_i \geq 0$, $i \in \Omega$ is the reward rate associated with state i . The quantity r_i has the meaning of "rate" at which reward is earned while X is in state i . The measure $m(t)$ is a particular case of $ETRR(t)$. The standard randomization method is based on the following result (see, for instance, [10, Theorem 4.19]). 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 $P_{i,j} = \lambda_{i,j}/\Lambda$, $i \neq j$, $P_{i,i} = 1 - \lambda_i/\Lambda$. The DTMC \hat{X} is called the *randomized* DTMC of X with randomization rate Λ . The CTMC X is said to be the *derandomized* CTMC of \hat{X} with randomization rate Λ . Let $Q = \{Q(t); t \geq 0\}$ be a

Poisson process with arrival rate Λ independent of \hat{X} ($P[Q(t) = k] = e^{-\Lambda t}(\Lambda t)^k/k!$). Then, $X = \{X(t); t \geq 0\}$ is probabilistically identical to $\{\hat{X}_{Q(t)}; t \geq 0\}$. That result allows expressing $ETRR(t)$ in terms of the transient regime of \hat{X} as:

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

with $d(k) = \sum_{i \in \Omega} r_i P[\hat{X}_k = i]$. Let $\alpha = (P[X(0) = i])_{i \in \Omega}$ be the initial probability row vector of X and let $\mathbf{q}(k) = (P[\hat{X}_k = i])_{i \in \Omega}$ be the probability row vector of \hat{X} at step k . We have $\mathbf{q}(0) = \alpha$. From $\mathbf{q}(0)$, $\mathbf{q}(k)$, $k > 0$ can be obtained using $\mathbf{q}(k+1) = \mathbf{q}(k)\mathbf{P}$, where $\mathbf{P} = (P_{i,j})_{i,j \in \Omega}$ is the transition probability matrix of \hat{X} .

An approximate value for $ETRR(t)$, $ETRR_N^a(t)$, can be obtained by truncating series (1):

$$ETRR_N^a(t) = \sum_{k=0}^N d(k) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}. \quad (2)$$

Using $d(k) \leq r_{\max} = \max_{i \in \Omega} r_i$, the truncation error can be upper bounded as

$$ETRR(t) - ETRR_N^a(t) \leq r_{\max} \sum_{k=N+1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}. \quad (3)$$

Then, ε being the allowed error for the computation of $ETRR(t)$, in the standard randomization method N is chosen as

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

and $ETRR(t)$ is approximated with error $\leq \varepsilon$ by the $ETRR_N^a(t)$ given by (2). The computational cost of standard randomization is essentially the cost of performing the N vector-matrix multiplications $\mathbf{q}(k+1) = \mathbf{q}(k)\mathbf{P}$, $k = 0, 1, \dots, N-1$. $Q(t)$ has, for $\Lambda t \rightarrow \infty$, an asymptotic normal distribution with mean and variance Λt [20], and, for large Λt and $\varepsilon \ll 1$, the required N is $\approx \Lambda t$, making standard randomization computationally very expensive if both X and Λt are large. Since the performance of standard randomization degrades as Λ increases, Λ is usually taken equal to $\max_{i \in \Omega} \lambda_i$. An algorithmic description of the standard randomization method is given in Fig. 2. The algorithm has as inputs the CTMC X , the reward rates r_i , $i \in \Omega$, the initial probability row vector α , the allowed error ε , the number of time points n at which $ETRR(t)$ has to be computed, and the time points t_1, t_2, \dots, t_n . The algorithm has as outputs the computed

Inputs: $X, r_i, i \in \Omega, \alpha, \varepsilon, n, t_1, t_2, \dots, t_n$

Outputs: $\widetilde{ETRR}(t_1), \widetilde{ETRR}(t_2), \dots, \widetilde{ETRR}(t_n)$

$r_{\max} = \max_{i \in \Omega} r_i;$

$t_{\max} = \max\{t_1, t_2, \dots, t_n\};$

$\Lambda = \max_{i \in \Omega} \lambda_i;$

Obtain $\mathbf{P};$

$N = \min\{m \geq 0 : r_{\max} \sum_{k=m+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k! \leq \varepsilon\};$

$\mathbf{q} = \alpha; d(0) = \sum_{i \in \Omega} r_i q_i;$

for ($k = 1; k \leq N; k++$) {

$\mathbf{nq} = \mathbf{qP}; \mathbf{q} = \mathbf{nq};$

$d(k) = \sum_{i \in \Omega} r_i q_i;$

}

for ($i = 1; i \leq n; i++$)

for ($k = 0, \widetilde{ETRR}(t_i) = 0; k \leq N; k++$) $\widetilde{ETRR}(t_i) += d(k)e^{-\Lambda t_i} (\Lambda t_i)^k / k!;$

Fig. 2. Algorithmic description of standard randomization.

values of $ETRR(t), \widetilde{ETRR}(t_1), \widetilde{ETRR}(t_2), \dots, \widetilde{ETRR}(t_n)$. The truncation error bound given by (3) increases with t and, therefore, that error is controlled for

$$t_{\max} = \max\{t_1, t_2, \dots, t_n\}.$$

We review next the regenerative randomization method for the CTMC models X considered in this paper and the measure $m(t)$. Let $S' = S - \{r\}$ and let $\alpha_i = P[X(0) = i], i \in \Omega$. We will use the notation $\alpha_B = \sum_{i \in B} \alpha_i$. In the method, the behavior of X from S' up to state r or a state f_i and from r until the next hit of r or a state f_i is approximately characterized by a truncated transformed model from which an approximate value with bounded error for $m(t)$ can be computed and that approximate value is computed solving the truncated transformed model by the standard randomization method. To build the truncated transformed model, two DTMCs, Z and Z' [5], obtained from the randomized DTMC \hat{X} of X with rate Λ and a version \hat{X}' of \hat{X} in which the initial probability distribution is concentrated in state r , have to be stepped in general. The randomization rate Λ is taken slightly larger than $\max_{i \in S} \lambda_i$ (i.e., $\Lambda = (1 + \theta) \max_{i \in S} \lambda_i$, θ being a small value, say 10^{-4}). This simplifies considerably the description and implementation of the method and has negligible impact on its performance. The transition probability matrix of \hat{X} will be denoted as before by $\mathbf{P} = (P_{i,j})_{i,j \in \Omega}$. The DTMC $Z = \{Z_k; k = 0, 1, 2, \dots\}$ follows \hat{X} from r till reentry in r . Z has state space $S \cup \{f_1, f_2, \dots, f_A, a\}$, where f_i and a are absorbing states and all states in S are transient, initial state r , and its (possibly nonnull) transition probabilities are:

$$\begin{aligned} P[Z_{k+1} = j | Z_k = i] &= P_{i,j}, i \in S, j \in S' \cup \{f_1, f_2, \dots, f_A\}, \\ P[Z_{k+1} = a | Z_k = i] &= P_{i,r}, i \in S, \\ P[Z_{k+1} = f_i | Z_k = f_i] &= P[Z_{k+1} = a | Z_k = a] = 1, 1 \leq i \leq A. \end{aligned}$$

The DTMC $Z' = \{Z'_k; k = 0, 1, 2, \dots\}$ follows \hat{X}' until its first visit to state r . Z' has state space $S' \cup \{f_1, f_2, \dots, f_A, a\}$, where f_i and a are absorbing states and all states in S' are transient. The initial probability distribution of Z' is $P[Z'_0 = i] = \alpha_i, i \in S' \cup \{f_1, f_2, \dots, f_A\}, P[Z'_0 = a] = \alpha_r$, and its (possibly nonnull) transition probabilities are:

$$\begin{aligned} P[Z'_{k+1} = j | Z'_k = i] &= P_{i,j}, i \in S', j \in S' \cup \{f_1, f_2, \dots, f_A\}, \\ P[Z'_{k+1} = a | Z'_k = i] &= P_{i,r}, i \in S', \\ P[Z'_{k+1} = f_i | Z'_k = f_i] &= P[Z'_{k+1} = a | Z'_k = a] = 1, 1 \leq i \leq A. \end{aligned}$$

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

$$\begin{aligned} a(k) &= \sum_{i \in S} \pi_i(k), \\ v_k^j &= \sum_{i \in S} \pi_i(k) P_{i,f_j} / a(k), \\ q_k &= \sum_{i \in S} \pi_i(k) P_{i,r} / a(k), \\ w_k &= \sum_{i \in S} \pi_i(k) P_{i,S'} / a(k), \end{aligned}$$

and, if $\alpha_{S'} > 0$, let

$$\begin{aligned} a'(k) &= \sum_{i \in S'} \pi'_i(k), \\ v_k^j &= \sum_{i \in S'} \pi'_i(k) P_{i,f_j} / a'(k), \\ q_k &= \sum_{i \in S'} \pi'_i(k) P_{i,r} / a'(k), \\ w_k &= \sum_{i \in S'} \pi'_i(k) P_{i,S'} / a'(k), \end{aligned}$$

where $P_{i,S'} = \sum_{j \in S'} P_{i,j}$ ($a(k)$ and, if $\alpha_{S'} > 0$, $a'(k)$ are guaranteed to be > 0 (see [5])). Then, for the case $\alpha_{S'} > 0$,

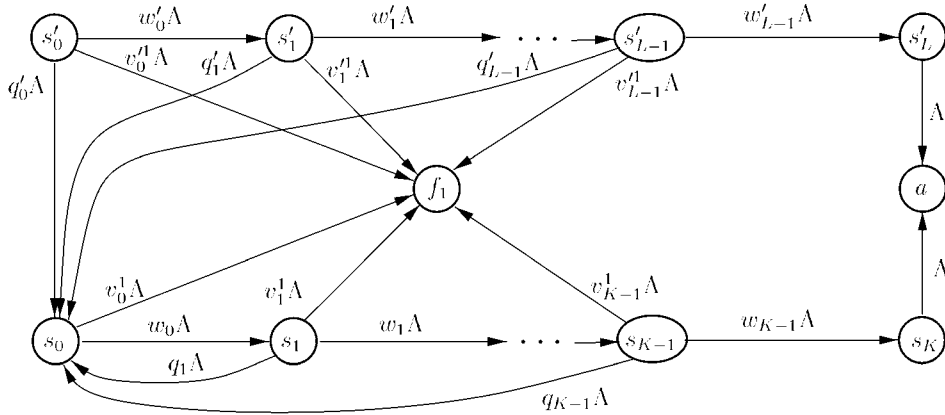


Fig. 3. State transition diagram of the CTMC $V_{K,L}$ for $A = 1$.

the truncated transformed model is the CTMC $V_{K,L} = \{V_{K,L}(t); t \geq 0\}$ with state space

$$\{s_k, 0 \leq k \leq K\} \cup \{s'_k, 0 \leq k \leq L\} \cup \{f_1, f_2, \dots, f_A, a\},$$

initial probability distribution

$$\begin{aligned} P[V_{K,L}(0) = s_0] &= \alpha_r, \\ P[V_{K,L}(0) = s'_0] &= \alpha_{s'}, \\ P[V_{K,L}(0) = f_i] &= \alpha_{f_i}, \\ P[V_{K,L}(0) = i] &= 0, \quad i \notin \{s_0, s'_0, f_1, f_2, \dots, f_A\}, \end{aligned}$$

and the state transition diagram illustrated in Fig. 3 for $A = 1$. For the case $\alpha_{s'} = 0$, the truncated transformed model is the CTMC $V_K = \{V_K(t); t \geq 0\}$ with initial probability distribution $P[V_K(0) = s_0] = \alpha_S$, $P[V_K(0) = f_i] = \alpha_{f_i}$, $P[V_K(0) = i] = 0$, $i \notin \{s_0, f_1, f_2, \dots, f_A\}$ and a state transition diagram identical to the state transition diagram of $V_{K,L}$, but without states s'_k .

For the case $\alpha_{s'} > 0$, the approximate value for $m(t)$ given by $V_{K,L}$ is

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

and we have

$$\begin{aligned} m(t) - m_{K,L}^a(t) &\leq r_{\max} a'(L) \sum_{k=L+1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^k}{k!} \\ &\quad + r_{\max} \alpha_S a(K) \sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}, \end{aligned} \quad (4)$$

where $r_{\max} = \max_{1 \leq i \leq A} r_{f_i}$. For the case $\alpha_{s'} = 0$, the approximate value for $m(t)$ given by V_K is:

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

and we have

$$m(t) - m_K^a(t) \leq r_{\max} \alpha_S a(K) \sum_{k=K+1}^{\infty} (k-K) e^{-\Lambda t} \frac{(\Lambda t)^k}{k!}. \quad (5)$$

The model truncation error bounds given by (4) and (5) decrease for increasing K and L and can be made arbitrarily small by choosing large enough values for K and L . In regenerative randomization, ε being the allowed error for the computation of $m(t)$, suitable truncation parameters K, L are chosen so that the model truncation error bounds are smaller than $\varepsilon/2$ and, then, an approximate value for $m(t)$ is obtained by computing $m_{K,L}^a(t)$ ($m_K^a(t)$) by solving the truncated transformed model $V_{K,L}$ (V_K) by standard randomization with error upper bounded by $\varepsilon/2$.

An algorithmic description of the regenerative randomization method is given in Fig. 4, where I_c denotes the indicator function returning the value 1 if condition c is satisfied and the value 0 otherwise. The algorithm has as inputs the CTMC X , the number A of absorbing states f_i , the reward rates $r_{f_1}, r_{f_2}, \dots, r_{f_A}$, an initial probability distribution vector $\alpha = (\alpha_i)_{i \in \Omega}$ with $\alpha_S > 0$, 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 computed values of $m(t)$, $\tilde{m}(t_1), \tilde{m}(t_2), \dots, \tilde{m}(t_n)$. Since the model truncation error bounds increase with t , they are controlled for $t_{\max} = \max\{t_1, t_2, \dots, t_n\}$. For the case $\alpha_{s'} > 0$, the $\varepsilon/2$ allocated for the model truncation error bound is divided equally between its two contributions. The truncation error bound associated with the solution of the truncated transformed model by standard randomization also increases with t and that error is controlled for t_{\max} . The method requires stepping the randomized DTMC $\hat{V}_{K,L}$ (\hat{V}_K) of $V_{K,L}$ (V_K) with rate Λ . The state transition diagram of $\hat{V}_{K,L}$ is illustrated in Fig. 5 for $A = 1$. The state transition diagram of \hat{V}_K is identical, but without the states s'_k .

The regenerative randomization method (as the standard randomization method) requires the computation of the Poisson probabilities $e^{-\Lambda t} (\Lambda t)^k / k!$. Stable and efficient computation of those Poisson probabilities, avoiding overflows and intermediate underflows, is a delicate issue and

```

Inputs:  $X, A, r_{f_1}, r_{f_2}, \dots, r_{f_A}, \alpha, r, \varepsilon, n, t_1, t_2, \dots, t_n$ 
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 ( $\alpha_{S'} > 0$ )  $tol_K = \varepsilon/A$ ; else  $tol_K = \varepsilon/2$ ;
 $\boldsymbol{\pi} = (l_{i=r})_{i \in S}; a = 1; K = 0;$ 
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 ( $r_{\max} a' \sum_{k=L+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k! \leq \varepsilon/4$ );
}
 $N = \min\{m \geq 0 : r_{\max} \sum_{k=m+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k! \leq \varepsilon/2\};$ 
if ( $\alpha_{S'} > 0$ )
  Give  $N$  steps to  $\hat{V}_{K,L}$  and compute  $d(k) = \sum_{i=1}^A r_{f_i} P[(\hat{V}_{K,L})_k = f_i], k = 0, 1, \dots, N;$ 
else
  Give  $N$  steps to  $\hat{V}_K$  and compute  $d(k) = \sum_{i=1}^A r_{f_i} P[(\hat{V}_K)_k = f_i], k = 0, 1, \dots, N;$ 
for ( $i = 1; i \leq n; i++$ )
  for ( $k = 0, \tilde{m}(t_i) = 0; k \leq N; k++$ )  $\tilde{m}(t_i) += d(k) e^{-\Lambda t_i} (\Lambda t_i)^k / k!;$ 

```

Fig. 4. Algorithmic description of regenerative randomization.

several alternatives have been proposed [4], [8], [11], [17]. The method described in [11, pp. 1028–1029] (see also [1]) has good numerical stability and is the one we follow in our implementations. The regenerative randomization method involves the computation of

$$S(m) = \sum_{k=m+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k!$$

and

$$S'(m) = \sum_{k=m+1}^{\infty} (k-m) e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k!$$

for increasing values of m (the standard randomization method also requires the computation of $S(m)$ for increasing values of m). Our implementations use the algorithms described in [5], which are numerically stable and efficient.

The computational cost of regenerative randomization has two components: cost associated with the construction of the truncated transformed model and cost associated with the solution of the truncated transformed model by standard randomization. The first is roughly proportional

to the number of steps on the DTMCs $Z, Z', K + L$ if $\alpha_{S'} > 0$ and K if $\alpha_{S'} = 0$, with a cost per step which, for large X , will typically be slightly larger than the cost per step in standard randomization. The second component is roughly proportional to the truncation parameter N (approximately equal to the truncation parameter N of standard randomization) and to the size of the truncated transformed model. It is shown in [5] that the required K is $O(\log(\Lambda t/\varepsilon))$ and, if $\alpha_{S'} > 0$, the required L is $O(\log(1/\varepsilon))$. That is called “benign” behavior and implies that, for large enough X and large enough Λt , regenerative randomization will be significantly faster than standard randomization.

The performance of regenerative randomization depends, of course, on the selection of the regenerative state r . That selection should be made so that $a(k)$ and $a'(k)$ decrease as fast as possible and the required K and L are as small as possible. Since class C'' models move fast to either state o or an absorbing state f_i , a natural selection for those models is $r = o$. Let $R' = \max_{i \in S - \{o\}} \lambda_i / \min_{i \in S - \{o\}} \lambda_i$. Then, we can state the following result:

Theorem 1. For class C'' , models with selection $r = o$, $a(k) \leq h(k)$ and $a'(k) \leq \alpha_{S'} h'(k)$, where, for $k \rightarrow \infty$,

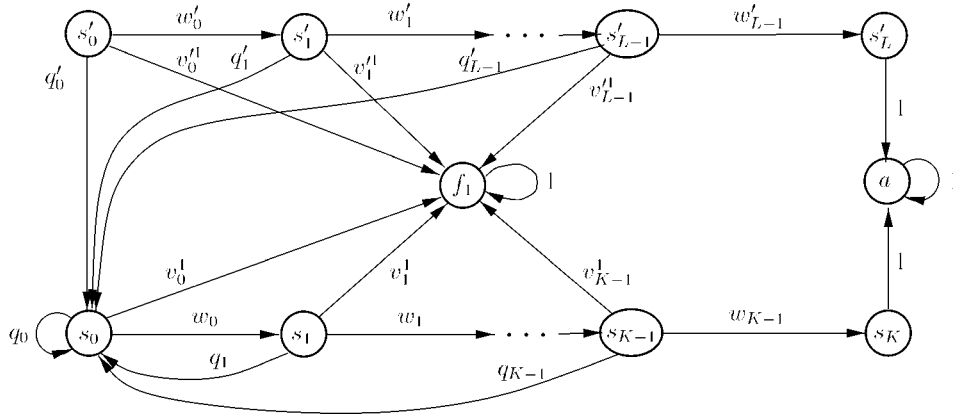


Fig. 5. State transition diagram of the DTMC $\hat{V}_{K,L}$ for $A = 1$.

$$h(k) \sim B\binom{k}{p-1} \rho^k,$$

and

$$h'(k) \sim B'\binom{k}{p'-1} \rho'^k,$$

with $B > 0$, $B' > 0$, p, p' integers ≥ 1 , $\rho \approx 1 - 1/R'$, and $\rho' \approx 1 - 1/R'^2$.

Proof. The model class C'' is a subset of the model class C considered in [5]. In [5], it is considered the parameter $R = \max_{i \in S} \lambda_i / \min_{i \in S'} \lambda_i$. For class C'' models, it follows from Property P3 that, with selection $r = o$, $R = R'$ and the result follows from Theorem 4 of [5] and the discussion following it. \square

Theorem 1 asserts that, for class C'' models, the performance of regenerative randomization with the natural selection $r = o$ should be mainly determined by the parameter R' : the larger R' , the more costly the method. In particular, for $R' = 1$, $\rho \approx 0$ and $\rho' \approx 0$ and the method should be very efficient. Those observations motivate the bounding regenerative randomization method.

3 THE BOUNDING REGENERATIVE RANDOMIZATION METHOD

The bounding regenerative randomization method obtains a lower bound for $m(t)$, an upper bound for $m(t)$, or both. The bounds are computed with an error upper bounded by a ε given by the user. Depending on the nature of the CTMC model X , one or the other bound or both bounds could be of interest. Thus, if X is an exact reliability model, both bounds would be of interest to have an assessment of the error on $ur(t)$. However, if an exact reliability model cannot be used because its size would be unmanageable, then we could use a lower bounding reliability model and an upper bounding reliability model and use bounding regenerative randomization to compute a lower bound for the lower bound for the unreliability given by the first model and an upper bound for the upper bound for the unreliability given by the second model: the exact unreliability would be bracketed by those values.

2. $c(k) \sim d(k)$ for $k \rightarrow \infty$ denotes $\lim_{k \rightarrow \infty} c(k)/d(k) = 1$.

The bounding regenerative randomization method requires the selection of a regenerative state $r \in S$ and has an input parameter D controlling the accuracy of the bounds. Let $\lambda_{\min} = \min_{i \in S'} \lambda_i$ and $\lambda_{\max} = \max_{i \in S'} \lambda_i$. The method assumes that the controlling parameter D is restricted by $1 \leq D < \lambda_{\max}/\lambda_{\min}$.³ To obtain the lower bound for $m(t)$, the method modifies the CTMC X to obtain a CTMC X^{lb} . The CTMC X^{lb} is obtained from X by scaling the transition rates from states in S' so that, calling λ_i^{lb} the output rates of X^{lb} , $\lambda_i^{\text{lb}} \leq \lambda_i$, $i \in S'$ and $\max_{i \in S'} \lambda_i^{\text{lb}} / \min_{i \in S'} \lambda_i^{\text{lb}} = D$. That scaling is defined by $\lambda_{i,j}^{\text{lb}} = \lambda_{i,j} (\lambda_i^{\text{lb}} / \lambda_i)$, $\lambda_i^{\text{lb}} = \min\{\lambda_i, D\lambda_{\min}\}$, $i \in S'$, where $\lambda_{i,j}^{\text{lb}}$ are the transition rates in X^{lb} . The lower bound for $m(t)$ is given by

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

That lower bound is obtained by solving X^{lb} by regenerative randomization with regenerative state r . To obtain the upper bound for $m(t)$, the method modifies the CTMC X to obtain a CTMC X^{ub} . The CTMC X^{ub} is obtained from X by scaling the transition rates from states in S' so that, calling λ_i^{ub} , $i \in S'$ the output rates of X^{ub} , $\lambda_i^{\text{ub}} \geq \lambda_i$, $i \in S'$ and $\max_{i \in S'} \lambda_i^{\text{ub}} / \min_{i \in S'} \lambda_i^{\text{ub}} = D$. That scaling is defined by $\lambda_{i,j}^{\text{ub}} = \lambda_{i,j} (\lambda_i^{\text{ub}} / \lambda_i)$, $\lambda_i^{\text{ub}} = \max\{\lambda_i, \lambda_{\max}/D\}$, $i \in S'$. The upper bound for $m(t)$ is given by

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

That upper bound is obtained by solving X^{ub} by regenerative randomization with regenerative state r .

The particular case in which both bounds are to be computed, $D = 1$ and $\lambda_{\min} \geq \lambda_r$ allows a more efficient implementation of the bounding regenerative randomization method than that described in the previous paragraph. To justify that particular implementation, we will use the following result:

3. For class C'' models with the selection $r = o$, when $\lambda_{\max} = \lambda_{\min}$ and no selection for D is possible, regenerative randomization should be very efficient because of Theorem 1 and the fact that $R' = 1$, obviating the need for the bounding regenerative randomization method.

Lemma 1. For $x > 0$, $K \geq 0$, and $R > 1$,

$$\frac{1}{R} \sum_{k=K+1}^{\infty} (k-K)e^{-Rx} \frac{(Rx)^k}{k!} > \sum_{k=K+1}^{\infty} (k-K)e^{-x} \frac{x^k}{k!}.$$

Proof. See the Appendix. \square

For that particular case, denoting by superscripts lb and ub the terms referred to, respectively, X^{lb} and X^{ub} and the objects involved in their solution by regenerative randomization, and letting $R'' = \lambda_{\max}/\lambda_{\min} > 1$ (because $D = 1$ and $D < \lambda_{\max}/\lambda_{\min}$), we have $\lambda_{r,j}^{\text{lb}} = \lambda_{r,j}^{\text{ub}}$ and, noting that $\lambda_i^{\text{lb}} = \lambda_{\min}$ and $\lambda_i^{\text{ub}} = \lambda_{\max}$, $i \in S'$, we have $\lambda_{i,j}^{\text{lb}} = \lambda_{i,j}^{\text{ub}}/R''$, $i \in S'$. We also have $\Lambda^{\text{lb}} = \Lambda^{\text{ub}}/R''$ (because $\lambda_r^{\text{lb}} = \lambda_r^{\text{ub}} = \lambda_r \leq \lambda_{\min}$, $\lambda_i^{\text{lb}} = \lambda_{\min}$, $i \in S'$, $\lambda_i^{\text{ub}} = \lambda_{\max}$, $i \in S'$, and $\lambda_{\max} > \lambda_{\min}$). It follows that the transition probabilities of the randomized DTMCs \hat{X}^{lb} and \hat{X}^{ub} are related as $P_{r,j}^{\text{lb}} = R''P_{r,j}^{\text{ub}}$, $j \neq r$, $P_{r,r}^{\text{lb}} = 1 - R''(1 - P_{r,r}^{\text{ub}})$ and, for $i \in S'$, $P_{i,j}^{\text{lb}} = P_{i,j}^{\text{ub}}$. Then, taking into account that the transition probabilities of the DTMCs Z^{lb} and Z^{ub} from r to $j \in S'$ are, respectively, $P_{r,j}^{\text{lb}}$ and $P_{r,j}^{\text{ub}}$, that the transition probabilities of Z^{lb} and Z^{ub} within S' are, respectively, $P_{i,j}^{\text{lb}}$ and $P_{i,j}^{\text{ub}}$ and that the initial state of Z^{lb} and Z^{ub} is r , we have, for $k \geq 1$, $\pi_i^{\text{lb}}(k) = R''\pi_i^{\text{ub}}(k)$ and $a^{\text{lb}}(k) = R''a^{\text{ub}}(k)$. Then, using $a^{\text{lb}}(0) = a^{\text{ub}}(0) = 1$, we have

$$\begin{aligned} v_0^{\text{jlb}} &= P_{r,f_j}^{\text{lb}} = R''P_{r,f_j}^{\text{ub}} = R''v_0^{\text{jub}}, \\ q_0^{\text{lb}} &= P_{r,r}^{\text{lb}} = 1 - R''(1 - P_{r,r}^{\text{ub}}) = 1 - R''(1 - q_0^{\text{ub}}), \\ w_0^{\text{lb}} &= P_{r,S'}^{\text{lb}} = R''P_{r,S'}^{\text{ub}} = R''w_0^{\text{ub}}, \end{aligned}$$

and, taking into account that $\pi_r^{\text{lb}}(k) = \pi_r^{\text{ub}}(k) = 0$, $k \geq 1$, for $k \geq 1$, we have

$$\begin{aligned} v_k^{\text{jlb}} &= \frac{\sum_{i \in S'} \pi_i^{\text{lb}}(k) P_{i,f_j}^{\text{lb}}}{a^{\text{lb}}(k)} = \frac{\sum_{i \in S'} R'' \pi_i^{\text{ub}}(k) P_{i,f_j}^{\text{ub}}}{R'' a^{\text{ub}}(k)} \\ &= \frac{\sum_{i \in S'} \pi_i^{\text{ub}}(k) P_{i,f_j}^{\text{ub}}}{a^{\text{ub}}(k)} = v_k^{\text{jub}} \end{aligned}$$

and, similarly, $q_k^{\text{lb}} = q_k^{\text{ub}}$ and $w_k^{\text{lb}} = w_k^{\text{ub}}$. On the other hand, the DTMCs Z^{lb} and Z^{ub} have identical initial probability distributions and transition probabilities and, then, $a^{\text{lb}}(k) = a^{\text{ub}}(k)$, and, using $P_{i,j}^{\text{lb}} = P_{i,j}^{\text{ub}}$, $i \in S'$, $v_k^{\text{jlb}} = v_k^{\text{jub}}$, $q_k^{\text{lb}} = q_k^{\text{ub}}$, $w_k^{\text{lb}} = w_k^{\text{ub}}$. For $m \geq 1$,

$$\begin{aligned} r_{\max} \alpha_S a^{\text{ub}}(m) &= \sum_{k=m+1}^{\infty} (k-m) e^{-\Lambda^{\text{ub}} t_{\max}} (\Lambda^{\text{ub}} t_{\max})^k / k! \\ &= r_{\max} \alpha_S (a^{\text{lb}}(m)/R'') \sum_{k=m+1}^{\infty} (k-m) e^{-R'' \Lambda^{\text{ub}} t_{\max}} (R'' \Lambda^{\text{ub}} t_{\max})^k / k! \\ &> r_{\max} \alpha_S a^{\text{lb}}(m) \sum_{k=m+1}^{\infty} (k-m) e^{-\Lambda^{\text{lb}} t_{\max}} (\Lambda^{\text{lb}} t_{\max})^k / k!, \end{aligned}$$

by Lemma 1 with $K = m$, $x = \Lambda^{\text{lb}} t_{\max}$ and $R = R''$, implying $K^{\text{ub}} \geq K^{\text{lb}}$. If $\alpha_{S'} > 0$, for $m \geq 1$, $\Lambda^{\text{ub}} > \Lambda^{\text{lb}}$ implies⁴

$$\begin{aligned} r_{\max} a^{\text{ub}}(m) &= \sum_{k=m+1}^{\infty} e^{-\Lambda^{\text{ub}} t_{\max}} (\Lambda^{\text{ub}} t_{\max})^k / k! \\ &= r_{\max} a^{\text{lb}}(m) \sum_{k=m+1}^{\infty} e^{-\Lambda^{\text{ub}} t_{\max}} (\Lambda^{\text{ub}} t_{\max})^k / k! \\ &> r_{\max} a^{\text{lb}}(m) \sum_{k=m+1}^{\infty} e^{-\Lambda^{\text{lb}} t_{\max}} (\Lambda^{\text{lb}} t_{\max})^k / k!, \end{aligned}$$

implying $L^{\text{ub}} \geq L^{\text{lb}}$. Then, if we start by computing the upper bounds $m^{\text{ub}}(t)$ using X^{ub} and save $a^{\text{ub}}(k)$, v_k^{jub} , q_k^{ub} , w_k^{ub} , and, if $\alpha_{S'} > 0$, $a^{\text{ub}}(k)$, v_k^{jub} , q_k^{ub} , w_k^{ub} , we can use the relationships between those parameters and the corresponding parameters for X^{lb} to avoid stepping Z^{lb} and Z^{ub} when computing the lower bounds $m^{\text{lb}}(t)$.

An algorithmic description of the bounding regenerative randomization method, including the previously discussed particular implementation, is given in Fig. 6. The algorithm has as inputs the CTMC X , the number A of absorbing states f_i , the reward rates $r_{f_1}, r_{f_2}, \dots, r_{f_A}$, an initial probability distribution vector $\alpha = (\alpha_i)_{i \in \Omega}$ with $\alpha_S > 0$, parameters lb and ub indicating, respectively, whether the lower and upper bounds for $m(t)$ are desired or not, the regenerative state r , the controlling parameter D , the allowed error ε , the number of time points n at which $m^{\text{lb}}(t)$, $m^{\text{ub}}(t)$ have to be computed, and the time points t_1, t_2, \dots, t_n . The algorithm has as outputs the computed values of $m^{\text{lb}}(t)$, $\tilde{m}^{\text{lb}}(t_1), \tilde{m}^{\text{lb}}(t_2), \dots, \tilde{m}^{\text{lb}}(t_n)$ and of $m^{\text{ub}}(t)$, $\tilde{m}^{\text{ub}}(t_1), \tilde{m}^{\text{ub}}(t_2), \dots, \tilde{m}^{\text{ub}}(t_n)$. The algorithmic description makes reference to DTMCs $\hat{V}_{K,L}$ ($\alpha_{S'} > 0$) and \hat{V}_K ($\alpha_{S'} = 0$). Those DTMCs are the randomized DTMCs with randomization rate Λ^{lb} of the truncated transformed models of X^{lb} used in the solution of X^{lb} by regenerative randomization. For the case $\alpha_{S'} > 0$, $\hat{V}_{K,L}$ has state space

$$\{s_k, 0 \leq k \leq K\} \cup \{s'_k, 0 \leq k \leq L\} \cup \{f_1, f_2, \dots, f_A, a\},$$

initial probability distribution

$$P[(\hat{V}_{K,L})_0 = s_0] = \alpha_r,$$

$$P[(\hat{V}_{K,L})_0 = s'_0] = \alpha_{S'},$$

$$P[(\hat{V}_{K,L})_0 = f_i] = \alpha_{f_i},$$

$$P[(\hat{V}_{K,L})_0 = i] = 0, \quad i \notin \{s_0, s'_0, f_1, f_2, \dots, f_A\}$$

and the state transition diagram illustrated in Fig. 5 for the case $A = 1$, where $K, L, v_k^{\text{lb}}, q_k^{\text{lb}}, w_k^{\text{lb}}, v_k^{\text{ub}}, q_k^{\text{ub}}$ and w_k^{ub} have the values computed in the algorithm. For the case $\alpha_{S'} = 0$, \hat{V}_K is the DTMC with initial probability distribution

$$P[(\hat{V}_K)_0 = s_0] = \alpha_S,$$

$$P[(\hat{V}_K)_0 = f_i] = \alpha_{f_i},$$

$$P[(\hat{V}_K)_0 = i] = 0, \quad i \notin \{s_0, f_1, f_2, \dots, f_A\}$$

4. The inequality comes from the fact that $\sum_{k=m+1}^{\infty} e^{-\Lambda t} (\Lambda t)^k / k!$ is the probability that the number of arrivals in the interval $[0, t]$ in a Poisson process with arrival rate Λ is $\geq m+1$, which is increasing with t , and, therefore, with Λ .


```

Inputs:  $X, A, r_{f_1}, r_{f_2}, \dots, r_{f_A}, \alpha, lb, ub, r, D, \varepsilon, n, t_1, t_2, \dots, t_n$ 
Outputs:  $\tilde{m}^{lb}(t_1), \tilde{m}^{lb}(t_2), \dots, \tilde{m}^{lb}(t_n), \tilde{m}^{ub}(t_1), \tilde{m}^{ub}(t_2), \dots, \tilde{m}^{ub}(t_n)$ 
 $\lambda_{\min} = \min_{i \in S'} \lambda_i; \lambda_{\max} = \max_{i \in S'} \lambda_i;$ 
if ( $!lb \ \&\& \ !ub \ || \ D < 1 \ || \ D \geq \lambda_{\max}/\lambda_{\min}$ ) exit;
if ( $lb \ \&\& \ ub \ \&\& \ D == 1 \ \&\& \ \lambda_{\min} \geq \lambda_r$ ) {
  Obtain  $X^{ub}$  from  $X$ ;
  Compute  $\tilde{m}^{ub}(t_1), \tilde{m}^{ub}(t_2), \dots, \tilde{m}^{ub}(t_n)$  by solving  $X^{ub}$  by regenerative
  randomization with regenerative state  $r$  and allowed error  $\varepsilon$ ;
  Let  $K^{ub}, a^{ub}(k), v_k^{jub}, q_k^{ub}, w_k^{ub}$  be the values of  $K, a(k), v_k^j, q_k, w_k$  obtained when
  solving  $X^{ub}$  by regenerative randomization and, if  $\alpha_{S'} > 0$ , let  $L^{ub}, a'^{ub}(k), v_k^{jub},$ 
 $q_k^{ub}, w_k^{ub}$  be the values of  $L, a'(k), v_k^j, q_k, w_k$  obtained when solving  $X^{ub}$  by regenerative
  randomization;
   $r_{\max} = \max\{r_{f_1}, r_{f_2}, \dots, r_{f_A}\}; t_{\max} = \max\{t_1, t_2, \dots, t_n\};$ 
 $\Lambda = (1 + 10^{-4})\lambda_{\min}; R'' = \lambda_{\max}/\lambda_{\min}; \alpha_{S'} = \sum_{i \in S'} \alpha_i; \alpha_S = \alpha_r + \alpha_{S'};$ 
  if ( $\alpha_{S'} > 0$ )  $tol_K = \varepsilon/4;$  else  $tol_K = \varepsilon/2;$ 
  for ( $k = 1; k \leq K^{ub}; k++$ )  $a(k) = R'' a^{ub}(k);$ 
  for ( $j = 1; j \leq A; j++$ )  $v_0^j = R'' v_0^{jub};$ 
 $q_0 = 1 - R''(1 - q_0^{ub}); w_0 = R'' w_0^{ub};$ 
 $K = \min\{m \geq 1 : r_{\max} \alpha_S a(m) \sum_{k=m+1}^{\infty} (k-m) e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k! \leq tol_K\};$ 
  for ( $k = 1; k \leq K - 1; k++$ ) {
    for ( $j = 1; j \leq A; j++$ )  $v_k^j = v_k^{jub}; q_k = q_k^{ub}; w_k = w_k^{ub};$ 
  }
  if ( $\alpha_{S'} > 0$ ) {
    for ( $k = 1; k \leq L^{ub}; k++$ )  $a'(k) = a'^{ub}(k);$ 
 $L = \min\{m \geq 1 : r_{\max} a'(m) \sum_{k=m+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k! \leq \varepsilon/4\};$ 
    for ( $k = 0; k \leq L - 1; k++$ ) {
      for ( $j = 1; j \leq A; j++$ )  $v_k^j = v_k^{jub}; q_k^j = q_k^{ub}; w_k^j = w_k^{ub};$ 
    }
  }
   $N = \min\{m \geq 0 : r_{\max} \sum_{k=m+1}^{\infty} e^{-\Lambda t_{\max}} (\Lambda t_{\max})^k / k! \leq \varepsilon/2\};$ 
  if ( $\alpha_{S'} > 0$ )
    Give  $N$  steps to  $\hat{V}_{K,L}$  and compute  $d(k) = \sum_{i=1}^A r_{f_i} P[(\hat{V}_{K,L})_k = f_i],$ 
 $k = 0, 1, \dots, N;$ 
  else
    Give  $N$  steps to  $\hat{V}_K$  and compute  $d(k) = \sum_{i=1}^A r_{f_i} P[(\hat{V}_K)_k = f_i],$ 
 $k = 0, 1, \dots, N;$ 
  for ( $i = 1; i \leq n; i++$ )
    for ( $k = 0, \tilde{m}^{lb}(t_i) = 0; k \leq N; k++$ )  $\tilde{m}^{lb}(t_i) += d(k) e^{-\Lambda t_i} (\Lambda t_i)^k / k!;$ 
}
else {
  if ( $lb$ ) {
    Obtain  $X^{lb}$  from  $X$ ;
    Compute  $\tilde{m}^{lb}(t_1), \tilde{m}^{lb}(t_2), \dots, \tilde{m}^{lb}(t_n)$  by solving  $X^{lb}$  by regenerative
    randomization with regenerative state  $r$  and allowed error  $\varepsilon$ ;
  }
  if ( $ub$ ) {
    Obtain  $X^{ub}$  from  $X$ ;
    Compute  $\tilde{m}^{ub}(t_1), \tilde{m}^{ub}(t_2), \dots, \tilde{m}^{ub}(t_n)$  by solving  $X^{ub}$  by regenerative
    randomization with regenerative state  $r$  and allowed error  $\varepsilon$ ;
  }
}
}

```

Fig. 6. Algorithmic description of bounding regenerative randomization.

and state transition diagram identical to the state transition diagram of $\hat{V}_{K,L}$, but without states s'_k .

In the following, we prove the correctness of the method, i.e., $m^{lb}(t) \leq m(t) \leq m^{ub}(t)$. To that end, we consider the embedded DTMC of X , $\Pi = \{\Pi_k; k = 0, 1, 2, \dots\}$: Π has same state space and initial probability distribution as X and transition probabilities $\psi_{i,j} = \lambda_{i,j}/\lambda_i, i \in S, j \in \Omega - \{i\},$

$\psi_{i,i} = 0, i \in S, \psi_{f_i, f_i} = 1, 1 \leq i \leq A, \psi_{f_i, j} = 0, j \neq f_i$. The behavior of X can be described in terms of the DTMC Π by saying that the sequence of states visited by X is given by Π with sojourn times in each state i of X exponentially distributed with parameter λ_i , independently on the path followed by Π . That interpretation is usually referred to as the "structure" of X [7, Section 8.3]. Since X^{lb} and X^{ub} have

been obtained from X by scaling transition rates from S' , keeping unmodified the relative values of the transition rates from a given state, X^{lb} and X^{ub} have embedded DTMCs with same initial probability distributions and transition probabilities as Π . The proof is made in a sequence of a lemma, two propositions, and a theorem. U being a continuous random variable with differentiable distribution function, $f_U(u)$ and $F_U(u)$ will denote, respectively, the probability density function and distribution function of U .

Lemma 2. *Let U_1, U_2, U'_1 , and U'_2 be independent continuous random variables with differentiable distribution functions. Assume $F_{U_1}(u) \leq F_{U'_1}(u)$ and $F_{U_2}(u) \leq F_{U'_2}(u)$. Then, $F_{U_1+U_2}(u) \leq F_{U'_1+U'_2}(u)$.*

Proof. Using the hypotheses:

$$\begin{aligned} F_{U_1+U_2}(u) &= \int_{-\infty}^u f_{U_1}(x)F_{U_2}(u-x) dx \\ &\leq \int_{-\infty}^u f_{U_1}(x)F_{U'_2}(u-x) dx \\ &= F_{U_1+U'_2}(u) = F_{U'_2+U_1}(u) \\ &= \int_{-\infty}^u f_{U'_2}(x)F_{U_1}(u-x) dx \\ &\leq \int_{-\infty}^u f_{U'_2}(x)F_{U'_1}(u-x) dx \\ &= F_{U'_2+U'_1}(u) = F_{U'_1+U'_2}(u). \end{aligned}$$

□

Proposition 1. *Let U_1, U_2, \dots, U_n and U'_1, U'_2, \dots, U'_n be independent continuous random variables with differentiable distribution functions. Assume $F_{U_i}(u) \leq F_{U'_i}(u)$, $1 \leq i \leq n$. Then, $F_{\sum_{i=1}^n U_i}(u) \leq F_{\sum_{i=1}^n U'_i}(u)$.*

Proof. By induction on n : The base case $n = 1$ is trivial. Assume that the result holds for $n - 1$ and let us prove the result for n . The random variables $U_1, \sum_{i=2}^n U_i, U'_1$ and $\sum_{i=2}^n U'_i$ are independent. By the induction hypothesis, $F_{\sum_{i=2}^n U_i}(u) \leq F_{\sum_{i=2}^n U'_i}(u)$. Then, using Lemma 2, we have

$$F_{\sum_{i=1}^n U_i}(u) = F_{U_1+\sum_{i=2}^n U_i}(u) \leq F_{U'_1+\sum_{i=2}^n U'_i}(u) = F_{\sum_{i=1}^n U'_i}(u).$$

□

Proposition 2.

$$P[X^{\text{lb}}(t) = f_i] \leq P[X(t) = f_i] \leq P[X^{\text{ub}}(t) = f_i].$$

Proof. Let \mathcal{P}_i be the set of paths of Π to absorption in f_i . Each path $p \in \mathcal{P}_i$ is a sequence $(x_1(p), x_2(p), \dots, x_{L(p)}(p))$ with $P[\Pi_0 = x_1(p)] > 0$, $L(p) > 1$, $x_i(p) \in S$, $1 \leq i \leq L(p) - 1$ and $x_{L(p)}(p) = f_i$, and has a probability

$$P(p) = \alpha_{x_1(p)} \prod_{i=1}^{L(p)-1} \psi_{x_i(p), x_{i+1}(p)}.$$

Let $\text{EXP}(\lambda)$ denote an exponential random variable with parameter λ . Since $\lambda_i^{\text{lb}} \leq \lambda_i \leq \lambda_i^{\text{ub}}$, $i \in S$, and $F_{\text{EXP}(\lambda)}(u) = 1 - e^{-\lambda u}$, increasing with λ , we have

$$F_{\text{EXP}(\lambda_i^{\text{ub}})}(u) \leq F_{\text{EXP}(\lambda_i)}(u) \leq F_{\text{EXP}(\lambda_i^{\text{lb}})}(u), \quad i \in S.$$

In addition, we have:

$$\begin{aligned} P[X(t) = f_i] &= P[X(0) = f_i] + \sum_{p \in \mathcal{P}_i} P(p) F_{\sum_{i=1}^{L(p)-1} \text{EXP}(\lambda_{x_i(p)})}(t), \\ P[X^{\text{lb}}(t) = f_i] &= P[X(0) = f_i] + \sum_{p \in \mathcal{P}_i} P(p) F_{\sum_{i=1}^{L(p)-1} \text{EXP}(\lambda_{x_i(p)}^{\text{lb}})}(t), \\ P[X^{\text{ub}}(t) = f_i] &= P[X(0) = f_i] \\ &\quad + \sum_{p \in \mathcal{P}_i} P(p) F_{\sum_{i=1}^{L(p)-1} \text{EXP}(\lambda_{x_i(p)}^{\text{ub}})}(t), \end{aligned}$$

where all exponential variables are independent among them. Using Proposition 1,

$$F_{\sum_{i=1}^{L(p)-1} \text{EXP}(\lambda_{x_i(p)}^{\text{ub}})} \leq F_{\sum_{i=1}^{L(p)-1} \text{EXP}(\lambda_{x_i(p)})} \leq F_{\sum_{i=1}^{L(p)-1} \text{EXP}(\lambda_{x_i(p)}^{\text{lb}})}$$

and the result follows. □

Theorem 2. $m^{\text{lb}}(t) \leq m(t) \leq m^{\text{ub}}(t)$.

Proof. The result follows immediately from Proposition 2, the definitions of $m(t)$, $m^{\text{lb}}(t)$, and $m^{\text{ub}}(t)$, and $r_{f_i} \geq 0$. □

Essentially, $m^{\text{lb}}(t)$ lower bounds $m(t)$ because the scaling which yields X^{lb} from X has been done by decreasing output rates of the model. Similarly, $m^{\text{ub}}(t)$ essentially upper bounds $m(t)$ because the scaling which yields X^{ub} from X has been done by increasing output rates of the model. As D decreases, λ_i^{lb} decreases for some $i \in S'$ and, as D decreases, λ_i^{ub} increases for some $i \in S'$. Then, it follows that $m^{\text{lb}}(t)$ and $m^{\text{ub}}(t)$ get looser as D decreases. For class C'' models, the CTMCs X^{lb} and X^{ub} obtained with $r = o$ also belong to the class, making $r = o$ the natural selection for the bounding regenerative randomization method. Since $\max_{i \in S'} \lambda_i^{\text{lb}} / \min_{i \in S'} \lambda_i^{\text{lb}} = D$ and $\max_{i \in S'} \lambda_i^{\text{ub}} / \min_{i \in S'} \lambda_i^{\text{ub}} = D$, for class C'' models with $r = o$, the R' parameter of both X^{lb} and X^{ub} is equal to D . Then, it follows that the cost of bounding regenerative randomization for class C'' models with the selection $r = o$ should decrease as D decreases. In particular, for $D = 1$, the required K and L to solve X^{lb} and X^{ub} by regenerative randomization should be very small and the bounding regenerative method should be very efficient. If, in addition, both bounds are requested, property P3 implies $\lambda_r < \lambda_{\min}$ and the method will only step Z^{ub} and, if $\alpha_{S-\{o\}} > 0$, Z^{ub} .

4 ANALYSIS AND COMPARISON

In this section, we analyze the performance of bounding regenerative randomization in terms of bounds tightness and computational efficiency and compare the latter with that of standard randomization and regenerative randomization using a large example belonging to class C'' .

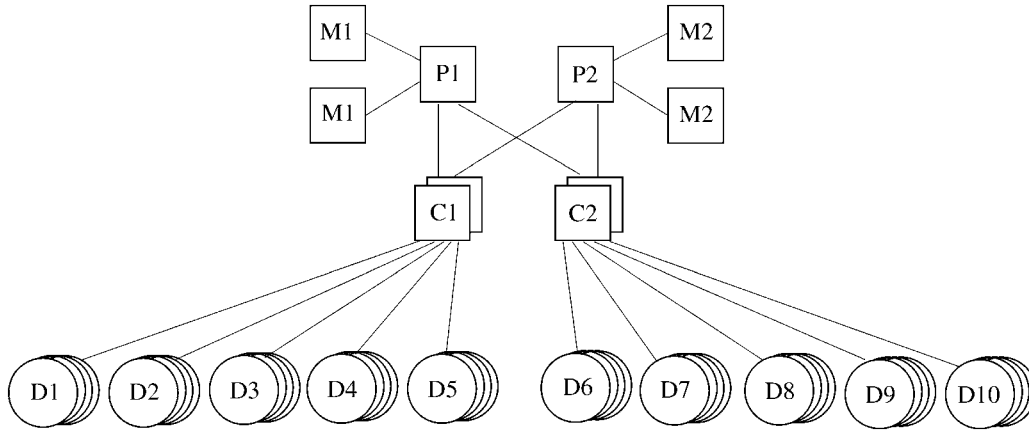


Fig. 7. Block diagram of the example.

The example is an exact reliability model of the fault-tolerant system whose block diagram is given in Fig. 7. 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 five sets of disks. The system is operational if at least one processor and one memory connected to it are operational, one controller of each set is operational, and three disks of each set are operational. Processors fail with rate 10^{-5} h^{-1} ; a processor failure is soft with probability 0.8 and hard with probability 0.2. Memories fail with rate $5 \times 10^{-5} \text{ h}^{-1}$. Controllers fail with rate 10^{-5} h^{-1} . Disks fail with rates $5 \times 10^{-6} \text{ h}^{-1}$. A failure of a controller is uncovered and leads to system failure with probability 0.01. There is an unlimited number of repairmen which perform restarts of processors in soft failure with rate 60 h^{-1} . The other repair actions are performed by another repairman with first priority given to disks, next to controllers, next to processors in hard failure, and last to memories. Components with the same repair priority are chosen at random. The repair rates are μ_{PH} for processors in hard failure, μ_M for memories, μ_C for controllers, and μ_D for disks. For the model parameters μ_{PH} , μ_M , μ_C , and μ_D , we will consider the sets of values A and B given in Table 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 initial probability distribution, we assume

TABLE 1
Model Parameter Value Sets

parameter	set A	set B
μ_{PH}	0.5	0.1
μ_M	0.5	0.1
μ_C	1	0.2
μ_D	0.2	0.04

that the initial state of X is the state o in which no component is failed, which is taken as regenerative state r for both regenerative randomization and bounding regenerative randomization. For set A, $R' \approx 300$, for set B, $R' \approx 1,500$, and, therefore, set B stresses more strongly regenerative randomization than set A. All methods are run with $\varepsilon = 10^{-12}$ and a single target time t . The bounding regenerative randomization method is run asking both the lower and the upper bound.

We first analyze bounding regenerative randomization with $D = 1$ and compare it with regenerative randomization and standard randomization. Tables 2 and 3 give, for sets A and B, respectively, the bounds obtained by bounding regenerative randomization (BRR), the number of steps which have to be given to the DTMC Z^{ub} in that method, the number of steps to the DTMC Z under regenerative randomization (RR), and the number of steps to \hat{X} under standard randomization (SR). Memory usage for all methods was about 70 MB. Figs. 8 and 9 give the corresponding CPU times, measured on a 167 MHz, 128 MB UltraSPARC1 workstation, except for standard randomization when t is large, for which the given CPU times were estimated from the number of steps required by the method. The bounds obtained by bounding regenerative randomization are quite tight and, as expected, are obtained with very few steps. Regarding the tightness, we want to note that, taking $(m^{\text{lb}}(t) + m^{\text{ub}}(t))/2$ as estimate for the unreliability at time t , a relative error bound $(m^{\text{ub}}(t) - m^{\text{lb}}(t))/(m^{\text{lb}}(t) + m^{\text{ub}}(t))$ ranging from 1.76 percent for the smallest t to 0.112 percent for the largest t is achieved for set A and a relative error bound ranging from 8.89 percent for the smallest t to 0.571 percent for the largest t is achieved for set B. The tightness of the bounds increases with t and is higher for set A than for set B. The latter is because, for set B, $\min_{i \in S - \{o\}} \lambda_i$ is significantly smaller (about five times smaller) than for set A, making the lower bound obtained from X^{lb} significantly less tight. Regenerative randomization requires a substantially larger number of steps which, as predicted, for large t , increases logarithmically with t . In accordance with theory, R' being larger for set B than for set A, the method requires more steps for set B than for set A. Also, the rule of thumb given in [5] that the K and L required by regenerative randomization is

TABLE 2

Bounds Given by Bounding Regenerative Randomization and Number of Steps Required by the Methods for Set A

t (h)	BRR			RR (steps)	SR (steps)
	lower bound	upper bound	steps		
1	4.0142×10^{-7}	4.1582×10^{-7}	8	107	123
2	8.0531×10^{-7}	8.3190×10^{-7}	8	184	207
5	2.0278×10^{-6}	2.0802×10^{-6}	8	399	436
10	4.0858×10^{-6}	4.1606×10^{-6}	8	737	792
20	8.2282×10^{-6}	8.3214×10^{-6}	10	1,385	1,474
50	2.0681×10^{-5}	2.0804×10^{-5}	10	3,238	3,447
100	4.1436×10^{-5}	4.1607×10^{-5}	10	6,059	6,657
200	8.2947×10^{-5}	8.3213×10^{-5}	10	7,443	12,985
500	2.0747×10^{-4}	2.0802×10^{-4}	10	7,927	31,737
1,000	4.1497×10^{-4}	4.1600×10^{-4}	10	8,192	62,746
2,000	8.2984×10^{-4}	8.3182×10^{-4}	10	8,430	124,466
5,000	2.0734×10^{-3}	2.0783×10^{-3}	12	8,729	308,895
10,000	4.1426×10^{-3}	4.1522×10^{-3}	12	8,950	615,507
20,000	8.2682×10^{-3}	8.2871×10^{-3}	12	9,168	1,227,787
50,000	2.0543×10^{-2}	2.0589×10^{-2}	12	9,456	3,062,315

TABLE 3

Bounds Given by Bounding Regenerative Randomization and Number of Steps Required by the Methods for Set B

t (h)	BRR			RR (steps)	SR (steps)
	lower bound	upper bound	steps		
1	4.0148×10^{-7}	4.7979×10^{-7}	10	106	122
2	8.0586×10^{-7}	9.6096×10^{-7}	12	184	205
5	2.0352×10^{-6}	2.4045×10^{-6}	12	400	431
10	4.1323×10^{-6}	4.8103×10^{-6}	12	739	782
20	8.4691×10^{-6}	9.6220×10^{-6}	12	1,394	1,456
50	2.2138×10^{-5}	2.4057×10^{-5}	12	3,298	3,404
100	4.5691×10^{-5}	4.8115×10^{-5}	14	6,400	6,574
200	9.3218×10^{-5}	9.6228×10^{-5}	14	12,494	12,820
500	2.3589×10^{-4}	2.4056×10^{-4}	14	30,151	31,329
1,000	4.7364×10^{-4}	4.8105×10^{-4}	14	40,557	61,934
2,000	9.4897×10^{-4}	9.6188×10^{-4}	14	42,808	122,850
5,000	2.3736×10^{-3}	2.4030×10^{-3}	16	44,769	304,870
10,000	4.7435×10^{-3}	4.8002×10^{-3}	16	46,043	607,470
20,000	9.4664×10^{-3}	9.5773×10^{-3}	16	47,249	1,211,736
50,000	2.3501×10^{-2}	2.3771×10^{-2}	16	48,797	3,022,234

approximately upper bounded by $30 \max_{i \in S} \lambda_i / \min_{i \in S'} \lambda_i = 30R'$ seems to work well.

As a result of the high stiffness of the model, standard randomization is extremely expensive for large t . Since the performance of standard randomization is determined by $\max_{i \in S} \lambda_i t$ and $\max_{i \in S} \lambda_i$ is very similar for both model parameter value sets, the number of steps required by standard randomization is very similar for both model parameter value sets. Regarding CPU times, bounding regenerative randomization is very inexpensive and much faster than both regenerative randomization and standard randomization. For set A, the CPU time under bounding regenerative randomization goes from 7.961 s to 135.0 s, whereas the CPU time under regenerative randomization goes from 76.86 s to 17,028 s (about 5 h) and the CPU time under standard randomization goes from 78.11 s to

1.963×10^6 s (about 23 days), making bounding regenerative randomization 126 times faster than regenerative randomization and 14,541 times faster than standard randomization for the largest t . For set B, the CPU time under bounding regenerative randomization goes from 9.325 s to 138.7 s, whereas the CPU time under regenerative randomization goes from 77.16 s to 109,019 s (about 30 h) and the CPU time under standard randomization goes from 77.14 s to 1.882×10^6 s (about 22 days), making bounding regenerative randomization 786 times faster than regenerative randomization and 13,569 times faster than standard randomization for the largest t . For large t , standard randomization is extremely expensive and regenerative randomization is very costly, whereas the bounding regenerative randomization is very fast. In spite of involving fewer steps, regenerative randomization has cost similar to or slightly larger than standard randomiza-

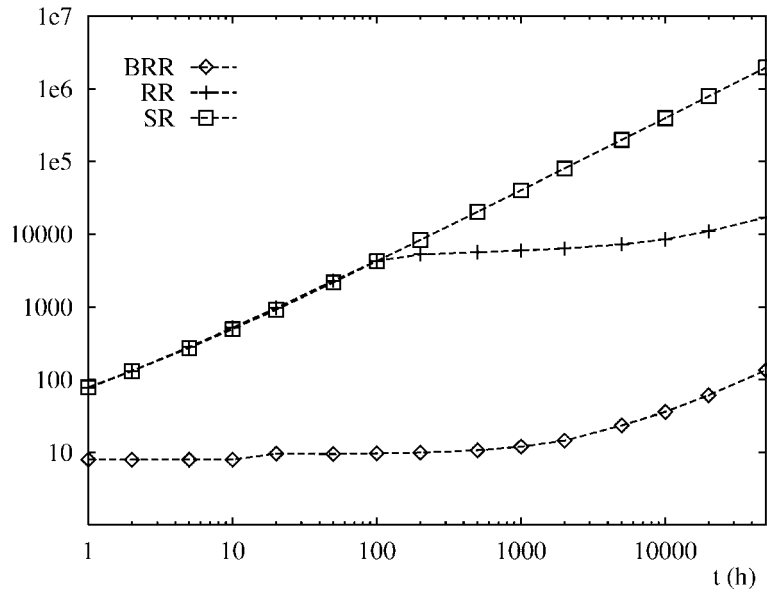


Fig. 8. CPU times in seconds consumed by the methods for set A.

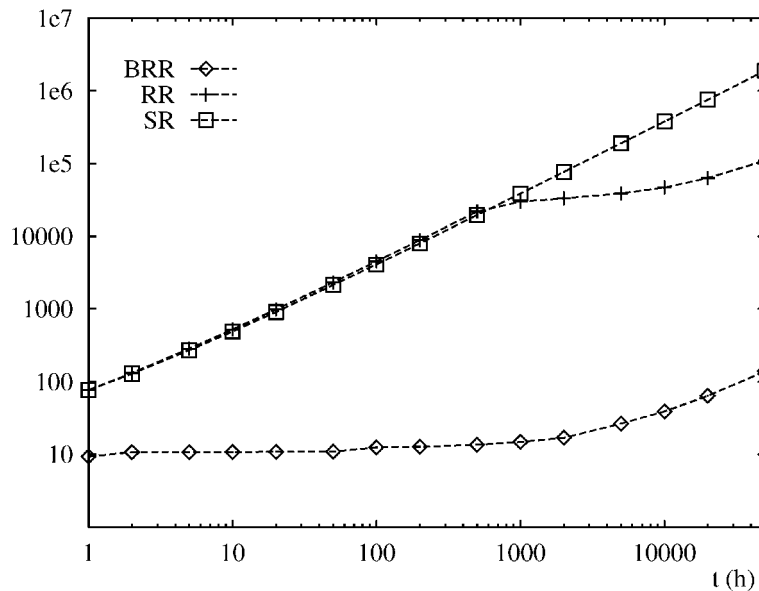


Fig. 9. CPU times in seconds consumed by the methods for set B.

tion for small and medium t . This is because the steps in regenerative randomization tend to be slightly more costly than the steps in standard randomization. For large t , the benign behavior of regenerative randomization enters into play and regenerative randomization is significantly less costly than standard randomization. The fast increase rate of the CPU times of both bounding regenerative randomization and regenerative randomization for large t is due to the fact that the cost of solving the truncated transformed models by standard randomization is linear in both the size of the truncated transformed model (which increases logarithmically with t) and t and that cost is important for large t .

Finally, we explore the trade-off in the bounding regenerative randomization method between bounds tight-

ness and computational cost, which is controlled by the parameter D . Table 4 gives the relative error bound $(m^{ub}(t) - m^{lb}(t))/(m^{lb}(t) + m^{ub}(t))$ and the required number of steps to Z^{ub} and CPU time as a function of D for $t = 10,000$ h and both model parameter value sets. As D increases, the bounds become tighter, but the computational effort increases considerably and can be even larger than the cost of regenerative randomization (8,518 s for set A and 46,581 s for set B).

5 CONCLUSIONS

Taking as the starting point the regenerative randomization method, we have developed a method called bounding

TABLE 4
Performance of Bounding Regenerative Randomization as a Function of D for Sets A and B and $t = 10,000$ h

D	set A			set B		
	Rel. error	steps	CPU time (s)	Rel. error	steps	CPU time (s)
1	1.152×10^{-3}	12	36.01	5.939×10^{-3}	16	39.01
2	8.247×10^{-4}	90	101.4	4.311×10^{-3}	99	107.6
5	7.213×10^{-4}	272	244.8	3.840×10^{-3}	290	256.1
10	6.989×10^{-4}	572	482.3	3.801×10^{-3}	607	510.1
20	6.602×10^{-4}	1,170	964.8	3.752×10^{-3}	1,240	1,023
50	5.490×10^{-4}	2,966	2,497	3.630×10^{-3}	3,138	2,647
100	3.779×10^{-4}	5,948	5,032	3.435×10^{-3}	6,302	5,124
200	1.741×10^{-4}	11,755	10,783	3.047×10^{-3}	12,629	10,481
500				1.957×10^{-3}	31,518	28,812

The symbol — indicates that the D value is not applicable because $D \geq \lambda_{\max}/\lambda_{\min}$.

regenerative randomization to compute bounds for a measure defined on CTMC models with a structure encompassing both exact and bounding reliability CTMC models of fault-tolerant systems. The method is numerically stable and computes the bounds with well-controlled and specifiable-in-advance error. Computational effort can be traded off with bounds accuracy. For a class of CTMC models, class C'' , including typical failure/repair models with exponential failure and repair time distributions and repair in every state with failed components, the method can yield reasonably tight bounds at a very small computational cost, allowing an approximate analysis with error bounds of very large CTMC models of that class in affordable CPU times.

APPENDIX

Let the function

$$f_y(z) = \frac{1}{z} \sum_{k=K+1}^{\infty} (k-K)e^{-zy} \frac{(zy)^k}{k!}.$$

We have to show $f_x(R) > f_x(1)$ for $x > 0$, $K \geq 0$, $R > 1$.

Taking derivatives:

$$\begin{aligned} f'_y(z) &= -\frac{1}{z^2} \sum_{k=K+1}^{\infty} (k-K)e^{-zy} \frac{(zy)^k}{k!} \\ &\quad - \frac{y}{z} \sum_{k=K+1}^{\infty} (k-K)e^{-zy} \frac{(zy)^k}{k!} \\ &\quad + \frac{y}{z} \sum_{k=K+1}^{\infty} (k-K)e^{-zy} \frac{(zy)^{k-1}}{(k-1)!}. \end{aligned}$$

But,

$$\begin{aligned} &\sum_{k=K+1}^{\infty} (k-K)e^{-zy} \frac{(zy)^{k-1}}{(k-1)!} \\ &= \sum_{k=K}^{\infty} (k-K+1)e^{-zy} \frac{(zy)^k}{k!} \\ &= \sum_{k=K}^{\infty} e^{-zy} \frac{(zy)^k}{k!} + \sum_{k=K+1}^{\infty} (k-K)e^{-zy} \frac{(zy)^k}{k!}, \end{aligned}$$

yielding

$$f'_y(z) = \frac{y}{z} \sum_{k=K}^{\infty} e^{-zy} \frac{(zy)^k}{k!} - \frac{1}{z^2} \sum_{k=K+1}^{\infty} (k-K)e^{-zy} \frac{(zy)^k}{k!},$$

which gives

$$f'_y(1) = y \sum_{k=K}^{\infty} e^{-y} \frac{y^k}{k!} - \sum_{k=K+1}^{\infty} (k-K)e^{-y} \frac{y^k}{k!}.$$

And,

$$\begin{aligned} y \sum_{k=K}^{\infty} e^{-y} \frac{y^k}{k!} &= y \sum_{k=1}^{\infty} e^{-y} \frac{y^{K+k-1}}{(K+k-1)!} = \sum_{k=1}^{\infty} e^{-y} \frac{y^{K+k}}{(K+k-1)!}, \\ \sum_{k=K+1}^{\infty} (k-K)e^{-y} \frac{y^k}{k!} &= \sum_{k=1}^{\infty} ke^{-y} \frac{y^{K+k}}{(K+k)!}, \end{aligned}$$

resulting in

$$\begin{aligned} f'_y(1) &= \sum_{k=1}^{\infty} e^{-y} \frac{y^{K+k}}{(K+k-1)!} - \sum_{k=1}^{\infty} ke^{-y} \frac{y^{K+k}}{(K+k)!} \\ &= \sum_{k=1}^{\infty} (K+k)e^{-y} \frac{y^{K+k}}{(K+k)!} - \sum_{k=1}^{\infty} ke^{-y} \frac{y^{K+k}}{(K+k)!} \\ &= K \sum_{k=1}^{\infty} e^{-y} \frac{y^{K+k}}{(K+k)!} = K \sum_{k=K+1}^{\infty} e^{-y} \frac{y^k}{k!}. \end{aligned}$$

For $y > 0$, $\sum_{k=K+1}^{\infty} e^{-y} y^k/k!$ is > 0 and increasing with y because it is the probability that the number of arrivals in the time interval $[0, y]$ of a Poisson process with arrival rate 1 is $\geq K+1$. This shows that, for $y > 0$, $f'_y(1)$ is > 0 and increasing with y . Let $m = f'_x(1) > 0$ since $x > 0$. We have $f'_y(1) \geq m$ for $y \geq x$.

On the other hand, $f''_y(z)$ is bounded for any y and z , $y > 0$, $z > 0$. Let $M > \sup_{x \leq y \leq xR, 1 \leq z \leq R} |f''_y(z)| \geq 0$, $M < \infty$. Let $q = 2m/M > 0$. Taylor's theorem applied to $f_y(z)$ at $z = 1$ gives

$$f_y(1+r) = f_y(1) + f'_y(1)r + \frac{f''_y(z)}{2}r^2, \quad r \geq 0, z \in [1, 1+r].$$

Then, for $x \leq y \leq xR$, $1 < 1+r \leq R$, and $r \leq q$, we have

$$\begin{aligned} f_y(1+r) - f_y(1) &= f'_y(1)r + \frac{f''_y(z)}{2}r^2 > mr - \frac{M}{2}r^2 \\ &= r\left(m - \frac{M}{2}r\right) \geq r\left(m - \frac{M}{2}q\right) = 0, \end{aligned}$$

implying $f_y(1+r) > f_y(1)$ for $x \leq y \leq xR$, $1 < 1+r \leq R$ and $r \leq q$.

The result asserted by the lemma can be proven from the fact that $f_y(1+r) > f_y(1)$ for $x \leq y \leq xR$, $1 < 1+r \leq R$ and $r \leq q$ as follows: Let N be the minimum integer $n > 0$ with $(1+q)^n \geq R$. We can write $R = (1+q)^{N-1}(1+r)$ with $r \leq q$, $1 < 1+r \leq R$ and, if $N > 1$, $1+q < R$. Then,

$$\begin{aligned} f_x(R) &= f_x((1+q)^{N-1}(1+r)) \\ &= \frac{1}{(1+q)^{N-1}} f_{x(1+q)^{N-1}}(1+r) > \frac{1}{(1+q)^{N-1}} f_{x(1+q)^{N-1}}(1) \\ &= \frac{1}{(1+q)^{N-2}} f_{x(1+q)^{N-2}}(1+q) > \frac{1}{(1+q)^{N-2}} f_{x(1+q)^{N-2}}(1) \\ &= \frac{1}{(1+q)^{N-3}} f_{x(1+q)^{N-3}}(1+q) > \dots > f_x(1). \end{aligned}$$

□

ACKNOWLEDGMENTS

The author thanks the anonymous reviewers for their comments, which have greatly contributed to improve the presentation. This research work was supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT) of the Ministry of Science and Technology of Spain under research grant TAP99-0443-C05-05.

REFERENCES

- [1] *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, M. Abramowitz and I.A. Stegun, eds. Dover, 1965.
- [2] A. Bobbio and K.S. Trivedi, "An Aggregation Technique for the Transient Analysis of Stiff Markov Chains," *IEEE Trans. Computers*, vol. 35, no. 9, pp. 803-814, Sept. 1986.
- [3] A. Bobbio and M. Telek, "A Benchmark for PH Estimation Algorithms: Results for Acyclic-PH," *Comm. Statistics—Stochastic Models*, vol. 10, no. 3, pp. 661-677, 1994.
- [4] P.N. Bowerman, R.G. Nolty, and E.M. Scheuer, "Calculation of the Poisson Cumulative Distribution Function," *IEEE Trans. Reliability*, vol. 39, no. 2, pp. 158-161, 1990.
- [5] J.A. Carrasco, "Transient Analysis of Large Markov Models with Absorbing States Using Regenerative Randomization," Technical Report DMSD_99_2, Universitat Politècnica de Catalunya, Jan. 1999, available at <ftp://ftp-eel.upc.es/techreports>.
- [6] J.A. Carrasco, "Computation of Bounds for Transient Measures of Large Rewarded Markov Models using Regenerative Randomization," Technical Report DMSD_99_4, Universitat Politècnica de Catalunya, May 1999, available at <ftp://ftp-eel.upc.es/techreports>.
- [7] E. Çinlar, *Introduction to Stochastic Processes*. Prentice Hall, 1975.
- [8] B.L. Fox and P.W. Glynn, "Computing Poisson Probabilities," *Comm. ACM*, vol. 31, no. 4 pp. 440-445, Apr. 1988.
- [9] B.W. Johnson, *Design and Analysis of Fault Tolerant Digital Systems*. Addison-Wesley, 1989.
- [10] M. Kijima, *Markov Processes for Stochastic Modeling*. Cambridge: Univ. Press, 1997.
- [11] L. Knüsel, "Computation of the Chi-Square and Poisson Distribution," *SIAM J. Scientific and Statistical Computing*, vol. 7, no. 3, pp. 1022-1036, July 1986.

- [12] 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, pp. 1825-1841, Nov. 1994.
- [13] M. Malhotra, "A Computationally Efficient Technique for Transient Analysis of Repairable Markovian Systems," *Performance Evaluation*, vol. 24, no. 1-2, pp. 311-331, 1995.
- [14] B. Melamed and M. Yadin, "Randomization Procedures in the Computation of Cumulative-Time Distributions over Discrete State Markov Processes," *Operations Research*, vol. 32, no. 4, pp. 926-944, July-Aug. 1984.
- [15] D.R. Miller, "Reliability Calculation Using Randomization for Markovian Fault-Tolerant Computing Systems," *Proc. 13th IEEE Int'l Symp. Fault-Tolerant Computing (FTCS-13)*, pp. 284-289, June 1983.
- [16] A.P. Moorsel and W.H. Sanders, "Adaptive Uniformization," *Comm. Statistics—Stochastic Models*, vol. 10, no. 3, pp. 619-648, 1994.
- [17] A.P. Moorsel and W.H. Sanders, "Transient Solution of Markov Models by Combining Adaptive and Standard Uniformization," *IEEE Trans. Reliability*, vol. 46, no. 3, pp. 430-440, 1997.
- [18] F. Neuts, *Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach*. Dover, 1994.
- [19] A. Reibman and K.S. Trivedi, "Numerical Transient Analysis of Markov Models," *Computers and Operations Research*, vol. 15, no. 1, pp. 19-36, 1988.
- [20] S.M. Ross, *Stochastic Processes*. John Wiley & Sons, 1983.
- [21] B. Sericola, "Availability Analysis of Repairable Computer Systems and Stationarity Detection," *IEEE Trans. Computers*, vol. 48, no. 11, pp. 1166-1172, Nov. 1999.
- [22] M.R. Spiegel, *Mathematical Handbook of Formulas and Tables*. McGraw-Hill, 1970.



Juan A. Carrasco received the Engineering degree in industrial engineering from the Polytechnical University of Catalonia (UPC), Barcelona, Spain, in 1982, the MSc degree in computer science from Stanford University in 1987, and the PhD degree in industrial engineering from UPC, also in 1987. He is currently an associate professor in the Electronics Engineering Department of the UPC. He held a visiting position at the Institut National de Recherche en Informatique et en Automatique (INRIA-CNRS), Rennes, France, in 1996, working on bounding methods for Markovian dependability models and "on-the-fly" model solution techniques. He is currently coleader of a research team working on reliability and fault tolerance of electronic systems in the Electronics Engineering Department of UPC. His current research interests encompass methodologies for the modeling and evaluation of fault-tolerant systems, including bounding methods, numerical techniques, and combinatorial methods, topics in which he has published more than 40 papers in refereed journals and conference proceedings. Many of those methodologies have been incorporated in METFAC-2, a Markovian modeling tool developed under his direction. He has been principal investigator of several research projects funded by public and private institutions and is currently the UPC representative of the CABERNET network of excellence in distributed and dependable computing systems, funded by the European Commission. He is a member of the IEEE and the IEEE Computer Society.

► For more information on this or any computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.