Regenerative Randomization: Theory and application examples

Juan A. Carrasco and Angel Calderón
Departament d'Enginyería Electrònica
Universitat Politécnica de Catalunya
Diagonal 647, plta. 9, 08028 Barcelona, Spain

Abstract

Randomization is a popular method for the transient solution of continuous-time Markov models. Its primary advantages over other methods (i.e., ODE solvers) are robustness and ease of implementation. It is however well-known that the performance of the method deteriorates with the "stiffness" of the model: the number of required steps to solve the model up to time t tends to Λt for $\Lambda t \to \infty$. In this paper we present a new method called regenerative randomization and apply it to the computation of two transient measures for rewarded irreducible Markov models. Regarding the number of steps required in regenerative randomization we prove that: 1) it is smaller than the number of steps required in standard randomization when the initial distribution is concentrated in a single state, 2) for $\Lambda t \to \infty$, it is upper bounded by a function $O(\log(\Lambda t/\epsilon))$, where ϵ is the desired relative approximation error bound. Using dependability and performability examples we analyze the performance of the method.

Keywords: Randomization, continuous-time Markov chains, regenerative models, transient solution.

1 Introduction

Continuous-time Markov chains (CTMC's) are often 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 model is large. This makes the development of efficient transient analysis techniques for CTMC's a research topic of great interest. Commonly used methods are ODE solvers and randomization. Good recent reviews can be found in [Rei88] and [Mal94]. The randomization method (also called uniformization) is numerically very stable and easy to implement. It was first proposed by Jensen [Jen53] and has been applied to analyze performance models [Gra77], [Gro84], dependability models [Koh82], [Rei88], and performability models [Qur93]. The randomization method is based on the following result. Let $X = \{X(t); t \geq 0\}$ be a CTMC with state space Ω; let λ_{ij}, $i, j \in \Omega$ be the transition rates of X and let $\lambda_i = \sum_{j \in \Omega} \lambda_{ij}$, $i \in \Omega$ be the output rates of X. Consider any $\Lambda \geq \max_{i \in \Omega} \lambda_i$ (usually Λ is taken to be $\max_{i \in \Omega} \lambda_i$), and define the discrete-time Markov chain (DTMC) $Y = \{Y_k; k = 0, 1, 2, \ldots\}$ with same state space and initial distribution as X and jump probabilities $q_{ij} = \lambda_{ij}/\Lambda$, $i \neq j$, $q_{ii} = 1 - \lambda_i/\Lambda$. Let $N = \{N(t); t \geq 0\}$ be a Poisson process with arrival rate Λ. Then, $X(t) = Y_{N(t)}$. A recent, short proof of the result with very general conditions on X can be found in [Dij90]. These conditions are satisfied if, as we assume, X is finite.

The randomization result can be exploited to compute several transient measures. In this paper we will consider two measures which can be defined assuming a reward rate structure $r_i \geq 0$, $i \in \Omega$ on X. Furthermore, we will assume that X is irreducible. These measures are the transient reward rate at time t, $TRR(t) = E[r_{X(t)}]$ and the mean reward rate in $[0, t]$, $MRR(t) = E[\int_0^t r_{X(\tau)} d\tau]/t$. Many dependability, performance and performability measures can be expressed as particular cases of $TRR(t)$ and $MRR(t)$ by appropriate selections of the reward rate structure. More complex transient measures can also be computed using randomization such as the distribution of the interval availability [Sou86], [Rub93] and the distribution of performability [Sou89]. These measures are not considered in this paper. Using the randomization result, the transient probabilities of X can be written as:

$$P[X(t) = i] = \sum_{k=0}^{\infty} P[Y_k = i] \frac{(\Lambda t)^k}{k!} e^{-\Lambda t}.$$

Then, $TRR(t)$ can be expressed as:

$$\begin{aligned} TRR(t) &= E[r_{X(t)}] = \sum_{i \in \Omega} r_i P[X(t) = i] \\ &= \sum_{k=0}^{\infty} d(k) \frac{(\Lambda t)^k}{k!} e^{-\Lambda t}, \end{aligned} \tag{1}$$

with

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

In a practical implementation of the randomization method, the infinite series (1) is approximated by truncating the series up to a given number of randomization steps, K, and the approximation error is bounded. Letting $r_{max} = \max_{i \in \Omega} r_i$:

$$\begin{aligned} TRR^a(t) &= \sum_{k=0}^{K} d(k) \frac{(\Lambda t)^k}{k!} e^{-\Lambda t}, \\ TRR^\epsilon(t) &= r_{max} \sum_{k=K+1}^{\infty} \frac{(\Lambda t)^k}{k!} e^{-\Lambda t}, \\ TRR(t) - TRR^a(t) &\leq TRR^\epsilon(t), \end{aligned} \tag{2}$$

SIGMETRICS '95, Ottawa, Ontario, Canada
© 1995 ACM 0-89791-695-6/95/0005 ..$3.50

where $TRR^a(t)$ is the approximated solution at t and $TRR^\epsilon(t)$ is the upper bound for the approximation error.

For $MRR(t)$, randomization gives:

$$\begin{aligned} MRR(t) &= E[\int_0^t r_{X(\tau)} d\tau]/t \\ &= \frac{1}{t}\int_0^t \sum_{i\in\Omega} r_i P[X(\tau)=i] d\tau \\ &= \frac{1}{t}\sum_{k=0}^{\infty} d(k) \int_0^t \frac{(\Lambda\tau)^k}{k!} e^{-\Lambda\tau} d\tau \\ &= \frac{1}{\Lambda t}\sum_{k=0}^{\infty} d(k) \sum_{l=k+1}^{\infty} \frac{(\Lambda t)^l}{l!} e^{-\Lambda t}, \end{aligned}$$

yielding the approximation and error formulae:

$$MRR^a(t) = \frac{1}{\Lambda t}\sum_{k=0}^{K} d(k) \sum_{l=k+1}^{\infty} \frac{(\Lambda t)^l}{l!} e^{-\Lambda t},$$

$$\begin{aligned} MRR^\epsilon(t) &= \frac{r_{max}}{\Lambda t}\sum_{k=K+1}^{\infty}\sum_{l=k+1}^{\infty} \frac{(\Lambda t)^l}{l!} e^{-\Lambda t} \\ &= \frac{r_{max}}{\Lambda t}\sum_{k=K+2}^{\infty} (k-K-1)\frac{(\Lambda t)^k}{k!} e^{-\Lambda t}. \end{aligned}$$

The infinite series appearing in the equations for $TRR^\epsilon(t)$, $MRR^a(t)$ and $MRR^\epsilon(t)$ can be approximated with arbitrary accuracy taken a large enough M as:

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

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

$$\sum_{k=K+2}^{\infty} (k-K-1)\frac{(\Lambda t)^k}{k!} e^{-\Lambda t} \approx$$

$$\sum_{k=K+2}^{M} (k-K-1)\frac{(\Lambda t)^k}{k!} e^{-\Lambda t} + \frac{\Lambda t}{1-\frac{\Lambda t}{M}} \frac{(\Lambda t)^M}{M!} e^{-\Lambda t}.$$

The computational efficiency of the randomization method to compute $TRR(t)$ is mainly determined by the dependence of the infinite sum $\sum_{k=K+1}^{\infty}((\Lambda t)^k/k!)e^{-\Lambda t}$ on K. Using the well-known result that $N(t)$ has an asymptotic normal distribution with mean and variance Λt, it is easy to show that the required number of randomization steps, K, tends to Λt for $\Lambda t \to \infty$, independently on the required approximation error. For large models, almost all the computational effort of randomization is due to the computation of the transient solution of Y, and the method can be very slow when Λt is large. Basically identical problems are exhibited by the randomization method for $MRR(t)$.

Several approaches have been proposed to speed up the randomization method. One of them [Rei88] involves going through the transient regime of Y for the steps k yielding negligible contributions to the randomization formula using multisteps. Denoting by Q the transition probability matrix of Y and by $\pi(k)$ the probability distribution row vector of Y at step k, we have $\pi(k+S) = \pi(k)Q^S$, where S is the length of the multistep. Thus, computing explicitly matrices Q^S it is possible to reduce significantly the number of matrix-vector multiplications. The approach can be very efficient when Q is dense. However, when Q is sparse Q^S can be much denser than Q, and the number of floating point operations may still be large. Adaptive uniformization [Moo93] is a recent method in which the randomization rate is adapted depending on the states in which the randomized DTMC can be in at a given step. For some models adaptive randomization can be faster than the standard method. In addition, it can be used to solve models with infinite state spaces and not uniformly bounded transition rates. Another recent proposal to speed up the randomization method is steady-state detection [Mal94]. The technique can reduce substantially the computational effort of the standard method, but it is difficult to implement.

In this paper we propose a new randomization method called *regenerative randomization* (in the following SR will denote the standard method and RR our method) which can be used to compute the transient regime of irreducible finite CTMC's. The method is so called because it exploits the regenerative structure of the model. Unlike SR, RR has a *benign* behavior for large Λt. Furthermore, for certain, very common, conditions we prove that RR requires at most the same number of randomization steps than SR *for any* t. The rest of the paper is organized as follows. In Section 2 we derive the regenerative randomization method and discuss the optimization of its two truncation parameters. Section 3 analyzes the theoretical properties of the method. Section 4 illustrates the performance of the method with dependability and performability examples. Section 5 concludes the paper and highlights directions in which the work can be extended.

2 Regenerative randomization

Regenerative randomization requires the selection of a regenerative state $u \in \Omega$. Let $\alpha_i = P[X(0)=i] = P[Y_0=i]$, $i \in \Omega$. Let $Z = \{Z_k; k = 0,1,2,\ldots\}$ be the transient DTMC following the DTMC Y from u till reentry in u. Let $\Omega_u = \Omega - u$. The state space of Z is $\Omega \cup \{ab\}$, where ab is an absorbing state, $Z_0 = u$ and the transition probabilities of Z are:

$$P[Z_{k+1}=j|Z_k=i] = P[Y_{k+1}=j|Y_k=i], \quad i \in \Omega, j \in \Omega_u,$$

$$P[Z_{k+1}=ab|Z_k=i] = P[Y_{k+1}=u|Y_k=i], \quad i \in \Omega.$$

$$P[Z_{k+1}=ab|Z_k=ab] = 1.$$

Let $Z' = \{Z'_k; k = 0,1,2,\ldots\}$ be the transient DTMC with state space $\Omega_u \cup \{ab\}$, where ab is an absorbing state, following Y except that Z' remains in ab if Y has been in u. The initial distribution of Z' is $P[Z'_0=i] = \alpha_i$, $i \in \Omega_u$, $P[Z'_0=ab] = \alpha_u$ and its transition probabilities are:

$$P[Z'_{k+1}=j|Z'_k=i] = P[Y_{k+1}=j|Y_k=i], \quad i,j \in \Omega_u,$$

$$P[Z'_{k+1}=ab|Z'_k=i] = P[Y_{k+1}=u|Y_k=i], \quad i \in \Omega_u,$$

$$P[Z'_{k+1}=ab|Z'_k=ab] = 1.$$

In the following let $\pi_i(k) = P[Z_k=i]$, $\pi'_i(k) = P[Z'_k=i]$.

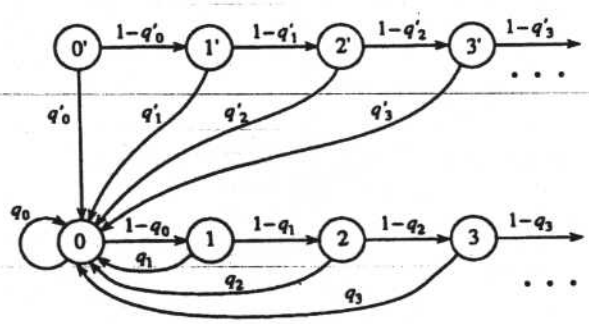

Figure 1: State transition diagram of the DTMC W.

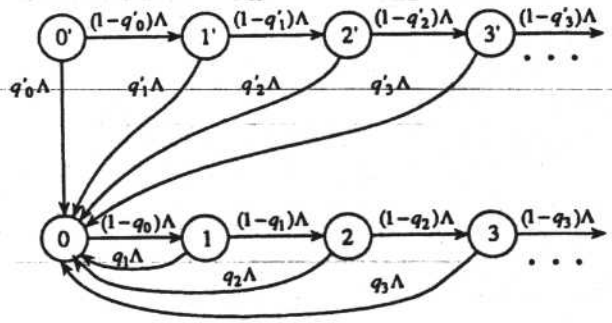

Figure 2: State transition diagram of the CTMC V.

The regenerative randomization formula is obtained considering the discrete-time stochastic process $W = \{W_k; k = 0, 1, 2, \ldots\}$, with state space $\{0, 1, 2, \ldots\} \cup \{0', 1', 2', \ldots\}$, defined from Y as follows:

$$W_k = \begin{cases} l & \text{if } Y_{k-l} = u \wedge Y_m \neq u, k-l < m \leq k \\ k' & \text{if } Y_m \neq u, 0 \leq m \leq k \end{cases}.$$

Informally, $W_k = l$ if Y has visited u by time k and has made its last visit to u l steps before the current step, k, and $W_k = k'$ if Y has not visited u by time k.

Proposition 1 *For $k \geq 0$ let $a(k) = \sum_{i \in \Omega} \pi_i(k)$, $a'(k) = \sum_{i \in \Omega_u} \pi'_i(k)$, $q_k = 1 - a(k+1)/a(k)$, $q'_k = 1 - a'(k+1)/a'(k)$. Then W is an (homogeneous) DTMC chain with initial probability distribution $P[W_0 = 0] = \alpha_u$, $P[W_0 = 0'] = 1 - \alpha_u$, $P[W_0 = i] = 0$, $i \notin \{0, 0'\}$ and the state transition diagram shown in Figure 1.*

Proof The initial distribution of W follows immediately from its definition. It is also clear that:
1) $W_k \in \{0, 1, \ldots, k, k'\}$, 2) from a state l, W can only jump to states 0 and $l+1$, and 3) from a state k', W can only jump to states 0 and $(k+1)'$. We will compute those transition probabilities and show that they only depend on the "from" state.
Case a ($W_k = l, l = 0, 1, \ldots, k$):

$$P[W_{k+1} = l+1 | W_k = l] = \frac{P[W_k = l \wedge W_{k+1} = l+1]}{P[W_k = l]}$$

$$= \frac{P[Y_{k-l} = u \wedge Y_m \neq u, k-l < m \leq k+1]}{P[Y_{k-l} = u \wedge Y_m \neq u, k-l < m \leq k]}$$

$$= \frac{P[Y_m \neq u, k-l < m \leq k+1 | Y_{k-l} = u]}{P[Y_m \neq u, k-l < m \leq k | Y_{k-l} = u]}$$

$$= \frac{P[Z_{l+1} \neq ab]}{P[Z_l \neq ab]} = \frac{\sum_{i \in \Omega} \pi_i(l+1)}{\sum_{i \in \Omega} \pi_i(l)} = \frac{a(l+1)}{a(l)} = 1 - q_l,$$

$$P[W_{k+1} = 0 | W_k = l] = 1 - P[W_{k+1} = l+1 | W_k = l] = q_l.$$

Case b ($W_k = k'$):

$$P[W_{k+1} = (k+1)' | W_k = k']$$

$$= \frac{P[W_k = k' \wedge W_{k+1} = (k+1)']}{P[W_k = k']}$$

$$= \frac{P[Y_m \neq u, 0 \leq m \leq k+1]}{P[Y_m \neq u, 0 \leq m \leq k]}$$

$$= \frac{P[Z'_{k+1} \neq ab]}{P[Z'_k \neq ab]} = \frac{\sum_{i \in \Omega_u} \pi'_i(k+1)}{\sum_{i \in \Omega_u} \pi'_i(k)}$$

$$= \frac{a'(k+1)}{a'(k)} = 1 - q'_k,$$

$$P[W_{k+1} = 0 | W_k = k']$$

$$= 1 - P[W_{k+1} = (k+1)' | W_k = k'] = q'_k \bigcirc.$$

The regenerative randomization formula can now be obtained from the standard randomization formula in two steps. The first step involves refining the state description of Y according to W. The second step involves the derandomization of W into a CTMC V. This allows to express the transient probabilities of X in terms of the transient probabilities of the DTMC's Z, Z' and the CTMC V. The final result is expressed by the following theorem; the theoretical developments are given in the proof. I_C denotes the indicator function returning the value 1 when condition C is true and value 0 when it is false.

Theorem 1 *Let $V = \{V(t); t \geq 0\}$ be the CTMC with state space $\{0, 1, 2, \ldots\} \cup \{0', 1', 2', \ldots\}$, initial distribution $P[V(0) = 0] = \alpha_u$, $P[V(0) = 0'] = 1 - \alpha_u$, $P[V(0) = i] = 0$, $i \notin \{0, 0'\}$, and the state transition diagram shown in Figure 2. Then:*

$$P[X(t) = i] = I_{i \neq u} \sum_{k=0}^{\infty} \frac{\pi'_i(k)}{a'(k)} P[V(t) = k'] + \sum_{k=0}^{\infty} \frac{\pi_i(k)}{a(k)} P[V(t) = k]. \quad (3)$$

Proof Starting from the SR formula and refining the state description of Y according to W:

$$P[X(t) = i] = \sum_{k=0}^{\infty} P[Y_k = i] P[N(t) = k]$$

$$= \sum_{k=0}^{\infty} \Big(P[Y_k = i \wedge W_k = k'] + \sum_{l=0}^{k} P[Y_k = i \wedge W_k = l] \Big) P[N(t) = k]. \quad (4)$$

The joint probability distribution of Y, W can be expressed in terms of the probability distributions of Y, Z and Z' as follows. First, consider the case $W_k = k'$. Clearly:

$$P[Y_k = u \wedge W_k = k'] = 0. \quad (5)$$

For $i \in \Omega_u$:

$$P[Y_k = i \wedge W_k = k'] = P[Y_k = i \wedge Y_m \neq u, 0 \leq m < k]$$
$$= P[Z'_k = i] = \pi'_i(k).$$

Adding for all $i \in \Omega_u$:

$$\begin{aligned} P[W_k = k'] &= \sum_{i \in \Omega_u} P[Y_k = i \wedge W_k = k'] \\ &= \sum_{i \in \Omega_u} \pi'_i(k) = a'(k), \end{aligned}$$

and dividing the equations:

$$P[Y_k = i \wedge W_k = k'] = \frac{\pi'_i(k)}{a'(k)} P[W_k = k'], \quad i \in \Omega_u. \quad (6)$$

For the case $W_k = l$, $0 \le l \le k$:

$$\begin{aligned} &P[Y_k = i \wedge W_k = l] \\ &= P[Y_{k-l} = u \wedge Y_m \neq u, k-l < m \le k \wedge Y_k = i] \\ &= P[Y_{k-l} = u] P[Z_l = i] = P[Y_{k-l} = u] \pi_i(l). \end{aligned}$$

Adding for all $i \in \Omega$:

$$\begin{aligned} P[W_k = l] &= \sum_{i \in \Omega} P[Y_k = i \wedge W_k = l] \\ &= \sum_{i \in \Omega} P[Y_{k-l} = u] \pi_i(l) \\ &= P[Y_{k-l} = u] \sum_{i \in \Omega} \pi_i(l) = P[Y_{k-l} = u] a(l), \end{aligned}$$

and dividing the equations:

$$P[Y_k = i \wedge W_k = l] = \frac{\pi_i(l)}{a(l)} P[W_k = l], \quad i \in \Omega. \quad (7)$$

Plugging (5),(6),(7) into (4), splitting terms and exchanging the indices k, l in the second summation:

$$\begin{aligned} P[X(t) = i] &= \sum_{k=0}^{\infty} \left(I_{i \neq u} \frac{\pi'_i(k)}{a'(k)} P[W_k = k'] \right. \\ &\qquad \left. + \sum_{l=0}^{k} \frac{\pi_i(l)}{a(l)} P[W_k = l] \right) P[N(t) = k] \\ &= I_{i \neq u} \sum_{k=0}^{\infty} \frac{\pi'_i(k)}{a'(k)} P[W_k = k'] P[N(t) = k] \\ &\qquad + \sum_{k=0}^{\infty} \frac{\pi_i(k)}{a(k)} \sum_{l=k}^{\infty} P[W_l = k] P[N(t) = l] \quad (8) \end{aligned}$$

Now consider the "derandomized" CTMC $V = \{V(t); t \ge 0\}$ defined as $V(t) = W_{N(t)}$. V has obviously the same state space and initial probability distribution as W; its state transition diagram is given in Figure 2. Using the standard randomization equation for V and considering that $P[W_l = k'] = 0$ for $l \neq k$ and $P[W_l = k] = 0$ for $l < k$:

$$\begin{aligned} P[V(t) = k'] &= \sum_{l=0}^{\infty} P[W_l = k'] P[N(t) = l] \\ &= P[W_k = k'] P[N(t) = k], \end{aligned}$$

$$\begin{aligned} P[V(t) = k] &= \sum_{l=0}^{\infty} P[W_l = k] P[N(t) = l] \\ &= \sum_{l=k}^{\infty} P[W_l = k] P[N(t) = l]. \end{aligned}$$

Substituting in (7), (8):

$$\begin{aligned} P[X(t) = i] &= I_{i \neq u} \sum_{k=0}^{\infty} \frac{\pi'_i(k)}{a'(k)} P[V(t) = k'] \\ &\quad + \sum_{k=0}^{\infty} \frac{\pi_i(k)}{a(k)} P[V(t) = k] \bigcirc. \end{aligned}$$

Using the regenerative randomization formula given by Theorem 1 we can easily express $TRR(t)$ and $MRR(t)$ in terms of the transient distributions of Z, Z' and V. As with SR, the formula contains infinite series which have to be truncated and the approximation error bounded. Since $MRR(t) = (1/t) \int_0^t TRR(\tau) d\tau$, we start with the measure $TRR(t)$.

Let $b(k) = \sum_{i \in \Omega} r_i \pi_i(k)$, $b'(k) = \sum_{i \in \Omega_u} r_i \pi'_i(k)$, $c(k) = b(k)/a(k)$, $c'(k) = b'(k)/a'(k)$. Using (3):

$$\begin{aligned} TRR(t) &= \sum_{i \in \Omega} r_i P[X(t) = i] \\ &= \sum_{i \in \Omega_u} r_i \sum_{k=0}^{\infty} \frac{\pi'_i(k)}{a'(k)} P[V(t) = k'] \\ &\quad + \sum_{i \in \Omega} r_i \sum_{k=0}^{\infty} \frac{\pi_i(k)}{a(k)} P[V(t) = k] \\ &= \sum_{k=0}^{\infty} \frac{\sum_{i \in \Omega_u} r_i \pi'_i(k)}{a'(k)} P[V(t) = k'] \\ &\quad + \sum_{k=0}^{\infty} \frac{\sum_{i \in \Omega} r_i \pi_i(k)}{a(k)} P[V(t) = k] \\ &= \sum_{k=0}^{\infty} \frac{b'(k)}{a'(k)} P[V(t) = k'] + \sum_{k=0}^{\infty} \frac{b(k)}{a(k)} P[V(t) = k] \\ &= \sum_{k=0}^{\infty} c'(k) P[V(t) = k'] + \sum_{k=0}^{\infty} c(k) P[V(t) = k] \end{aligned}$$

The infinite summations of the exact RR formula will be truncated assuming that Z has been solved up to step K and Z' has been solved up to step L. The parameters K, L control the computational effort of RR and the approximation error. To that end, consider the CTMC $V_{K,L} = \{V_{K,L}(t); t \ge 0\}$ with state space $\{0, 1, \ldots, K\} \cup \{0', 1', \ldots, L'\} \cup \{ab\}$ defined as follows: $V_{K,L}(t) = V(t)$ if $V(\tau) \in \{0', 1', \ldots, (L-1)'\} \cup \{0, 1, \ldots, K-1\}$ for $\tau \in [0, t]$, or $V(t) = L'$ and L' has been entered only once in $[0, t]$, or $V(t) = K$ and K has been entered only once in $[0, t]$; $V_{K,L}(t) = ab$ otherwise

The state transition diagram of $V_{K,L}$ is given in Figure 3. Note that the computation of the transition probabilities of $V_{K,L}$ and the reward structure $c(k)$, $0 \le k \le K$; $c'(k)$, $0 \le k \le L$ can be done with the assumed truncated transient solutions of Z, Z'. ab is an absorbing state representing the behavior of V not captured by the states $\{0, 1, \ldots, K\} \cup \{0', 1', \ldots, L'\}$ of $V_{K,L}$. Since

$$P[V_{K,L}(t) = k'] \le P[V(t) = k'], \quad 0 \le k \le L,$$

$$P[V_{K,L}(t) = k] \le P[V(t) = k], \quad 0 \le k \le K,$$

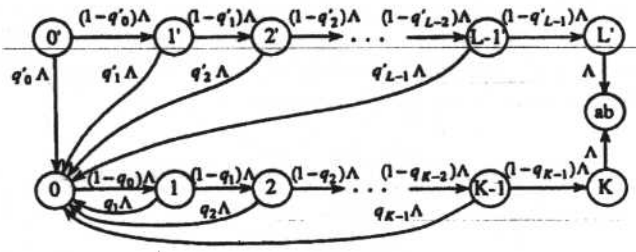

Figure 3: State transition diagram of the CTMC $V_{K,L}$

a lower bound to $TRR(t)$ is given by:

$$TRR^a(t) = \sum_{k=0}^{L} c'(k) P[V_{K,L}(t) = k'] + \sum_{k=0}^{K} c(k) P[V_{K,L}(t) = k]. \quad (9)$$

To bound the approximation error we can use $c'(k), c(k) \leq r_{max}$ to obtain:

$$TRR(t) - TRR^a(t) \leq r_{max} P[V_{K,L} = ab] = TRR^\epsilon(t) \quad (10)$$

The model $V_{K,L}$ can be solved in the Laplace transform domain, inverting the terms for which an easy antitransform can be derived, yielding (see [Car94] for details):

$$TRR^a(t) = \sum_{k=0}^{L} b'(k) \frac{(\Lambda t)^k}{k!} e^{-\Lambda t} + \mathcal{L}^{-1}\left\{ \tilde{p}_0(s) \sum_{k=0}^{K} b(k) \left(\frac{\Lambda}{s+\Lambda}\right)^k \right\}, \quad (11)$$

$$TRR^\epsilon(t) = r_{max} \left[a'(L) \sum_{k=L+1}^{\infty} \frac{(\Lambda t)^k}{k!} e^{-\Lambda t} + a(K)\Lambda \mathcal{L}^{-1}\left\{ \frac{\tilde{p}_0(s)}{s} \left(\frac{\Lambda}{s+\Lambda}\right)^K \right\} \right], \quad (12)$$

$$\tilde{p}_0(s) = \frac{1 - \frac{s}{s+\Lambda} \sum_{k=0}^{L} a'(k) \left(\frac{\Lambda}{s+\Lambda}\right)^k - a'(L) \left(\frac{\Lambda}{s+\Lambda}\right)^{L+1}}{s \sum_{k=0}^{K} a(k) \left(\frac{\Lambda}{s+\Lambda}\right)^k + a(K)\Lambda \left(\frac{\Lambda}{s+\Lambda}\right)^K}. \quad (13)$$

The extension to $MRR(t)$ is easily done noting that $MRR(t) = (1/t)\int_0^t TRR(\tau)d\tau$. Using $\int_0^t ((\Lambda\tau)^k/k!)e^{-\Lambda\tau}d\tau = (1/\Lambda)\sum_{l=k+1}^{\infty}((\Lambda t)^l/l!)e^{-\Lambda t}$:

$$\begin{aligned} MRR^a(t) &= \frac{1}{t}\int_0^t TRR^a(\tau)d\tau \\ &= \sum_{k=0}^{L} \frac{b'(k)}{t} \int_0^t \frac{(\Lambda\tau)^k}{k!} e^{-\Lambda\tau} d\tau \\ &\quad + \frac{1}{t}\mathcal{L}^{-1}\left\{ \frac{\tilde{p}_0(s)}{s} \sum_{k=0}^{K} b(k) \left(\frac{\Lambda}{s+\Lambda}\right)^k \right\} \\ &= \sum_{k=0}^{L} \frac{b'(k)}{\Lambda t} \sum_{l=k+1}^{\infty} \frac{(\Lambda t)^l}{l!} e^{-\Lambda t} \\ &\quad + \frac{1}{t}\mathcal{L}^{-1}\left\{ \frac{\tilde{p}_0(s)}{s} \sum_{k=0}^{K} b(k) \left(\frac{\Lambda}{s+\Lambda}\right)^k \right\} \quad (14) \end{aligned}$$

For $MRR^\epsilon(t)$:

$$\begin{aligned} MRR^\epsilon(t) &= \frac{1}{t}\int_0^t TRR^\epsilon(\tau)d\tau \\ &= r_{max}\left[\frac{a'(L)}{t} \sum_{k=L+1}^{\infty} \int_0^t \frac{(\Lambda\tau)^k}{k!} e^{-\Lambda\tau} d\tau \right. \\ &\quad \left. + \frac{a(K)\Lambda}{t}\mathcal{L}^{-1}\left\{ \frac{\tilde{p}_0(s)}{s^2} \left(\frac{\Lambda}{s+\Lambda}\right)^K \right\} \right]. \end{aligned}$$

Working out the first term:

$$\begin{aligned} \sum_{k=L+1}^{\infty} \int_0^t \frac{(\Lambda\tau)^k}{k!} e^{-\Lambda\tau} d\tau &= \frac{1}{\Lambda} \sum_{k=L+1}^{\infty} \sum_{l=k+1}^{\infty} \frac{(\Lambda t)^l}{l!} e^{-\Lambda t} \\ &= \frac{1}{\Lambda} \sum_{k=L+2}^{\infty} (k-L-1) \frac{(\Lambda t)^k}{k!} e^{-\Lambda t}, \end{aligned}$$

yielding:

$$\begin{aligned} MRR^\epsilon(t) = & \\ & r_{max}\left[\frac{a'(L)}{\Lambda t} \sum_{k=L+2}^{\infty} (k-L-1) \frac{(\Lambda t)^k}{k!} e^{-\Lambda t} \right. \\ & \left. + \frac{a(K)\Lambda}{t}\mathcal{L}^{-1}\left\{ \frac{\tilde{p}_0(s)}{s^2} \left(\frac{\Lambda}{s+\Lambda}\right)^K \right\} \right]. \quad (15) \end{aligned}$$

RR involves in general two truncation parameters. This opens the issue of optimizing the distribution of the total number of steps. We next describe a simple but efficient optimization method which can be implemented with little effort. The method is based on the following observation. The approximation error bound can be split into two contributions: one corresponding to the entries of $V_{K,L}$ in ab from L', and the other corresponding to the entries of $V_{K,L}$ in ab from K. For instance, for the $TRR(t)$ measure these contributions are (12):

$$TRR^\epsilon_1(t) = r_{max} a'(L) \sum_{k=L+1}^{\infty} \frac{(\Lambda t)^k}{k!} e^{-\Lambda t},$$

$$TRR^\epsilon_2(t) = r_{max} a(K)\Lambda \mathcal{L}^{-1}\left\{ \frac{\tilde{p}_0(s)}{s} \left(\frac{\Lambda}{s+\Lambda}\right)^K \right\}.$$

A path analysis of $V_{K,L}$ makes clear that the former decreases when L increases, while the second decreases when K increases but increases with L. This suggests that the optimum distribution should be close to the distribution for which both contributions are approximately equal. Then, starting with the distribution $(K = 1, L = 1)$ we increment K if the second contribution to the error approximation bound is greater than the first and increment L otherwise. This strategy is repeated at each (K, L) pair until the total approximation error bound satisfies the accuracy requirements. Using this strategy the contributions to the approximation error bound tend to equalize. Then, at worst, the total number of steps would be close to the minimum (with optimum distribution) number of steps required to achieve half the approximation error bound. Since the dependence of the number of steps on the approximation error bound is typically smooth, this guarantees in practice a behavior very close to the optimum.

3 Theoretical properties

In this section we explore the theoretical properties of regenerative randomization (RR) in comparison with standard randomization (SR). We start considering the case $P[X(0) = u] = 1$. In this case $a'(k) = b'(k) = 0$ and the RR formulae (11-13), (14-15) are simplified, requiring only the transient solution of the DTMC Z at the steps $k = 0, 1, \ldots, K$. We have the following result:

Theorem 2 *When $P[X(0) = u] = 1$ regenerative randomization requires at most the same number of steps as standard randomization for both* $\mathrm{TRR}(t)$ *and* $\mathrm{MRR}(t)$.

Proof Let $TRR^a_R(t)$, $TRR^\epsilon_R(t)$ denote the approximation and approximation error bound given by RR and let $TRR^a_S(t)$, $TRR^\epsilon_S(t)$ denote the approximation and approximation error bound given by SR. We will show that for the same truncation parameter K (the truncation parameter L is not involved in the RR formulae for the case considered) $TRR^a_R(t) \geq TRR^a_S(t)$ and $TRR^\epsilon_R(t) \leq TRR^\epsilon_S(t)$. This implies immediately the result for $TRR(t)$, and, since $MRR(t) = (1/t)\int_0^t TRR(\tau)d\tau$, it also implies the result for $MRR(t)$.

Consider the set of all paths P of the DTMC Y. A path $p \in P$ is a sequence of $L(p) + 1$ states, where $L(p)$ denotes the length of the path. Let $p(i)$, $0 \leq i \leq L(p)$ be the *ith* state of the path p, and let $P[p]$ be the probability of the path p, i.e.:

$$P[p] = P[Y_0 = p(0) \wedge Y_1 = p(1) \wedge \ldots \wedge Y_{L(p)} = p(L(p))].$$

Let P^k be the subset of P including all paths of length k and let P^k_i be the subset of P^k including the paths of length k which end in state i ($p(k) = i$). Clearly, $P[Y_k = i] = \sum_{p \in P^k_i} P[p]$. Then, we can write $TRR^a_S(t)$ as:

$$\begin{aligned} TRR^a_S(t) &= \sum_{k=0}^{K}\sum_{i\in\Omega} r_i P[Y_k = i]P[N(t) = k] \\ &= \sum_{k=0}^{K}\sum_{i\in\Omega} r_i \sum_{p\in P^k_i} P[p]P[N(t) = k] \\ &= \sum_{k=0}^{K}\sum_{p\in P^k} r_{p(k)} P[p]P[N(t) = k]. \quad (16) \end{aligned}$$

For $TRR^\epsilon_S(t)$ we have:

$$\begin{aligned} TRR^\epsilon_S(t) &= r_{max}\sum_{k=K+1}^{\infty} P[N(t) = k] \\ &= r_{max}\sum_{k=K+1}^{\infty}\sum_{p\in P^k} P[p]P[N(t) = k]. \quad (17) \end{aligned}$$

Let us define the DTMC $W_K = \{(W_K)_k; k = 0, 1, 2, \ldots\}$ with state space $\{0, 1, \ldots, K\} \cup \{ab\}$ and $(W_K)_0 = 1$ as follows: $(W_K)_k = W_k$ if $W_m \in \{0, 1, \ldots, K-1\}$ for $0 \leq m \leq k$, or $W_k = K$ and K has been entered only once; $(W_K)_k = ab$ otherwise.

W_K is the discrete parallel to $V_{K,L}$ for the particular case $X(0) = u$. Then, remembering the theoretical developments of the last section, we can write for RR:

$$TRR^a_R(t) = \sum_{k=0}^{\infty}\sum_{i\in\Omega} r_i P[Y_k = i \wedge (W_K)_k \neq ab]P[N(t) = k].$$

Let $P^{k,K}_i$ be the subset of P including the paths of length k which end in state i and do not give a realization of W_K entering the absorbing state of W_K. Let $P^{k,K} = \bigcup_{i\in\Omega} P^{k,K}_i$. We can rewrite the last equation as:

$$\begin{aligned} TRR^a_R(t) &= \sum_{k=0}^{\infty}\sum_{i\in\Omega} r_i \sum_{p\in P^{k,K}_i} P[p]P[N(t) = k] \\ &= \sum_{k=0}^{\infty}\sum_{p\in P^{k,K}} r_{p(k)} P[p]P[N(t) = k]. \end{aligned}$$

According to the definition of W_K, it is clear that W_K can only enter ab after at least $K + 1$ steps. Then, $P^{k,K} = P^k$ for $0 \leq k \leq K$ and, using (16):

$$\begin{aligned} TRR^a_R(t) &= \sum_{k=0}^{K}\sum_{p\in P^k} r_{p(k)} P[p]P[N(t) = k] \\ &\quad + \sum_{k=K+1}^{\infty}\sum_{p\in P^{k,K}} r_{p(k)} P[p]P[N(t) = k] \\ &\geq TRR^a_S(t). \end{aligned}$$

The approximation error bound for RR can be written in terms of path subset probabilities as:

$$\begin{aligned} TRR^\epsilon_R(t) &= r_{max}\sum_{k=0}^{\infty} P[(W_K)_k = ab]P[N(t) = k] \\ &= r_{max}\sum_{k=0}^{\infty}\sum_{p\in P^k - P^{k,K}} P[p]P[N(t) = k]. \end{aligned}$$

Considering again $P^{k,K} = P^k$ for $0 \leq k \leq K$, and using (17):

$$\begin{aligned} TRR^\epsilon_R(t) &= r_{max}\sum_{k=K+1}^{\infty}\sum_{p\in P^{k,K}} P[p]P[N(t) = k] \\ &\leq r_{max}\sum_{k=K+1}^{\infty}\sum_{p\in P^k} P[p]P[N(t) = k] \\ &= TRR^\epsilon_S(t)\,\bigcirc. \end{aligned}$$

While the case $X(0) = u$ considered in the previous theorem seems rather particular, the CTMCs X of interest very often have an initial probability distribution concentrated in a single state, say r. Then, we can take $u = r$ and invoke Theorem 2 to arrive immediately to the following result:

Corollary 1 *When X has an initial probability distribution concentrated in a single state, RR with an appropriate selection of the regenerative state requires at most the same number of steps as SR.*

We now study the limiting behavior of RR. As we will prove, RR has a "benign" limiting behavior; more specifically, we will prove that the number of required steps to achieve a relative approximation error bound smaller than ϵ is, for $\Lambda t \to \infty$, upper bounded by a function which is $O(\log(\Lambda t/\epsilon))$. According to this result we should expect a significant influence of ϵ on the number of required steps. This is confirmed by the experiments of Section 4

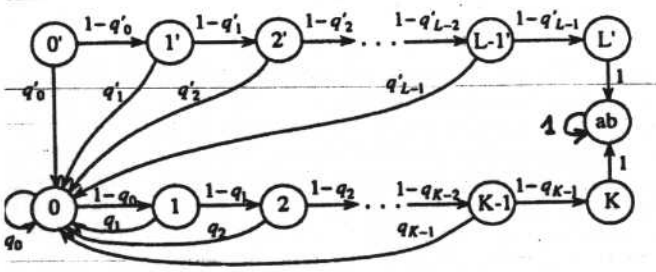

Figure 4: State transition diagram of $W_{K,L}$.

and contrasts with the behavior of SR, whose performance is asymptotically independent of ϵ for $\Lambda t \to \infty$. The proof of this result is achieved by finding suitable upper bounds for $TRR^{\epsilon}(t)$ and $MRR^{\epsilon}(t)$. These bounds are obtained in the following sequence of two propositions. Before that we note that $a(0) = P[Z_0 = u] = 1$, $a'(0) = \sum_{i \in \Omega_u} P[Y_0 = i] = 1 - \alpha_u$, and therefore:

$$\prod_{i=0}^{k-1}(1-q_i) = \prod_{i=0}^{k-1}\frac{a(i+1)}{a(i)} = \frac{a(k)}{a(0)} = a(k) \qquad (18)$$

$$\prod_{i=0}^{k-1}(1-q'_i) = \prod_{i=0}^{k-1}\frac{a'(i+1)}{a'(i)} = \frac{a'(k)}{a'(0)} = \frac{a'(k)}{1-\alpha_u} \qquad (19)$$

Proposition 2 *Let $W_{K,L} = \{(W_{K,L})_k; k = 0, 1, \ldots\}$ be the DTMC obtained from $V_{K,L}$ under randomization rate Λ, i.e., $V_{K,L}(t) = (W_{K,L})_{N(t)}$ (its state transition diagram is given in Figure 4). Then, $P[(W_{K,L})_k = ab] \leq I_{k>L}a'(L) + I_{k>K}(k-K)a(K)$.*

Proof Let $\mu(k) = P[(W_{K,L})_k = ab | (W_{K,L})_0 = 0]$. We start showing that $\mu(k) \leq I_{k>K}(k-K)a(K)$. To that end define:

$$\nu(k) = P[(W_{K,L})_k = 0 \wedge (W_{K,L})_m \neq 0, 1 \leq m < k \mid (W_{K,L})_0 = 0].$$

From the state transition diagram of $W_{K,L}$ and using (18):

$$\begin{aligned}\nu(k) &= q_{k-1}\prod_{i=0}^{k-2}(1-q_i) = \left(1 - \frac{a(k)}{a(k-1)}\right)a(k-1) \\ &= a(k-1) - a(k).\end{aligned}$$

Since at least $K+1$ steps are required to arrive to ab from 0, $\mu(k) = 0$ for $k \leq K$. For $k = K+1$ there exists only one path from 0 to ab with $K+1$ steps and

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

For $k > K+1$, the result $\mu(k) \leq I_{k>K}(k-K)a(K)$ is shown by complete induction on k. Let $i_{max} = \min\{K, k-K-1\}$. Classifying the paths with k steps which end in ab without revisiting 0 or reentering 0 after a number i of steps and using $a(0) \leq 1$:

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

Let

$$\psi(k) = P[(W_{K,L})_k = 0 \wedge (W_{K,L})_m \neq 0, 1 \leq m < k \mid (W_{K,L})_0 = 0'],$$

$$\phi(k) = P[(W_{K,L})_k = ab \wedge (W_{K,L})_m \neq 0, 1 \leq m < k \mid (W_{K,L})_0 = 0'].$$

From the state transition diagram it is clear that $\psi(k) = 0$ for $k > L$. For $1 \leq k \leq L$, using (19):

$$\begin{aligned}\psi(k) &= q'_{k-1}\prod_{i=0}^{k-2}(1-q'_i) = \left(1 - \frac{a'(k)}{a'(k-1)}\right)\frac{a'(k-1)}{1-\alpha_u} \\ &= \frac{a'(k-1) - a'(k)}{1-\alpha_u}.\end{aligned}$$

From the state transition diagram, $\phi(k) = 0$ for $k \leq L$. For $k > L$ (19):

$$\phi(k) = \prod_{i=0}^{L-1}(1-q'_i) = \frac{a'(L)}{1-\alpha_u},$$

yielding

$$\phi(k) = I_{k>L}\frac{a'(L)}{1-\alpha_u}.$$

Let $i_{max} = \min\{L, k-K-1\}$. Analyzing the behavior of $W_{K,L}$ according to its visits to the state 0 and using the results obtained so far:

$$\begin{aligned}&P[(W_{K,L})_k = ab] \\ &= (1-\alpha_u)\phi(k) + (1-\alpha_u)\sum_{i=1}^{i_{max}}\psi(i)\mu(k-i) + \alpha_u\mu(k) \\ &= I_{k>L}a'(L) + I_{k>K+1}\sum_{i=1}^{i_{max}}(a'(i-1) - a'(i))\mu(k-i) \\ &\quad + I_{k>K}\alpha_u\mu(k). \qquad (20)\end{aligned}$$

Bounding the second term, noting that $a'(i) \leq 1-\alpha_u$ and $i_{max} \leq k-K-1$:

$$\begin{aligned}&\sum_{i=1}^{i_{max}}(a'(i-1) - a'(i))\mu(k-i) \\ &\leq \sum_{i=1}^{i_{max}}(a'(i-1) - a'(i))(k-K-i)a(K)\end{aligned}$$

$$= a(K)\sum_{i=1}^{i_{max}}(a'(i-1)-a'(i))(k-K-i)$$

$$= a(K)\left[a'(0)+\sum_{i=1}^{i_{max}-1}a'(i)-(k-K-i_{max})a'(i_{max})\right]$$

$$\leq a(K)\sum_{i=0}^{i_{max}-1}a'(i) \leq (1-\alpha_u)(k-K-1)a(K).$$

Finally, substituting in (18):

$$\begin{aligned} P[(W_{K,L})_k = ab] &\leq I_{k>L}a'(L) + I_{k>K}\alpha_u(k-K)a(K) \\ &\quad + I_{k>K+1}(1-\alpha_u)(k-K-1)a(K) \\ &\leq I_{k>L}a'(L) + I_{k>K}(k-K)a(K)\,\bigcirc\,. \end{aligned}$$

Playing this result in the standard randomization equation, it is possible to upper bound $TRR^\epsilon(t)$ and $MRR^\epsilon(t)$ as follows.

Proposition 3 *The error approximation bounds of regenerative randomization satisfy* $\mathrm{TRR}^\epsilon(t) < r_{max}(a'(L)+a(K)\Lambda t)$, $\mathrm{MRR}^\epsilon(t) < r_{max}(a'(L)+(a(K)/2)\Lambda t)$.

Proof For $TRR^\epsilon(t)$:

$$\frac{TRR^\epsilon(t)}{r_{max}} = P[V_{K,L}(t)=ab]$$

$$= \sum_{k=0}^{\infty}P[(W_{K,L})_k = ab]P[N(t)=k]$$

$$\leq \sum_{k=L+1}^{\infty}a'(L)\frac{(\Lambda t)^k}{k!}e^{-\Lambda t} + \sum_{k=K+1}^{\infty}(k-K)a(K)\frac{(\Lambda t)^k}{k!}e^{-\Lambda t}$$

$$< a'(L) + a(K)\sum_{k=1}^{\infty}k\frac{(\Lambda t)^{K+k}}{(K+k)!}e^{-\Lambda t}$$

$$< a'(L) + a(K)\sum_{k=1}^{\infty}\frac{(\Lambda t)^{K+k}}{(K+k-1)!}e^{-\Lambda t}$$

$$= a'(L) + a(K)\Lambda t\sum_{k=1}^{\infty}\frac{(\Lambda t)^{K+k-1}}{(K+k-1)!}e^{-\Lambda t}$$

$$= a'(L) + a(K)\Lambda t\sum_{k=K}^{\infty}\frac{(\Lambda t)^k}{k!}e^{-\Lambda t} < a'(L) + a(K)\Lambda t.$$

For $MRR^\epsilon(t)$, remembering its relation with $TRR^\epsilon(t)$:

$$MRR^\epsilon(t) = \frac{1}{t}TRR^\epsilon(t) < \frac{r_{max}}{t}\int_0^t(a'(L)+a(K)\Lambda\tau)d\tau$$

$$= r_{max}(a'(L)+\frac{a(K)}{2}\Lambda t)\,\bigcirc\,.$$

We are now in position to show the second main theoretical result about RR.

Theorem 3 *The number of steps required by RR to achieve a relative approximation error bound smaller than ϵ is, for $\Lambda t \to \infty$, upper bounded by a function which is $O(log(\Lambda t/\epsilon))$, both for* TRR(t) *and* MRR(t).

Proof The proof uses as a basic tool results for the spectrum of non-negative matrices (see, for instance, [Cin75]). Let Q_R be the restriction of the transition probability matrix of Z to its reachable transient states. Let $Q_1, Q_2, \ldots, Q_M$ be the irreducible submatrices of the normal form of Q_R. Using Frobenius theorem, each Q_i has a real, positive and simple eigenvalue ρ_i such that any other eigenvalue ζ of Q_i satisfies $|\zeta| \leq \rho_i$. The equality can only be given in the case in which Q_i is periodic and, in that case, the eigenvalues ζ with $|\zeta| = \rho_i$ form with ρ_i a equally spaced rotated set of complex numbers. Also, each Q_i has rows which sum less than 1, then $\rho_i < 1$. Let $\rho = \max_{1\leq i\leq M}\rho_i$. Using this characterization of the eigenvalues of the submatrices Q_i of Q_R, it is easy to prove that $a(k) = \sum_{i\in\Omega}P[Z_k = i]$ is, for $k \to \infty$, upper bounded by a function of the form $A\rho^k$, with $A > 0$. Similarly for Z', let ρ'_i, $1 \leq i \leq M'$ be the "dominant" eigenvalues of the irreducible matrices of the normal form of Q'_R, the restriction of the transition probability matrix of Z' to its reachable transient states. Let $\rho' = \max_{1\leq i\leq M}\rho'_i$, then, for $k \to \infty$, $a'(k)$ is upper bounded by $A'\rho'^k$, with $A' > 0$.

Let us start with $TRR(t)$. Using Proposition 3 and the previous results, for $\Lambda t \to \infty$:

$$TRR^\epsilon(t) < r_{max}(A'\rho'^L + A\rho^K\Lambda t).$$

Let $S = K + L$. Since $K = L$ may not be the optimal distribution, the approximation error for $K = L = S/2$ upper bounds the approximation error $TRR^\epsilon(t)$ achieved by RR with optimal distribution of the steps for a given S. Let $\rho_m = \max\{\rho, \rho'\}$. Then:

$$TRR^\epsilon(t) < r_{max}(A'\rho'^{S/2} + A\rho^{S/2}\Lambda t) < r_{max}(A' + A\Lambda t)\rho_m^{S/2}.$$

Let RR be the steady-state reward rate of X. Clearly, $\lim_{\Lambda t\to\infty}TRR(t) = RR$, and the number of steps required to achieve a relative approximation error is upper bounded by the S^* satisfying:

$$r_{max}(A' + A\Lambda t)\rho_m^{S^*/2} = RR\epsilon.$$

Solving in S^*, considering $\rho_m < 1$:

$$S^* = \frac{2}{\log(1/\rho_m)}\log\left(\frac{r_{max}(A'+A\Lambda t)}{RR\epsilon}\right) = O(\log\frac{\Lambda t}{\epsilon}).$$

For $MRR(t)$:

$$MRR^\epsilon(t) < r_{max}(A' + A\Lambda t/2)\rho_m^{S/2},$$

and, considering that $\lim_{\Lambda t\to\infty}MRR(t) = RR$:

$$S^* = \frac{2}{\log(1/\rho_m)}\log\left(\frac{r_{max}(A'+A\Lambda t/2)}{RR\epsilon}\right) = O(\log\frac{\Lambda t}{\epsilon})\,\bigcirc\,.$$

4 Examples and numerical analysis

In this section we illustrate the properties of RR and compare its performance with SR using dependability and performability models of fault-tolerant systems. We use the number of randomization steps in Y for SR and the number of steps in Z and Z' $(K + L)$ for RR as comparison metric. RR requires the use of a Laplace inversion algorithm. We have used Crump's method [Cru76] with the parameter T of the inversion formula set to $2t$, other parameters set so that the relative antitransform error is smaller than 10^{-8}, and

a maximum number of Laplace abscissae to achieve convergence equal to 50. In general, between 20 and 30 transform abscissae have been enough to achieve convergence. RR is implemented by advancing K, L according to the heuristic described at the end of section 2, till the goal relative error is satisfied. Crump's method adds a time complexity $O(m^2)$ per step, where m is the number of abscissae. The significance of this overhead depends on the size of the CTMC X. For the relatively small models of the examples (a few hundred states) the overhead is around 30%. For large models, the overhead would be insignificant. The examples are:

Fault-tolerant database system: The system includes two front-ends, two databases and two processing subsystems, each one made up of a switch, a memory and two processors. The fault-tolerant database system is operational if at least one front-end and one database are unfailed and one processing subsystem is operational; a processing subsystem is operational if its switch, its memory and at least one processor are unfailed. Unfailed components of a non-operational processing subsystem and all unfailed components if the system is not operational become dormant. Dormant components do not fail. All components fail with constant rates. Front-ends and databases have failure rate 10^{-4}; switches and memories have failure rate 5×10^{-5}, and processors have failure rate 2×10^{-4}. When a processor fails it contaminates (fails) the operational databases with probability 0.01. For the repair rates we consider two sets of values. In the data set 1 all components are repaired with rate 1. In the data set 2, databases have repair rate 0.5, front-ends, switches and memories have repair rate 1, and processors have repair rate 5. There is only one repairman which follows a preemptive resume priority strategy with random selection of the component to repair among the failed components with the same repair priority. Front-ends and databases have the highest repair priority, followed next by switches and memories, followed by processors. The measure of interest is the transient unavailability $ua(t)$, which can be formalized as $TRR(t)$ by assigning a reward rate 1 to the down states and a reward rate 0 to the operational states.

Fault-tolerant server: This is a performability model of a fault-tolerant system which performs on-line fault detection by comparison. The architecture of the system is given in Figure 5. The system comprises two servers and a buffer of capacity C to hold arriving tasks (including those currently served). When both servers are thought to be unfailed, the server set is configured in dual mode if the number of tasks in the buffer is $\leq M$, and in simplex mode if the number of tasks in the buffer is $> M$. In dual mode, both servers execute the same task and compare the results. If they agree, the next task is served. If there is a miscomparison, the task is reexecuted in dual mode to provide transient fault detection. If the new results agree, a transient fault is assumed to have been the cause of the previous disagreement and the next task is served; otherwise, a permanent fault is assumed and the servers are diagnosed. To simplify the model we assume that the diagnosis coverage is 1. When the diagnosis finishes, the faulty servers are removed and substituted by servers taken from a an infinite server pool. In simplex mode there is no fault detection and faulty servers cause the tasks to be served erroneously. The system can also work with only one server active. This mode occurs, for instance, while a faulty server is replaced by a server taken from the server pool. In this mode there is no fault detection. There is also the possibility of task overflow. If a task arrives when the buffer is full, the task is simply rejected

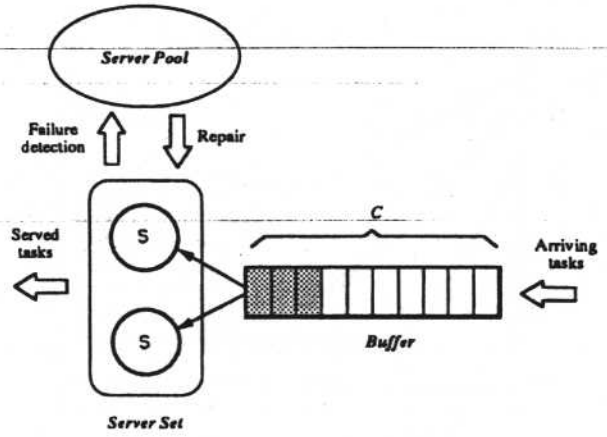

Figure 5: Fault tolerant server architecture.

and not served. Thus, there are two types of misbehavior: task rejection due to buffer overflow and tasks erroneously served due to undetected faults when the server set is not configured in dual mode. In general, selecting an appropriate value for M allows to balance both phenomena. A low value of M reduces the probability of buffer overflow while a large value of M reduces the probability that the server will not be configured in dual mode and, therefore, the rate of erroneously served tasks. We assume that the tasks have a 2-Erlang service distribution with mean $1/\psi$ when the server set is configured in simplex mode and $(1+\delta)/\psi$ when the server set is configured in dual mode (the parameter $\delta > 0$ accounts for the overhead due to comparison). Tasks arrive at rate ν. The other parameters of the model are the transient server failure rate λ_t, the permanent server failure rate λ_p, the server diagnosis rate μ_d and the server repair rate μ_r. To simplify the model, we assume that the buffer does not fail. This could be a realistic assumption for some implementations (error detecting, configurable memory, for instance). All the numerical results will be obtained for $C = 20$, $\psi = 1$, $\delta = 0.2$, $\lambda_t = 2 \times 10^{-4}$, $\lambda_p = 10^{-5}$, $\mu_d = 0.5$, $\mu_r = 0.5$, leaving free only the model parameters M and ν.

The measure of interest is the mean rate in the interval $[0, t]$ of tasks either rejected or erroneously served. We will denote this measure as $\overline{n}(t)$ and can be formalized as $MRR(t)$ by superimposing into the model two additive reward rate structures. The former assigns a reward rate ν to the states in which the buffer is full, and a reward rate 0 to the remaining states. The second assigns an impulse reward 1 to the transitions modeling the end of the service of a task in simplex mode by a faulty server (a server in permanent fault), an impulse reward equal to the probability of having some transient fault during service to the transitions modeling the end of the service of a task in simplex mode by a non-faulty server, and an impulse reward 0 to the remaining transitions of the model. These impulse rewards are multiplied by the corresponding transition rates and included as reward rates in the "from" state. As pointed out, often M has an optimum value. To illustrate this, Figure 6 gives the steady-state value of $\overline{n}(t)$ for several values of ν as a function of M. For $\nu = 0.2$ (light load) erroneous task processing due to latent server faults is dominant and the optimum M is high ($M = 17$). As the load increases task rejection becomes more important and the optimum M decreases: $M = 15$ for $\nu = 0.4$, and $M = 12$ for $\nu = 0.6$. It can also be noted that an increase in M beyond its optimal value has an impact

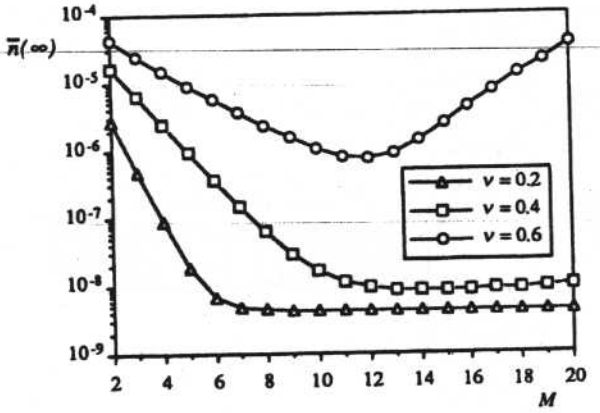

Figure 6: Steady-state value of $\overline{n}(t)$ for the fault-tolerant server for several values of ν as a function of M.

Table 1: Number of steps in SR and RR for the transient unavailability $ua(t)$ of the fault-tolerant database model for data set 1, initial state s_1, $u = s_1$, $\epsilon = 10^{-5}$ and several values of t.

t	$ua(t)$	steps	
		SR	RR
0.01	7.960×10^{-9}	5	3
0.1	7.616×10^{-7}	7	4
1	5.072×10^{-6}	13	5
10	8.059×10^{-6}	36	8
100	8.059×10^{-6}	170	9
10^3	8.059×10^{-6}	1,210	10
10^4	8.059×10^{-6}	10,659	12

which increases with the load.

We start our analysis of RR using the fault-tolerant database model. We consider two initial states: s_1 (the state in which all components are unfailed) and s_2 (the state in which one front-end is failed). Both are operational states and $ua(0) = 0$. Table 1 gives the number of required steps under SR and RR with a relative approximation error bound goal $\epsilon = 10^{-5}$ for the data set 1 (repair rates equal) and initial state s_1, taking $u = s_1$ for RR. Table 2 gives the same results for the data set 2 (repair rates unequal). In both cases u is the initial state and RR is guaranteed to require at most the same number of steps as SR. This is confirmed by the numerical results. We can also see the benign behavior of RR when t increases. For data set 1 RR is remarkably fast; for data set 2 RR is somehow slower but still significantly faster than SR, specially for large t. The difference in performance of RR in both cases can be explained by the dispersion of repair rates in the data set 2. In the first case the output rate of all the states with failed components is approximately equal to 1 and the DTMC's Z, Z' move to the absorbing state very fast; then, the quantities $a(k)$, $a'(k)$ decrease very rapidly with k and very few steps are enough to achieve a small approximation error.

The number of steps required under SR is independent on the approximation error for $\Lambda t \to \infty$. The asymptotic bound for the number of required steps under RR when $\Lambda t \to \infty$ given by Theorem 3 suggests a more sensitive behavior of RR in relation to ϵ. Figure 7 shows that behavior for the fault-tolerant database model, initial state s_1, $u = s_1$ and data set 2. For this data set $\Lambda = 5.001$ and the number of required steps S^* follows quite neatly a law $A + B \log((\Lambda t)/\epsilon)$ for large Λt. Thus, it seems that the upper bound for large Λt S^* used in the proof of Theorem 3 is characteristic of the actual behavior of RR and that, in RR, solution accuracy can be sensibly traded-off with computational effort, even for large Λt.

Table 2: Number of steps in SR and RR for the transient unavailability $ua(t)$ of the fault-tolerant database model for data set 2, initial state s_1, $u = s_1$, $\epsilon = 10^{-5}$ and several values of t.

t	$ua(t)$	steps	
		SR	RR
0.01	7.980×10^{-8}	6	4
0.1	7.806×10^{-7}	10	7
1	6.313×10^{-6}	25	18
10	1.601×10^{-5}	100	72
100	1.612×10^{-5}	647	181
10^3	1.612×10^{-5}	5,452	207
10^4	1.612×10^{-5}	51,424	230

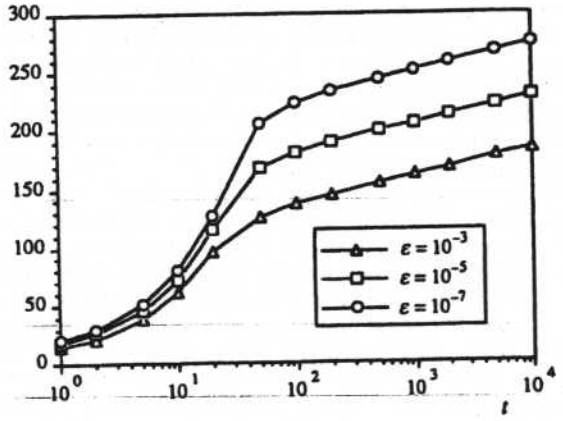

Figure 7: Number of steps in RR with $u = s_1$ for the transient unavailability $ua(t)$ of the fault-tolerant database model for data set 2, initial state s_1, as a function of ϵ and t.

We next consider the fault-tolerant database model with data set 2 and initial state s_2. s_2 is a state which is rarely visited in the DTMC Y. In such a situation, taking the initial state as u may degrade significantly the typical performance of RR, making it very close to SR even though RR is still theoretically better. For the fault-tolerant database model which is highly skewed to state s_1, the choice $u = s_1$ is the reasonable one. This is clearly illustrated by the results obtained with both choices which are given in Table 3. We also give the number of steps required under SR. RR with $u = s_2$ requires almost the same number of steps as SR for all t. In addition, for $t = 10^3, 10^4$ and the choice $(u = s_2)$, Crump's method did not converge, apparently due to cancellation errors caused by the very small rate at which $a(k)$ decreases. This lead us to suspect about the numerical stability of the method. However, by comparing the results given by the Laplace inversion algorithm with

able 3: Number of steps in SR, RR with $u = s_1$, and RR ith $u = s_2$ for the steady-state unavailability $ua(t)$ of the ault-tolerant database model for data set 2, initial state s_2, $= 10^{-5}$, and several values of t.

t	$ua(t)$	steps		
		SR	RR (s_1)	RR (s_2)
0.01	1.070×10^{-6}	6	8	5
0.1	9.835×10^{-6}	10	14	9
1	4.343×10^{-5}	24	36	23
10	1.614×10^{-5}	100	147	99
100	1.612×10^{-5}	647	346	644
10^3	1.612×10^{-5}	5,452	372	*
10^4	1.612×10^{-5}	51,424	395	*

Table 4: Distribution of steps among the truncation parameters of RR and optimal distribution for the fault-tolerant database model, data set 2, $u = s_1$, $\epsilon = 10^{-5}$, and several values of t.

t	steps	K	L
0.01	8	2	6
0.1	14	5	9
1	36	15	21
10	147	70	77
100	346	187	159
10^3	372	213	159
10^4	395	236	159

the solution of the $V_{K,L}$ model with SR we checked that our implementation of Crump's method was robust whenever it converged (the relative error for both the approximation and the error bound was below 10^{-8} in all cases we have tested). RR with $u = s_1$ requires slightly more steps for small t and significantly less for large t, exhibiting the "benign" behavior. Table 4 assesses the performance of the method for the optimization of the truncation parameters K, L. We give, for several values of t, the total number of steps and the final parameters K, L selected by the method. By exhaustively trying all distributions for each total number of steps we found out that the distributions selected by the heuristic optimization method are the optimum ones in all cases except for $t = 10^3$. In that case, the optimum distribution differs from the distribution reached by the method in only one step and the difference in the relative approximation error bounds achieved by the distributions is negligible. The excellent performance of the optimization heuristic is not fortunate coincidence. For large Λt the approximation error bound contributions follow very closely laws $A\rho_m^k$, $A'\rho_m^k$, where ρ_m is the (for this model, common) dominant eigenvalue of the transition probability matrices of Z and Z' (see proof of Theorem 3). A relatively simple analysis which is omitted here shows that with such a behavior of the error bound contributions the heuristic gives indeed the optimal distribution of the truncation parameters.

Finally, we compare RR with SR for the performability model. Figure 8 gives the transient behavior of $\overline{n}(t)$ for $\nu = 0.2, 0.4, 0.6$ and, for each ν, the optimum M, taking as initial state the state with no faulty servers and no tasks in the buffer. If we imagine the fault-tolerant server as a component which can be dynamically created at the application

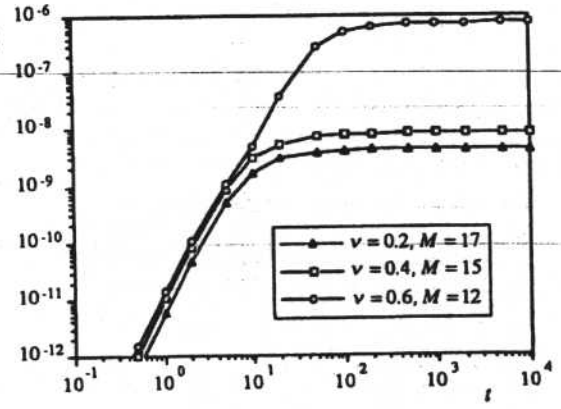

Figure 8: $\overline{n}(t)$ of the fault-tolerant server for several values of ν,and for each ν optimum M.

level, this is the state in which a fault-tolerant server would be created. Table 5 compares SR with RR for the three values of ν considered and $\epsilon = 10^{-5}$. Again, RR requires less steps than SR and significantly less for large Λt ($2 < \Lambda < 3$ in all three cases). The degradation in the performance of RR when ν increases can be easily explained by considering that the mean recurrence time to u increases significantly with ν, while Λ has little change, resulting in an increase in the average number of steps to absorption of the DTMC Z with the corresponding slower decay rate in $a(k)$.

5 Conclusions

A new randomization method called *regenerative randomization* (RR) has been proposed and applied to the computation of two transient measures of rewarded CTMC's: the transient reward rate, $TRR(t)$, and the mean reward rate in $[0, t]$, $MRR(t)$. As described here, the method can be used to compute the transient solution of irreducible finite CTMC's. We have proved that in the common case in which the initial distribution of the model is concentrated in a single state, RR requires at most the same number of steps as standard randomization (SR). Furthermore, in contrast with SR, RR has a benign behavior as Λt increases. Using numerical examples it has been shown that RR can be orders of magnitude faster than SR, specially in dependability and performability models which often have "rare" states and, at the same time, frequently visited states. We have observed numerical unstability, apparently due to cancellation errors, when RR requires many steps and its performance is almost identical to SR. The solution of this problem is a subject of future research.

Acknowledgements

The authors are indebted to José L. Domingo for helping them to prepare the numerical examples and to Albert de Marco for its help in programming the regenerative randomization method.

References

[Car94] J. A. Carrasco and A. Calderón, "Regenerative Randomization: Theory and applicaton examples," *Technical Report*, UPC, 1994.

Table 5: Number of steps in SR and RR to estimate $\overline{n}(t)$ of the fault-tolerant server with $\epsilon = 10^{-5}$ for $C = 20$, $\psi = 1$, $\delta = 0.2$, $\lambda_t = 2 \times 10^{-4}$, $\lambda_p = 10^{-5}$, $\mu_d = 0.5$, $\mu_r = 0.5$, several values of ν, and for each ν optimum M.

t	$\overline{n}(t)$	steps	
		SR	RR
$\nu = 0.2$, $M = 17$			
0.1	1.218×10^{-15}	14	12
1	5.752×10^{-12}	24	21
10	1.689×10^{-9}	73	59
100	4.109×10^{-9}	393	215
10^3	4.306×10^{-9}	3,063	255
10^4	4.392×10^{-9}	28,105	275
$\nu = 0.4$, $M = 15$			
0.1	2.412×10^{-15}	14	13
1	1.044×10^{-11}	25	23
10	2.942×10^{-9}	75	68
100	7.724×10^{-9}	415	349
10^3	8.313×10^{-9}	3,271	726
10^4	8.372×10^{-9}	30,129	789
$\nu = 0.6$, $M = 12$			
0.1	3.582×10^{-15}	14	13
1	1.424×10^{-11}	26	24
10	4.785×10^{-9}	78	73
100	4.930×10^{-7}	428	397
10^3	7.216×10^{-7}	3,445	1,773
10^4	7.445×10^{-9}	32,045	2,019

[Cin75] E. Çinlar, *Introduction to Stochastic Processes*, Prentice-Hall, 1975, pp. 371–378.

[Cru76] K. S. Crump, "Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation," *Journal of the ACM*, vol. 23, pp. 89–96, 1976.

[Dij90] N. M. Dijk, "On a Simple Proof of Uniformization for Continuous and Discrete-State Continuous-Time Markov Chains," *Adv. Appl. Prob.*, vol. 22, pp. 749–750, 1990.

[Gra77] W. K. Grassmann, "Transient solutions in Markovian queuing systems," *Comput. Operations Res.*, vol. 4, pp. 47–53, 1977.

[Gro84] D. Gross and D. R. Miller, "The randomization technique as a modelling tool and solution procedure for transient Markov processes," *Operations Res.*, vol. 32, pp. 343–361, 1984.

[Jen53] A. Jensen, "Markoff chains as an aid in the study of Markoff processes," *Skand. Akuarietidskrift*, vol. 36, pp. 87–91, 1953.

[Koh82] J. Kohlas, *Stochastic Methods of Operations Research*, Cambridge University Press, Cambridge, 1982.

[Mal94] M. Malhotra, J. K. Muppala and K. S. Trivedi, "Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains," *Microelectron. Reliab.*, vol. 34, no. 11, pp. 1825–1841, 1994.

[Moo93] A. P. Moorsel and W. H. Sanders, "Adaptive Uniformization", Technical report, University of Arizona, 1993.

[Qur93] M. A. Qureshi and W. H. Sanders, "Reward Model Solution Methods with Impulse and Rate Rewards: An Algorithm and Numerical Results," Technical Report, University of Arizona, 1993.

[Rei88] A. Reibman and K. S. Trivedi, "Numerical Transient Analysis of Markov Models," *Comput. Operations Res.*, vol. 15, pp. 19–36, 1988.

[Rub93] G. Rubino and B. Sericola, "Interval Availability Distribution Computation," in *Proc. 23th Int. Symp. on Fault-Tolerant Computing FTCS-23*, Toulouse, pp. 48–55, June 1993.

[Sou86] E. de Souza e Silva and H. R. Gail, "Calculating Cumulative Operational Time Distributions of Repairable Computer Systems," *IEEE Trans. on Computers*, vol. 35, pp. 322–332, 1986.

[Sou89] 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, pp. 179–199, 1989.