Solving Dependability/Performability Irreducible Markov Models Using Regenerative Randomization

Juan A. Carrasco, *Senior Member, IEEE*

***Abstract*—Markov models are commonly used to asses the dependability/performability of fault-tolerant systems. Computation of many dependability/performability measures for repairable fault-tolerant systems requires the transient analysis of irreducible Markov models. Examples of such measures are the unavailability at time t and the s-expected interval unavailability at time t. Randomization (also called uniformization) is a well-known Markov transient analysis method and has good properties: numerical stability, well-controlled computation error, and ability to specify the computation error in advance. However, the randomization method is computationally expensive when the model is stiff, as is the case for Markov models of repairable fault-tolerant systems when the mission time of interest is large. Steady-state detection is a technique recently proposed to speedup randomization when the model is irreducible. This paper points out that another method, regenerative randomization, which has the same good properties as randomization, also covers irreducible models, and compares, for the important class of irreducible failure/repair models with exponential failure and repair time distributions and repair in every state with failed components, the efficiency of the regenerative randomization method with that of randomization with steady-state detection. In the frequent case in which the initial state is the state without failed components the regenerative randomization method can be faster than randomization with steady-state detection, specially when the model is large and the failure rates are much smaller than the repair rates. For other initial probability distributions, the regenerative randomization method seems to perform worse than randomization with steady-state detection.**

***Index Terms*—Irreducible markov models, randomization, repairable fault-tolerant systems, transient analysis.**

ACRONYMS[1]

CTMC	homogeneous continuous-time Markov chain
DTMC	homogeneous discrete-time Markov chain
ODE	ordinary differential equation
pmf	probability mass function
RSD	randomization with steady-state detection as proposed in [27]
RR	regenerative randomization
hr	hour(s)
sec	second(s)
Sf	Survivor function

Manuscript received January 18, 2000; revised July1, October 10, 2000 and May 20, July 29, 2001. This work was supported by the "Comisión Interministerial de Ciencia y Tecnología" (CICYT) of the Ministry of Science and Technology of Spain under the research grant TAP99-0443-C05-05. Responsible Editor: J. W. Rupe.

J. A. Carrasco is with Dept. d'Enginyeria Electrònica, UPC, Diagonal 647 plta. 9, 08028 Barcelona, SPAIN (e-mail: Carrasco@eel.upc.es).

Digital Object Identifier 10.1109/TR.2003.818715

NOTATION

X	CTMC modeling the system
S	finite state space of X
r	regenerative state of X
S'	$S-\{r\}$ (assumed to be nonempty)
α_i	$\Pr\{X(0)=i\}$
$\boldsymbol{\alpha}$	row vector $(\alpha_i)_{i\in S}$
α_B	$\sum_{i\in B}\alpha_i$
$\lambda_{i,j}$	transition rate of X from state i to state j, $i\neq j$
λ_i	$\sum_{j\in S-\{i\}}\lambda_{i,j}$: output rate of X from state i
Λ	randomization rate
r_i	reward rate associated with state i of $X(r_i\geq 0)$
$r_{\max}$	$\max_{i\in S}[r_i]$
Q	Poisson process with arrival rate Λ
$\widehat{X}$	DTMC obtained by randomizing X with rate Λ
$\widehat{X}'$	version of $\widehat{X}$ in which the initial probability distribution is concentrated in state r
$P_{i,j}$	jump probability of $\widehat{X}$ from state i to state j
$\mathbf{P}$	$(P_{i,j})_{i,j\in S}$: transition probability matrix of $\widehat{X}$
$P_{i,B}$	$\sum_{j\in B}P_{i,j}$
$\mathbf{q}(k)$	$(\Pr\{\widehat{X}_k=i\})_{i\in S}$: probability row vector of $\widehat{X}$ at step k
ε	allowed error for computing $\mathrm{ETRR}(t)$ and $\mathrm{EARR}(t)$
$\mathrm{ETRR}(t)$	s-expected transient reward of X at time t
$\mathrm{ETRR}(t)$	s-expected average reward of X at time t
$\mathrm{ETRR}_N^a(t)$	approximation for $\mathrm{ETRR}(t)$ computed by standard randomization
$\mathrm{ETRR}_N^a(t)$	approximation for $\mathrm{EARR}(t)$ computed by standard randomization
$\mathrm{ETRR}^{K,L,a(t),}$, $\mathrm{EARR}^{K,a(t)}$	approximation for $\mathrm{ETRR}(t)$ given by the truncated transformed model used in regenerative randomization for, respectively, $\alpha_{S'}>0$ and $\alpha_{S'}=0$
$\mathrm{ETRR}^{K,L,a(t)}$, $\mathrm{EARR}^{K,a(t)}$	approximation for $\mathrm{EARR}(t)$ given by the truncated transformed model used in regenerative randomization for, respectively, $\alpha_{S'}>0$ and $\alpha_{S\prime}=0$
$\mathrm{poim}(i;\mu)$	$\exp(-\mu)\cdot\mu_i/i!$: Poisson pmf
$\mathrm{poim}(i;\mu)$	$\sum_{j=i}^{\infty}\mathrm{poim}(j;\mu)$: Poisson Sf

[1] The singular and plural of an acronym are always spelled the same.

$Y_{m:n}C$ (for a DTMC $Y = \{Y_k, k = 0, 1, 2, \ldots\}$) predicate which is true when Y_k satisfies condition c for all k, $m \leq k \leq n$ (by convention, the predicate is true for $m > n$)

$\#(Y_{m:n}C)$ (for a DTMC $Y = \{Y_k, k = 0, 1, 2, \ldots\}$) number of indexes k, $m \leq k \leq n$, for which Y_k satisfies condition c

Z, Z' DTMC derived from $\widehat{X}$ used in RR

$\mathbf{P}_Z$ transition probability matrix of Z restricted to S

$\mathbf{P}_{Z'}$ transition probability matrix of Z' restricted to S'

$\pi_i(k)$ $\Pr\{Z_k = i\}$

$\boldsymbol{\pi}(k)$ row vector $(\pi_i(k))_{i \in S}$

$\pi'_i(k)$ $\Pr\{Z'_k = i\}$

$\boldsymbol{\pi}'(k)$ row vector $(\pi'_i(k))_{i \in S'}$

V_K, $V_{K,L}$ truncated transformed CTMC

$\widehat{V}_K$, $\widehat{V}_{K,L}$ DTMC obtained by randomizing, respectively, V_K and $V_{K,L}$ with rate Λ

I_c indicator function: 1 if condition c is satisfied, and 0 otherwise

I. Introduction

CTMC are frequently used to asses the dependability/performability of fault-tolerant systems. Due to model stiffness, the transient analysis of these models can be appreciably more costly than the steady-state analysis, and very costly in absolute terms when the CTMC is large. This makes the development of efficient transient analysis techniques for CTMC dependability/performability models a research topic of great interest. Commonly used methods are ODE solvers and randomization (also called uniformization). Good reviews of these methods with new results are in [15], [16], [23]. The randomization method (also called uniformization) is attractive because of its excellent numerical stability and because the computation error is well-controlled and can be specified in advance[2]. It was first proposed in [10] and then was further developed in [12]. The method is also offered by well-known performance, dependability, and performability modeling packages [2], [6], [7], [9]. The randomization method is based on the following result [13 theorem 4.19]. Let $X = \{X(t);\ t \geq 0\}$ be a CTMC with finite state space S; consider any $\Lambda \geq \max_{i \in S} \lambda_i$, and define the DTMC $\widehat{X} = \{\widehat{X}_k;\ k = 0, 1, 2, \ldots\}$ with the same state space and initial probability distribution as X, and transition probabilities

$$P[\widehat{X}_{k+1} = j | \widehat{X}_k = i] = P_{i,j} = \frac{\lambda_{i,j}}{\Lambda}, i \neq j,$$
$$P[\widehat{X}_{k+1} = i | \widehat{X}_k = i] = P_{i,i} = 1 - \frac{\lambda_i}{\Lambda}.$$

Let $Q = \{Q(t);\ t \geq 0\}$ be a Poisson process with arrival rate Λ s-independent of $\widehat{X}$. We have

$$\Pr\{Q(t) = k\} = \operatorname{poim}(k; \Lambda \cdot t).$$

Then, $X = \{X(t);\ t \geq 0\}$ is probabilistically identical to $\{\widehat{X}_{Q(t)}; t \geq 0\}$; this is called the "randomization result".

The DTMC $\widehat{X}$ is called the randomized DTMC of X with rate Λ.

The CTMC X is called the derandomized CTMC of $\widehat{X}$ with rate Λ.

The randomization result immediately gives a scheme for computing the transient probabilities of X, but it can also be used to compute more complex measures [21], [22], [25], [26], [28]–[30]. This paper considers finite irreducible rewarded CTMC models X with a reward rate structure $r_i \geq 0$, $i \in S$. The reward rate of X at time t is the r.v. $r_{X(t)}$, $t \geq 0$. Its behavior can be summarized using several measures. Two such measures considered here are:

$$\operatorname{ETRR}(t) = E[r_{X(t)}] = \sum_{i \in S} r_i \cdot \Pr\{X(t) = i\};$$
$$\operatorname{EARR}(t) = E\left[\frac{\int_0^t r_{X(\tau)} d\tau}{t}\right].$$

Note that

$$\operatorname{EARR}(t) = \frac{1}{t} \cdot \int_0^t \operatorname{ETRR}(\tau) d\tau. \tag{1}$$

Important and useful dependability/performability measures for fault-tolerant systems are particular cases of $\operatorname{ETRR}(t)$ and $\operatorname{EARR}(t)$. Thus, taking $r_i = 1, i \in D$ and $r_i = 0$, $i \in U$, where D is the subset of states of X in which the system is down, and U is the subset of states of X in which the system is up, $\operatorname{ETRR}(t)$ becomes the unavailability of the system at time t and $\operatorname{EARR}(t)$ becomes the s-expected interval unavailability at time t (s-expected value of the fraction of time that the system is down in $[0, t]$). Also, if r_i is the performance rate (*e.g.*, the rate at which tasks are served by a multiprocessor system subject to component failures and repairs) when the system is in state i, then $\operatorname{ETRR}(t)$ becomes the s-expected performance rate of the system at time t, and $\operatorname{EARR}(t)$ becomes the s-expected averaged performance rate of the system during $[0, t]$.

A review of the (standard) randomization method for computing $\operatorname{ETRR}(t)$ and $\operatorname{EARR}(t)$ follows. The review is convenient because RR uses standard randomization. To have maximum efficiency, let $\Lambda \equiv \max_{i \in S} [\lambda_i]$ in standard randomization. Using the randomization result, $\operatorname{ETRR}(t)$ can be expressed in terms of the transient regime of $\widehat{X}$ as

$$\operatorname{ETRR}(t) = \sum_{i \in S} r_i \cdot \left(\sum_{k=0}^{\infty} \Pr\{\widehat{X}_k = i\} \cdot \Pr\{Q(t) = k\} \right)$$
$$= \sum_{k=0}^{\infty} d(k) \cdot \operatorname{poim}(k; \Lambda \cdot t)$$
$$d(k) \equiv \sum_{i \in S} r_i \Pr\{\widehat{X}_k = i\}. \tag{2}$$

[2] The computation error has 2 components: truncation error and round-off error; the truncation error can be made arbitrarily small, the round-off error has a very small relative value due to the numerical stability of the method if double precision is used. Rigorous bounds for the round-off errors have been obtained [11] under certain conditions concerning the values that transition rates can have and assuming a special method for computing Poisson probabilities.

The $\mathbf{q}(k)$, $k > 0$ can be obtained from $\mathbf{q}(0)$ using

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

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

$$\mathrm{ETRR}_N^a(t) = \sum_{k=0}^{N} d(k) \cdot \mathrm{poim}(k; \Lambda \cdot t),$$

and, taking into account that $d(k) \leq r_{\max} = \max_{i \in S}[r_i]$, the error is upper bounded using

$$\mathrm{ETRR}(t) - \mathrm{ETRR}_N^a(t) \leq r_{\max} \cdot \mathrm{poifc}(N+1; \Lambda \cdot t).$$

A common accuracy requirement is to limit the error in $\mathrm{ETRR}(t)$ to a value $\leq \varepsilon$; then, N is chosen as

$$N = \min\left[m \geq 0 : r_{\max} \cdot \mathrm{poifc}(m+1; \Lambda \cdot t) \leq \varepsilon\right].$$

For $\mathrm{EARR}(t)$, using (1) and (2), the approximate value is [5]:

$$\mathrm{EARR}_N^a(t) = \frac{1}{\Lambda \cdot t} \cdot \sum_{k=1}^{N+1} \mathrm{poim}(k; \Lambda \cdot t) \cdot \sum_{l=0}^{k-1} d(l),$$

with error upper bounded [5] as

$$\mathrm{EARR}(t) - \mathrm{EARR}_N^a(t) \leq r_{\max} \cdot \mathrm{poifc}(N+1; \Lambda \cdot t),$$

and, limiting the error in EARR to a value $\leq \varepsilon$, N can be chosen as

$$N = \min\left[m \geq 0 : r_{\max} \cdot \mathrm{poifc}(m+1; \Lambda \cdot t) \leq \varepsilon\right].$$

Stable and efficient computation of poim $(k; \Lambda \cdot t)$ while avoiding overflows and intermediate underflows is a delicate issue, and several alternatives have been proposed [3], [8], [14], [20]. The implementations, in this paper, of both RR and RSD use the method in [14: pp 1028–1029] (see also [1]) which has good numerical stability.

For large models, the computational cost of the randomization method is roughly due to the vector-matrix multiplications (3). Using the well-known result [24] that $Q(t)$ has for $\Lambda \cdot t \to \infty$ an asymptotic Gaussian distribution with mean and variance $\Lambda \cdot t$, it is easy to see that for large $\Lambda \cdot t$ and $\varepsilon \ll 1$ the required $N \approx \Lambda \cdot t$. To solve the model for values of t for which $\Lambda \cdot t$ is large, the standard randomization method is highly inefficient. For instance, consider, a CTMC model of a fault-tolerant system with hot restarts having an exponential duration with mean 1 minute so that the required Λ is of the order of 1/min. For $t = 1$ year, $\Lambda \cdot t \approx 525{,}600$, making standard randomization very inefficient if the model is large.

Several variants of the standard randomization method have been proposed to improve its efficiency. Miller [18] has used selective randomization to solve reliability models with detailed representation of error handling activities. The idea behind selective randomization [17] is to randomize the model only in a subset of states. [23] proposes an approach based on the multi-step concept: compute $\mathbf{P}^M$ explicitly, where M is the length of the multi-step, and use the recurrence $\mathbf{q}(k+M) = \mathbf{q}(k) \times \mathbf{P}^M$ to advance $\widehat{X}$ faster for steps which have negligible contributions to the transient solution of X. Since the number of $\mathbf{q}(k)$'s with important contributions is of the order of $\sqrt{\Lambda \cdot t}$, the multi-step concept allows an appreciable reduction of the required number of vector-matrix multiplications. However, when computing $\mathbf{P}^M$, appreciable fill-in can occur if $\mathbf{P}$ is sparse. Adaptive uniformization [19] is a recent method in which the randomization rate is adapted, depending on the states in which the randomized DTMC can be at a given step. Numerical experiments have shown that adaptive uniformization can be appreciably faster than standard randomization for short-to-medium mission times. In addition, it can be used to solve models with infinite state spaces and not uniformly-bounded output rates. Recently, the combination of adaptive and standard randomization has been proposed to obtain a method which outperforms both adaptive uniformization and standard randomization for most models [20]. Another recent proposal to speed up the standard randomization method for irreducible models is steady-state detection [15]. Recently, RSD, based on steady-state detection, which gives error bounds, has been developed [27]. RSD has the same good properties as standard randomization. For short mission times, RSD performs about as well as standard randomization, but for long mission times (once X has reached steady-state) RSD outperforms standard randomization. Another recent proposal is the RR method [4], [5], which covers CTMC models with finite state space $S \cup \{f_1, f_2, \ldots, f_A\}$, $|S| \geq 2$, $A \geq 0$, where f_i are absorbing states, and, either a) all states in S are transient, or b) S has a single trapping component and the chosen regenerative state $r \in S$ belongs to that component[3], such that all states are reachable from some state with nonnull initial probability. The method also assumes that the CTMC model has some transition from r to $S - \{r\}$, although that condition can be easily circumvented in practice [4], [5]. RR can be appreciably faster than standard randomization for large models and long mission times.

RR covers finite irreducible CTMC models: it suffices to consider the case in which $A = 0$ and S has a single trapping component. Then, it is meaningful to compare the performances for irreducible models of RR and RSD.

Section II reviews the RR method for irreducible models. Section III compares RR and RSD for the important class of irreducible failure/repair models with exponential failure and repair time distributions and repair in every state with failed components.

II. The RR Method for Irreducible Models

The RR method requires selecting a state $r \in S$ as the "regenerative" state. The basic idea in RR is to obtain a truncated transformed model of potentially smaller size than the original model X, by characterizing with enough accuracy the behavior of X up to state r and from r until the next hit of r and, then, solve

[3] Two states of a CTMC i, j 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 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.

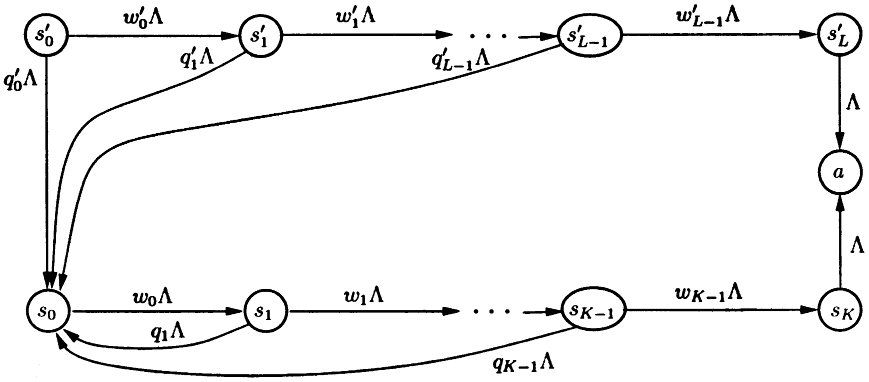

Fig. 1. State transition diagram of the CTMC $V_{K,L}$.

the truncated transformed model, which yields with some arbitrarily small error the same measures $\mathrm{ETRR}(t)$ and $\mathrm{EARR}(t)$ as the original model, by standard randomization.

In the RR method, two transient DTMC are considered [4], [5]:

1) $Z = \{Z_k; k = 0, 1, 2, \ldots\}$, formally defined as

$$\begin{aligned} Z_0 =& r, \\ Z_k =& \begin{cases} i \in S' & \text{if } \widehat{X}'_{1:k} \neq r \wedge \widehat{X}'_k = i, \\ a & \text{if } \hat{X}'_{1:k} = r > 0, \end{cases} \qquad k > 0. \end{aligned}$$

Z has state space $S \cup \{a\}$, where a is an absorbing state and all states in S are transient, and its (possibly) nonnull transition probabilities are

$$\begin{aligned} \Pr\{Z_{k+1} =& j | Z_k = i\} = P_{i,j}, i \in S, j \in S', \\ \Pr\{Z_{k+1} =& a | Z_k = i\} = P_{i,r}, i \in S, \\ \Pr\{Z_{k+1} =& a | Z_k = a\} = 1. \end{aligned}$$

2) $Z' = \{Z'_k; k = 0, 1, 2, \ldots\}$, formally defined as

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

Z' has state space $S' \cup \{a\}$, where a is an absorbing state and all states in S' are transient. The initial probability distribution of Z' is $\Pr\{Z'_0 = i\} = \alpha_i$, $i \in S'$, $\Pr\{Z'_0 = a\} = \alpha_r$, and its (possibly) nonnull transition probabilities are

$$\begin{aligned} \Pr\{Z'_{k+1} =& j | Z'_k = i\} = P_{i,j}, i, j \in S', \\ \Pr\{Z'_{k+1} =& a | Z'_k = i\} = P_{i,r}, i \in S', \\ \Pr\{Z'_{k+1} =& a | Z'_k = a\} = 1. \end{aligned}$$

Row vector $\boldsymbol{\pi}(0)$ has components $\pi_r(0) = 1, \pi_i(0) = 0, i \in S'$. From $\boldsymbol{\pi}(0)$, the $\boldsymbol{\pi}(k)$, $k > 0$ can be obtained using

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

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

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

In RR, Λ is taken slightly larger than $\max_{i \in S}[\lambda_i]$, *i.e.*, $\Lambda = (1+\theta) \cdot \max_{i \in S}[\lambda_i]$, where θ is small *e.g.*, 10^{-4}. This guarantees that the $a(k)$ to be defined next are > 0 and that, if $\alpha_{S'} > 0$, then the $a'(k)$ to be defined next are > 0.

Let

$$\begin{aligned} a(k) =& \sum_{i \in S} \pi_i(k), \\ q_k =& \frac{1}{a(k)} \cdot \sum_{i \in S} \pi_i(k) \cdot P_{i,r}, \\ w_k =& \frac{1}{a(k)} \cdot \sum_{i \in S} \pi_i(k) \cdot P_{i,S'}, \\ b(k) =& \frac{1}{a(k)} \cdot \sum_{i \in S} r_i \cdot \pi_i(k); \end{aligned}$$

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

$$\begin{aligned} a'(k) =& \sum_{i \in S'} \pi'_i(k), \\ q'_k =& \frac{1}{a'(k)} \cdot \sum_{i \in S'} \pi'_i(k) \cdot P_{i,r}, \\ w'_k =& \frac{1}{a'(k)} \cdot \sum_{i \in S'} \pi'_i(k) \cdot P_{i,S'}, \\ b'(k) =& \frac{1}{a'(k)} \cdot \sum_{i \in S'} r_i \cdot \pi'_i(k). \end{aligned}$$

For $\alpha_{S'} > 0$, let $V_{K,L} = \{V_{K,L}(t),\ t \geq 0\}$, $K \geq 1$, $L \geq 1$, be the CTMC with state space $\{s_k;\ 0 \leq k \leq K\} \cup \{S'_k;\ \ 0 \leq k \leq L\} \cup \{a\}$, initial probability distribution

$$\begin{aligned} \Pr\{V_{K,L}(0) = s_0\} =& \alpha_r, \\ \Pr\{V_{K,L}(0) = S'_0\} =& \alpha_{S'}, \\ \Pr\{V_{K,L}(0) = i\} =& 0, \quad i \notin \{s_0, S'_0\} \end{aligned}$$

and the state transition diagram of Fig. 1. CTMC $V_{K,L}$ is the truncated transformed model for $\alpha_{S'} > 0$. Let $\mathrm{ETRR}^{K,L,a}(t)$ be the s-expected transient reward rate of $V_{K,L}$ at time t with reward rate structure $r'_{s_k} = b(k)$, $r'_{S'_k} = b'(k)$, and $r'_a = 0$. Then, $\mathrm{ETRR}^{K,L,a}(t)$ can be taken as an approximation for $\mathrm{ETRR}(t)$ with error upper bounded as

$$\begin{aligned} \mathrm{ETRR}(t) - \mathrm{ETRR}^{K,L,a}(t) \leq& r_{\max} \cdot a'(L) \\ & \cdot \mathrm{poifc}(L+1; \Lambda \cdot t) \\ & + r_{\max} \cdot a(K) \cdot \sum_{k=K+1}^{\infty} (k-K) \\ & \cdot \mathrm{poim}(k; \Lambda \cdot t). \end{aligned} \tag{4}$$

```
**Inputs**: $X$, $r_i$, $i \in S$, $\boldsymbol{\alpha}$, $r$, $\varepsilon$, $n$, $t_1, t_2, \ldots, t_n$
**Outputs**: $\widetilde{ETRR}(t_1), \widetilde{ETRR}(t_2), \ldots, \widetilde{ETRR}(t_n)$
$r_{\max} = \max_{i \in S} r_i$;
$t_{\max} = \max\{t_1, t_2, \ldots, t_n\}$;
$\Lambda = (1 + 10^{-4}) \max_{i \in S} \lambda_i$;
Obtain $\mathbf{P}$;
for $(i \in S)$ $P_{i,S'} = \sum_{j \in S', P_{i,j} \neq 0} P_{i,j}$; $\alpha_{S'} = \sum_{i \in S'} \alpha_i$;
if $(\alpha_{S'} > 0)$ $tol\_K = \varepsilon/4$; else $tol\_K = \varepsilon/2$;
$\boldsymbol{\pi} = (I_{i=r})_{i \in S}$; $a = 1$; $K = 0$;
do {
    $q_K = \sum_{i \in S, P_{i,r} \neq 0} \pi_i P_{i,r}/a$; $w_K = \sum_{i \in S} \pi_i P_{i,S'}/a$; $b(K) = \sum_{i \in S} \pi_i r_i/a$;
    $\boldsymbol{n\pi} = \boldsymbol{\pi}\mathbf{P}_Z$; $\boldsymbol{\pi} = \boldsymbol{n\pi}$;
    $K{+}{+}$;
    $a = \sum_{i \in S} \pi_i$;
}
until $(r_{\max} a \sum_{k=K+1}^{\infty} (k - K)\, \mathrm{poim}(k; \Lambda t_{\max}) \leq tol\_K)$;
$b(K) = \sum_{i \in S} \pi_i r_i/a$;
if $(\alpha_{S'} > 0)${
    $\boldsymbol{\pi}' = (\alpha_i)_{i \in S'}$; $a' = \alpha_{S'}$; $L = 0$;
    do {
        $q'_L = \sum_{i \in S', P_{i,r} \neq 0} \pi'_i P_{i,r}/a'$; $w'_L = \sum_{i \in S'} \pi'_i P_{i,S'}/a'$; $b'(L) = \sum_{i \in S'} \pi'_i r_i/a'$;
        $\boldsymbol{n\pi}' = \boldsymbol{\pi}'\mathbf{P}_{Z'}$; $\boldsymbol{\pi}' = \boldsymbol{n\pi}'$;
        $L{+}{+}$;
        $a' = \sum_{i \in S'} \pi'_i$;
    }
    until $(r_{\max} a' \sum_{k=L+1}^{\infty} \mathrm{poim}(k; \Lambda t_{\max}) \leq \varepsilon/4)$;
    $b'(L) = \sum_{i \in S'} \pi'_i r_i/a'$;
}
$N = \min\{m \geq 0 : r_{\max} \sum_{k=m+1}^{\infty} \mathrm{poim}(k; \Lambda t_{\max}) \leq \varepsilon/2\}$;
if $(\alpha_{S'} > 0)$
    Give $N$ steps to $\widehat{V}_{K,L}$ and compute $d(k) = \sum_{l=0}^{K} b(l) \Pr\{(\widehat{V}_{K,L})_k = s_l\}$
    $+ \sum_{l=0}^{L} b'(l) \Pr\{(\widehat{V}_{K,L})_k = s'_l\}$, $k = 0, 1, \ldots, N$;
else
    Give $N$ steps to $\widehat{V}_K$ and compute $d(k) = \sum_{l=0}^{K} b(l) \Pr\{(\widehat{V}_K)_k = s_l\}$,
    $k = 0, 1, \ldots, N$;
for $(i = 1;\ i \leq n;\ i{+}{+})$
    for $(k = 0,\ \widetilde{ETRR}(t_i) = 0;\ k \leq N;\ k{+}{+})$ $\widetilde{ETRR}(t_i)$ += $d(k)\, \mathrm{poim}(k; \Lambda t_i)$;
```

Fig. 2. C-like algorithmic description of RR for $\mathrm{ETRR}(t)$.

Let $\mathrm{EARR}^{K,L,a}(t)$ be the s-expected averaged reward rate of $V_{K,L}$ at time t with the same reward rate structure as before. Then, $\mathrm{EARR}^{K,L,a}(t)$ can be taken as an approximation for $\mathrm{EARR}(t)$ with error upper bounded by

$$\mathrm{EARR}(t) - \mathrm{EARR}^{K,L,a}(t) \leq \frac{r_{\max} \cdot a'(L)}{\Lambda \cdot t} \cdot \sum_{k=L+2}^{\infty} (k - L - 1) \cdot \mathrm{poim}(k; \Lambda \cdot t) + \frac{r_{\max} \cdot a(K)}{\Lambda \cdot t} \cdot \sum_{k=K+2}^{\infty} \frac{(k-K) \cdot (k-K-1)}{2} \cdot \mathrm{poim}(k; \Lambda \cdot t). \quad (5)$$

For $\alpha_{S'} = 0$, let $V_K = \{V_K(t);\ t \geq 0\}$, $K \geq 1$, be the CTMC differing from $V_{K,L}$ only in that the states S'_k are not present. CTMC V_K is the truncated transformed model for $\alpha_{S'} = 0$. Let $\mathrm{ETRR}^{K,a}(t)$ be the s-expected transient reward rate of V_K at time t; let

$\mathrm{EARR}^{K,a}(t)$ be the s-expected averaged reward rate of V_K at time t; both with reward rate structure $r'_{s_k} = b(k)$ and $r'_a = 0$. Then

$$\mathrm{ETRR}(t) - \mathrm{ETRR}^{K,a}(t) \leq r_{\max} \cdot a(K) \cdot \sum_{k=K+1}^{\infty} (k - K) \cdot \mathrm{poim}(k; \Lambda \cdot t); \quad (6)$$

$$\mathrm{EARR}(t) - \mathrm{EARR}^{K,a}(t) \leq \frac{r_{\max} \cdot a(K)}{\Lambda \cdot t} \cdot \sum_{k=K+2}^{\infty} \frac{(k-K) \cdot (k-K-1)}{2} \cdot \mathrm{poim}(k; \Lambda \cdot t). \quad (7)$$

In RR, the truncated transformed model $V_{K,L}(V_K)$ with its reward rate structure is obtained by stepping Z and, if $\alpha_{S'} > 0$,

```
**Inputs**: $X$, $r_i$, $i \in S$, $\boldsymbol{\alpha}$, $r$, $\varepsilon$, $n$, $t_1, t_2, \ldots, t_n$
**Outputs**: $\widetilde{EARR}(t_1), \widetilde{EARR}(t_2), \ldots, \widetilde{EARR}(t_n)$
$r_{\max} = \max_{i \in S} r_i$;
$t_{\max} = \max\{t_1, t_2, \ldots, t_n\}$;
$\Lambda = (1 + 10^{-4}) \max_{i \in S} \lambda_i$;
Obtain $\mathbf{P}$;
for $(i \in S)$ $P_{i,S'} = \sum_{j \in S', P_{i,j} \neq 0} P_{i,j}$; $\alpha_{S'} = \sum_{i \in S'} \alpha_i$;
if $(\alpha_{S'} > 0)$ $tol\_K = \varepsilon/4$; else $tol\_K = \varepsilon/2$;
$\boldsymbol{\pi} = (I_{i=r})_{i \in S}$; $a = 1$; $K = 0$;
do {
    $q_K = \sum_{i \in S, P_{i,r} \neq 0} \pi_i P_{i,r}/a$; $w_K = \sum_{i \in S} \pi_i P_{i,S'}/a$; $b(K) = \sum_{i \in S} \pi_i r_i/a$;
    $\boldsymbol{n\pi} = \boldsymbol{\pi}\mathbf{P}_Z$; $\boldsymbol{\pi} = \boldsymbol{n\pi}$;
    $K$++;
    $a = \sum_{i \in S} \pi_i$;
}
until $((r_{\max} a/(\Lambda t_{\max}))$
        $\sum_{k=K+2}^{\infty}((k-K)(k-K-1)/2)\,\mathrm{poim}(k; \Lambda t_{\max}) \leq tol\_K)$;
$b(K) = \sum_{i \in S} \pi_i r_i/a$;
if $(\alpha_{S'} > 0)${
    $\boldsymbol{\pi}' = (\alpha_i)_{i \in S'}$; $a' = \alpha_{S'}$; $L = 0$;
    do {
        $q'_L = \sum_{i \in S', P_{i,r} \neq 0} \pi'_i P_{i,r}/a'$; $w'_L = \sum_{i \in S'} \pi'_i P_{i,S'}/a'$; $b'(L) = \sum_{i \in S'} \pi'_i r_i/a'$;
        $\boldsymbol{n\pi}' = \boldsymbol{\pi}'\mathbf{P}_{Z'}$; $\boldsymbol{\pi}' = \boldsymbol{n\pi}'$;
        $L$++;
        $a' = \sum_{i \in S'} \pi'_i$;
    }
    until $((r_{\max} a'/(\Lambda t_{\max})) \sum_{k=L+2}^{\infty}(k-L-1)\,\mathrm{poim}(k; \Lambda t_{\max}) \leq \varepsilon/4)$;
    $b'(L) = \sum_{i \in S'} \pi'_i r_i/a'$;
}
$N = \min\{m \geq 0 : r_{\max} \sum_{k=m+1}^{\infty} \mathrm{poim}(k; \Lambda t_{\max}) \leq \varepsilon/2\}$;
if $(\alpha_{S'} > 0)$
    Give $N$ steps to $\widehat{V}_{K,L}$ and compute $d(k) = \sum_{l=0}^{K} b(l)\Pr\{(\widehat{V}_{K,L})_k = s_l\}$
    $+ \sum_{l=0}^{L} b'(l)\Pr\{(\widehat{V}_{K,L})_k = s'_l\}$, $k = 0, 1, \ldots, N$;
else
    Give $N$ steps to $\widehat{V}_K$ and compute $d(k) = \sum_{l=0}^{K} b(l)\Pr\{(\widehat{V}_K)_k = s_l\}$,
    $k = 0, 1, \ldots, N$;
for $(k = 1, c = 0; k \leq N + 1; k$++$)${
    $c$ += $d(k-1)$;
    $c(k) = c$;
}
for $(i = 1; i \leq n; i$++$)$
    for $(k = 1, \widetilde{EARR}(t_i) = 0; k \leq N + 1; k$++$)$ $\widetilde{EARR}(t_i)$ += $c(k)\,\mathrm{poim}(k; \Lambda t_i)/(\Lambda t_i)$;
```

Fig. 3. C-like algorithmic description of RR for $\mathrm{EARR}(t)$.

also stepping Z' until the model truncation error upper bounds (4)–(7) are $\leq \varepsilon/2$. An approximate value for the corresponding measure with error upper bounded by ε is, then, obtained by solving the truncated transformed model by standard randomization with truncation error $\leq \varepsilon/2$.

Algorithmic descriptions of RR (for irreducible models) in C-like syntax are given in Figs. 2 and 3 for, respectively, $\mathrm{ETRR}(t)$ and $\mathrm{EARR}(t)$. The inputs of the algorithms are:

- the CTMC X,
- the r_i, $i \in S$,
- the $\boldsymbol{\alpha} = (\alpha_i)_{i \in S}$,
- the regenerative state r,
- the allowed error ε,
- the number of time points n at which the measure has to be computed,
- the time points $t_1, t_2, \ldots, t_n$.

The algorithms have as outputs the estimates for the measure at the t_i. For $\alpha_{S'} > 0$, the K and L are independently obtained by making each term of the model truncation error upper bounds (4) and (5) $\leq \varepsilon/4$. Advantage is taken of the increasing

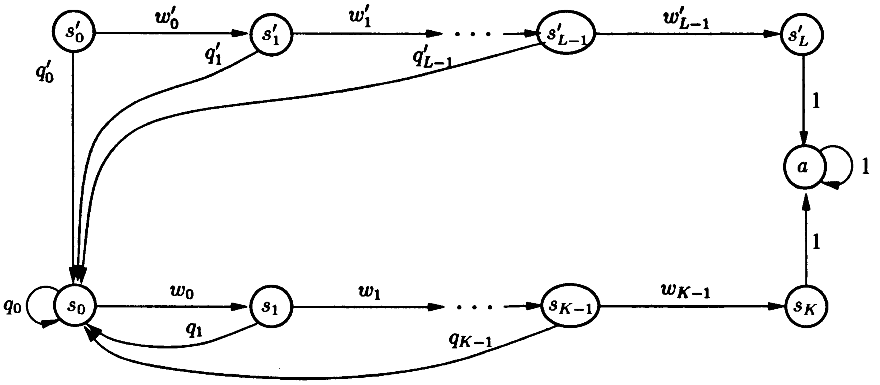

Fig. 4. State transition diagram of the DTMC $\widehat{V}_{K,L}$.

character with t of the error bounds [4], [5] and, then, the error bounds are controlled for $t = t_{\max} = \max\{t_1, t_2, \ldots, t_n\}$. Solution of the truncated transformed model by standard randomization involves stepping the randomized DTMC $\widehat{V}_{K,L}(\widehat{V}_K)$ of $V_{K,L}(V_K)$ with randomization rate Λ. Fig. 4 shows the state transition diagram of $\widehat{V}_{K,L}$. The state transition diagram of $\widehat{V}_K$ is identical to the state transition diagram of $\widehat{V}_{K,L}$ but without its upper part, corresponding to the states S'_k.

The algorithms in Figs. 2 and 3 require computing:

$$S(m) = \text{poifc}(m+1, \Lambda \cdot t),$$

$$S'(m) = \sum_{k=m+1}^{\infty} (k-m) \cdot \text{poim}(k, \Lambda \cdot t),$$

$$S''(m) = \sum_{k=m+2}^{\infty} \frac{(k-m) \cdot (k-m-1)}{2} \cdot \text{poim}(k, \Lambda \cdot t)$$

for $t = t_{\max}$ and for increasing values of m. Efficient and numerically stable procedures to perform these computations are in [4], [5].

As discussed in Section I, standard randomization requires a number of steps on $\widehat{X}$ which, for large $\Lambda \cdot t$ and $\varepsilon \ll 1$, is approximately equal to $\Lambda \cdot t$. Regarding RR, [4], [5] show that the number of required steps K on Z is $O(\log(\Lambda \cdot t/\varepsilon))$ and, if $\alpha_{S'} > 0$, the number of required steps L on Z' is $O(\log(1/\varepsilon))$. Thus, contrary to standard randomization, the number of steps required in regenerative randomization is, for large $\Lambda \cdot t$, a smooth function of t. That property is called 'benign behavior' and implies [4], [5] that, for large enough models and large enough $\Lambda \cdot t$, RR is appreciably faster than standard randomization.

III. Analysis and Comparison With RSD

This section compares the performances of RR and RSD for irreducible failure/repair models with exponential failure and repair time distributions and repair in every state with failed components. These models are the result of conceptualizing a fault-tolerant system as made up of components which can fail and be repaired with constant (possibly, state-dependent) rates. The CTMC X has only 'failure' and 'repair' transitions and there is a single state, o, without failed components. Every state in $S - \{o\}$ has some outgoing repair transition. When failure rates are much smaller than repair rates, a reasonable selection for the regenerative state for those models is $r = o$ [4], [5]. RR and RSD are compared for that selection.

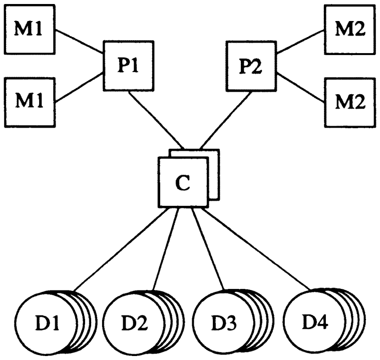

Fig. 5. Architecture of the parametric example for $N_{\text{DS}} = 4$.

To make that comparison, a parametric example will be considered (its architecture is depicted in Fig. 5 for $N_{\text{DS}} = 4$). The system consists of 2 processing subsystems, each including: 1 processor P and 2 memories M, 2 controllers, and N_{DS} sets of 4 disks each. The system is operational if at least:

- 1 processor and 1 memory connected to it are operational,
- 1 controller is operational, and
- 3 disks of each set are operational.

Components do not fail when the system is down. Processors fail with rate λ_P; a processor failure is soft with probability P_S and hard with probability $1 - P_S$. Memories fail with rate λ_M. Controllers fail with rate λ_C. Disks fail with rate λ_D. Failure of a controller is propagated to '2 disks of 1 set' with probability $1 - C_C$. The disk-set over which the failure is propagated is taken at random. Two people repair every processor in soft failure with rate μ_{PS}. The other repair actions are performed by another person, with priority given first 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 mode, μ_M for memories, μ_C for controllers, and μ_D for disks.

The comparison will be done using the sets of model parameter values in Table I. In sets B1 and B2, failure rates are larger

TABLE I
SETS (A1, A2, B1, B2) OF MODEL PARAMETER VALUES FOR THE EXAMPLE. ALL RATES ARE IN 'PER HOUR'

	A1	A2	B1	B2
λ_P	$2 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
λ_M	10^{-4}	10^{-4}	10^{-3}	10^{-3}
λ_C	$2 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
λ_D	10^{-5}	10^{-5}	10^{-4}	10^{-4}
P_S	0.8	0.8	0.8	0.8
C_C	0.99	0.99	0.99	0.99
μ_{PS}	0.5	10	0.5	10
μ_{PH}	0.2	0.2	0.2	0.2
μ_M	0.2	0.2	0.2	0.2
μ_C	0.5	0.5	0.5	0.5
μ_D	0.5	0.5	0.5	0.5

than in sets A1 and A2. In sets A1 and B1, repair rates are very similar; in sets A2 and B2 repair rates are more different.

The reward rate structure is

$$r_i = \begin{cases} 1 & \text{if } i \text{ is a `down' state} \\ 0 & \text{if } i \text{ is an `up' state} \end{cases}$$

This makes the ETRR(t) measure to be the unavailability of the system at time t, and the EARR(t) measure to be the s-expected interval unavailability of the system at time t.

2 values of N_{DS} will be considered: $N_{\text{DS}} = 4$ and $N_{\text{DS}} = 8$. The CTMC X has

- 8,096 states and 36 109 transitions for $N_{\text{DS}} = 4$,
- 162 304 states and 1 059 021 transitions for $N_{\text{DS}} = 8$.

The performance of RSD does not depend on the initial probability distribution of the model. However, the performance of RR does depend on the initial probability distribution of the model: when $\alpha_{S'} = 0$ (the initial probability distribution of the model is concentrated in the state without failed components) then RR only steps the DTMC Z; when $\alpha_{S'} > 0$ (there is some initial probability distribution outside the state without failed components) both Z and Z' have to be stepped, and RR tends to be more expensive.

To compare the methods in both scenarios consider 2 initial probability distributions:

#1. the initial state is the state without failed components;
#2. the initial state is the state in which 1 disk of the first set is failed.

For both RR and RSD a value $\varepsilon = 10^{-12}$ for the error control parameter will be considered. To give an idea of how dependable the considered systems are, for $N_{\text{DS}} = 4$, the steady-state unavailability is

- $8.2585 \cdot 10^{-7}$ for set A1,
- $8.2358 \cdot 10^{-7}$ for set A2,
- $1.0693 \cdot 10^{-5}$ for set B1,
- $1.0438 \cdot 10^{-5}$ for set B2.

The cost of RSD is roughly proportional to the number of steps. The cost of RR has 2 components:

1) cost of obtaining the truncated transformed model (roughly proportional to the number of steps on Z, Z'),
2) cost of the solution of the truncated transformed model by standard randomization.

TABLE II
NUMBER OF STEPS REQUIRED BY RR AND RSD FOR THE EXAMPLE WITH $N_{\text{DS}} = 4$ AND THE UNAVAILABILITY MEASURE WHEN THE INITIAL STATE IS THE STATE WITHOUT FAILED COMPONENTS

t	A1		A2		B1		B2	
(hr)	RR	RSD	RR	RSD	RR	RSD	RR	RSD
10	38	50	283	315	40	50	290	315
20	58	77	511	561	61	77	521	561
50	103	145	1,141	1,259	110	145	1,163	1,259
100	151	237	2,057	2,378	170	245	2,122	2,378
200	179	237	2,582	3,441	218	247	3,109	3,567
500	190	237	2,750	3,441	233	247	3,345	3,567
1,000	197	237	2,841	3,441	241	247	3,455	3,567
2,000	202	237	2,922	3,441	248	247	3,551	3,567
5,000	209	237	3,023	3,441	256	247	3,670	3,567
10,000	214	237	3,098	3,441	262	247	3,756	3,567

TABLE III
NUMBER OF STEPS REQUIRED BY RR AND RSD FOR THE EXAMPLE WITH $N_{\text{DS}} = 4$ AND THE UNAVAILABILITY MEASURE WHEN THE INITIAL STATE IS THE STATE IN WHICH 1 DISK OF THE FIRST SET IS FAILED

t	A1		A2		B1		B2	
(hr)	RR	RSD	RR	RSD	RR	RSD	RR	RSD
10	79	50	584	315	83	50	594	315
20	119	77	1,041	561	125	77	1,058	561
50	211	145	2,311	1,259	224	145	2,353	1,259
100	311	237	4,176	2,378	346	245	4,334	2,378
200	348	237	5,028	3,441	425	247	6,099	3,567
500	359	237	5,200	3,441	441	247	6,343	3,567
1,000	366	237	5,291	3,441	449	247	6,454	3,567
2,000	371	237	5,373	3,441	455	247	6,551	3,567
5,000	378	237	5,475	3,441	464	247	6,669	3,567
10,000	383	237	5,550	3,441	470	247	6,756	3,567

The importance of component 2 decreases as the size of X increases, and increases as the mission-time increases. In addition, the cost of a step in RR (to DTMC Z or Z') is generally larger than the cost of a step in RSD because

- a step in RSD basically involves a vector-matrix multiplication with matrix $\mathbf{P}$ and a scalar product of vectors of size $|S|$ with $2|S|$ floating point operations,
- a step of Z (see Figs. 2 and 3) involves the vector-matrix multiplication $\boldsymbol{\pi} \times \mathbf{P}_Z$ and about $5|S|$ floating point operations,
- a step of Z' (see Figs. 2 and 3) involves the vector-matrix multiplication $\boldsymbol{\pi}' \times \mathbf{P}_{Z'}$ and about $5(|S|-1)$ floating point operations.

How costly a step in RR is with respect to a step in RSD depends basically on the average number of nonzero entries per row of $\mathbf{P}$: the larger that number, the closer (relatively) the costs of the steps in both methods are. In general, as X gets larger, the average number of nonzero entries per row tends to increase and the costs of the steps in RR and RSD tend to equalize.

First, RR and RSD are compared in terms of the number of required steps (on Z and Z' for RR). This gives an idea on the relative performances of the methods for large enough models. The results obtained for the example with $N_{\text{DS}} = 4$ are

- in Table II for the unavailability measure when the initial state is the state without failed components,
- in Table III for the unavailability measure when the initial state is the state in which 1 disk of the first set is failed,

TABLE IV
NUMBER OF STEPS REQUIRED BY RR AND RSD FOR THE EXAMPLE, WITH $N_{\mathrm{DS}} = 4$ AND THE s-EXPECTED INTERVAL UNAVAILABILITY MEASURE WHEN THE INITIAL STATE IS THE STATE WITHOUT FAILED COMPONENTS

t	A1		A2		B1		B2	
(hr)	RR	RSD	RR	RSD	RR	RSD	RR	RSD
10	35	50	270	315	38	50	277	315
20	54	77	491	561	57	77	502	561
50	96	145	1,101	1,259	103	145	1,127	1,259
100	142	237	1,941	2,378	159	245	2,061	2,378
200	169	237	2,425	3,441	204	247	2,899	3,567
500	183	237	2,647	3,441	225	247	3,217	3,567
1,000	191	237	2,753	3,441	234	247	3,350	3,567
2,000	197	237	2,841	3,441	241	247	3,456	3,567
5,000	204	237	2,947	3,441	250	247	3,581	3,567
10,000	209	237	3,023	3,441	256	247	3,670	3,567

TABLE V
NUMBER OF STEPS REQUIRED BY RR AND RSD FOR THE EXAMPLE WITH $N_{\mathrm{DS}} = 4$, AND THE s-EXPECTED INTERVAL UNAVAILABILITY MEASURE WHEN THE INITIAL STATE IS THE STATE IN WHICH 1 DISK OF THE FIRST SET IS FAILED

t	A1		A2		B1		B2	
(hr)	RR	RSD	RR	RSD	RR	RSD	RR	RSD
10	73	50	560	315	78	50	569	315
20	111	77	1,002	561	117	77	1,022	561
50	197	145	2,235	1,259	210	145	2,283	1,259
100	287	237	3,965	2,378	322	245	4,179	2,378
200	331	237	4,781	3,441	403	247	5,746	3,567
500	350	237	5,069	3,441	430	247	6,174	3,567
1,000	359	237	5,190	3,441	440	247	6,329	3,567
2,000	365	237	5,286	3,441	449	247	6,446	3,567
5,000	373	237	5,396	3,441	457	247	6,576	3,567
10,000	378	237	5,474	3,441	464	247	6,667	3,567

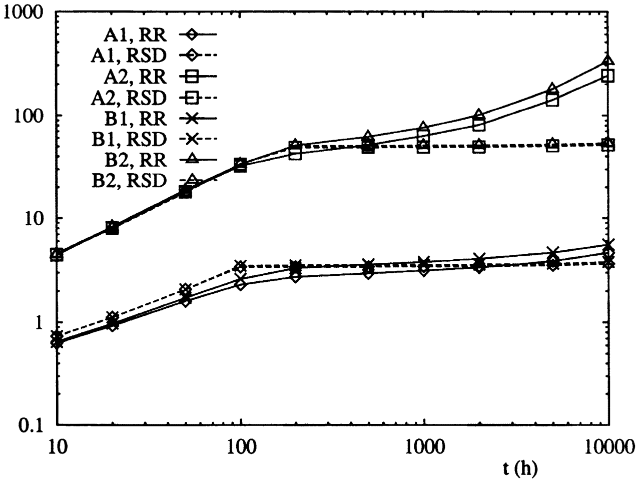

Fig. 6. CPU times (sec) required by RR and RSD for the example with N_{DS}=4 and the unavailability measure when the initial state is the state without failed components.

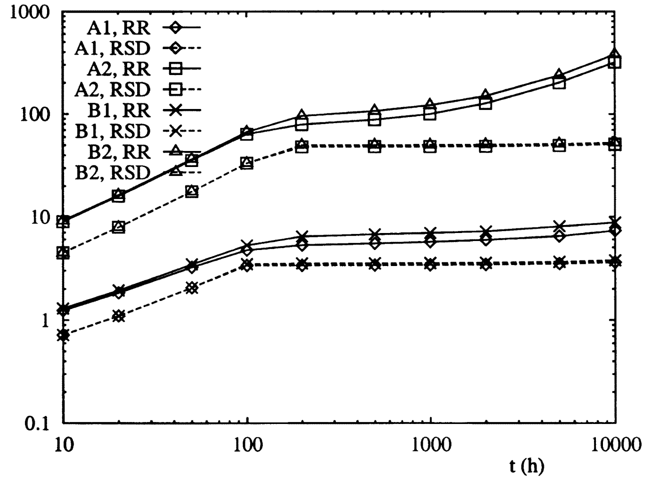

Fig. 7. CPU times (sec) required by RR and RSD for the example with N_{DS}=4 and the unavailability measure when the initial state is the state in which 1 disk of the first set is failed.

- in Table IV for the s-expected interval unavailability measure when the initial state is the state without failed components,
- in Table V for the s-expected interval unavailability measure when the initial state is the state in which 1 disk of the first set is failed.

The results for $N_{\mathrm{DS}} = 8$ are very similar.

For RR, the number of steps increases with t but, in accordance with the benign behavior, does so smoothly for large t. For RSD, the number of steps increases with t until it reaches the 'discrete time to stationarity' [27] and then remains constant. RSD requires exactly the same number of steps for both measures and both initial probability distributions. The performance of RR is slightly different for the unavailability and the s-expected interval-unavailability, and is affected by the initial probability distribution of the model. When the initial probability distribution is not concentrated in the state without failed components, RR always requires more steps than RSD. When the initial state is the state without failed components, then RR requires for small t less steps than RSD and there is a cross-point time t_C below which RR requires fewer steps than RSD, and above which RR requires more steps than RSD.

The t_C seems to increase as failure rates become smaller and is very large when failure rates are small (sets A1 and A2). For small t, the relative reduction in number of steps of RR in relation to RSD seems to decrease as the repair rates of the model become more different. Thus, for the unavailability measure and $t = 50$ hr, that reduction is 29% for set A1 and 9.4% for set A2. For large t, however the gain of RR over RSD in terms of number of steps is little affected by how different the repair rates of the model are. For instance, for the unavailability measure and $t = 10^4$ hr, the reduction achieved by RR over RSD is 9.7% for set A1 and 10% for set A2. The number of steps of both RR and RSD seem to depend mostly on the ratio between the largest and the smallest total repair rate from the states of the model (that ratio is 5 for sets A1 and B1 and 100 for sets A2 and B2). A theoretical explanation for that behavior for RR is in [4], [5] in terms of $R = \max_{i \in S}[\lambda_i]/\min_{i \in S'}[\lambda_i]$, which is approximately the ratio between the largest and smallest total repair rate from the states in S' of the model.

Next, RR and RSD are compared in terms of CPU times. The CPU times for the unavailability measure are given in

- Fig. 6 for the example with $N_{\mathrm{DS}} = 4$ and initial state the state without failed components,
- Fig. 7 for the example with $N_{\mathrm{DS}} = 4$ and initial state the state in which 1 disk of the first set is failed,

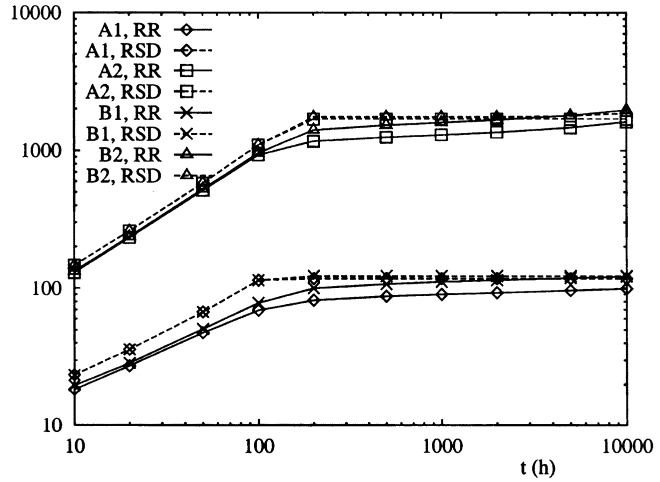

Fig. 8. CPU times (sec) required by RR and RSD for the example with $N_{\rm DS}$=8 and the unavailability measure when the initial state is the state without failed components.

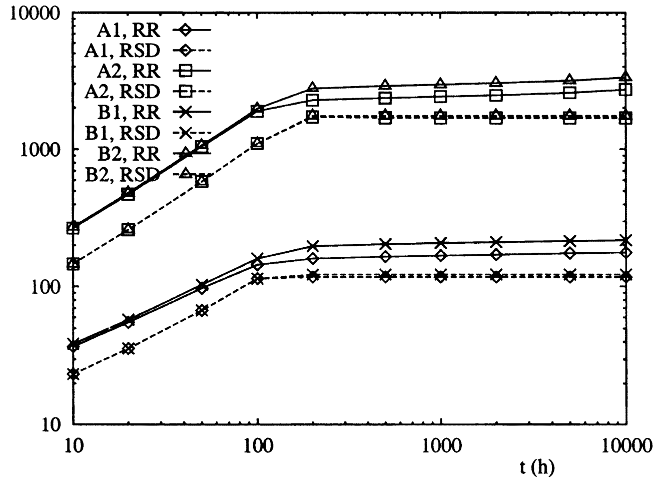

Fig. 9. CPU times (sec) required by RR and RSD for the example with $N_{\rm DS}$=8 and the unavailability measure when the initial state is the state in which 1 disk of the first set is failed.

- Fig. 8 for the example with $N_{\rm DS} = 8$ and initial state the state without failed components,
- Fig. 9 for the example with $N_{\rm DS} = 8$ and initial state the state in which 1 disk of the first set is failed.

The results for the s-expected interval unavailability measure are similar.

When the initial probability distribution is not concentrated in the state without failed components, then RR performs worse than RSD for all t—because,

- in that case, RR requires more steps and the steps in RR tend to be more costly than the steps in RSD, and
- the cost of the solution of the truncated transformed model by standard randomization in RR.

The last cost penalizes RR more strongly for larger t, and is relatively more important for $N_{\rm DS} = 4$, because of the smaller size of X. When the initial distribution of the model is concentrated in the state without failed components, RR is faster than RSD in some intervals. For large models ($N_{\rm DS} = 8$), the cost in RR of the solution of the truncated transformed model is relatively smaller and RR is more efficient than RSD for times below a quite large value. The width of the interval in which RR is faster than RSD decreases as failure rates become larger and as repair rates are more different. The last trend is because the relative importance of the second cost component in RR (solution of the truncated transformed model by standard randomization) increases as repair rates become more different (*e.g.*, for the unavailability measure with initial state the state without failed components and $t = 10^4$ hr that component accounted for 1.67% of the CPU time for set A1 and 13.2% for set A2). Because the relative importance of that component decreases with the size of X, for larger models that trend would be less important.

ACKNOWLEDGMENT

The author is pleased to thank the anonymous reviewers for their comments, which helped to improve this paper.

REFERENCES

[1] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions*: Dover, 1964.

[2] C. Béounes *et al.*, "SURF-2: A program for dependability evaluation of complex hardware and software systems," in *Proc. 23rd IEEE Int. Symp. Fault-Tolerant Computing , FTCS-23*, 1993, pp. 142–150.

[3] P. N. Bowerman, R. G. Nolty, and E. M. Schener, "Calculation of the poisson cumulative distribution function," *IEEE Trans. Rel.*, vol. 39, no. 2, pp. 158–161, 1990.

[4] J. A. Carrasco. (1999) Transient Analysis of Large Markov Models With Absorbing States Using Regenerative Randomization Technical Report DMSD_99_2. Univ. Politècnica de Catalunya. [Online]ftp://ftp-eel.upc.es/techreports

[5] ——, "Computation of bounds for transient measures of large rewarded markov models using regenerative randomization," *Computer and Operations Research*, vol. 30, pp. 1005–1035, 2003.

[6] G. Ciardo, J. Muppala, and K. Trivedi, "SPNP: Stochastic petri net package," in *Proc. 3rd IEEE Int. Workshop on Petri Nets and Performance Models, PNPM89*, 1989, pp. 142–151.

[7] J. Couvillon *et al.*, "Performability modeling with ultraSAN," *IEEE Software*, pp. 69–80, Sep 1991.

[8] B. L. Fox and P. W. Glynn, "Computing poisson probabilities," *Communications of the ACM*, vol. 31, pp. 440–445, 1988.

[9] A. Goyal, W. C. Carter, E. de Souza e Silva, and S. S. Lavenberg, "The system availability estimator," in *Proc. 16th IEEE Int. Symp. Fault-Tolerant Computing , FTCS-16*, 1986, pp. 84–89.

[10] W. K. Grassman, "Transient solutions in markovian queuing systems," *Computers and Operations Research*, vol. 4, pp. 47–53, 1977.

[11] ——, "Rounding errors in certain algorithms involving markov chains," *ACM Trans. Mathematical Software*, vol. 19, no. 4, pp. 496–508, Dec. 1993.

[12] D. Gross and D. R. Miller, "The randomization technique as a modeling tool and solution procedure for transient markov processes," *Operations Research*, vol. 32, pp. 343–361, 1984.

[13] M. Kijima, *Markov Processes for Stochastic Modeling*: Cambridge University Press, 1997.

[14] L. Knüsel, "Computation of the chi-square and poisson distribution," *SIAM J. Scientific and Statistical Computing*, vol. 7, no. 3, pp. 1023–1036, July 1986.

[15] 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, 1994.

[16] M. Malhotra, "A computationally efficient technique for transient analysis of repairable markovian systems," *Performance Evaluation*, no. 1–2, pp. 311–331, Nov. 1995.

[17] B. Melamed and M. Yadin, "Randomization procedures in the computation of cumulative-time distributions over discrete state markov processes," *Operations Research*, vol. 32, pp. 926–944, 1984.

[18] D. R. Miller, “Reliability calculation using randomization for markovian fault-tolerant computing systems,” in *Proc. 13th IEEE Int. Symp. on Fault-Tolerant Computing , FTCS-13*, 1983, pp. 284–289.

[19] A. P. Moorsel and W. H. Sanders, “Adaptive uniformization,” *Communications in Statistics — Stochastic Models*, vol. 10, no. 3, pp. 619–647, 1994.

[20] ——, “Transient solution of markov models by combining adaptive and standard uniformization,” *IEEE Trans. Rel.*, vol. 46, no. 3, pp. 430–440, 1997.

[21] H. Nabli and B. Sericola, “Performability analysis: A new algorithm,” *IEEE Trans. Computers*, vol. 45, no. 4, pp. 491–494, Apr. 1996.

[22] M. A. Qureshi and W. H. Sanders, “Reward model solution methods with impulse and rate rewards: An algorithm and numerical results,” *Performance Evaluation*, vol. 20, pp. 413–436, 1994.

[23] A. Reibman and K. S. Trivedi, “Numerical transient analysis of markov models,” *Computers and Operations Research*, vol. 15, pp. 19–36, 1988.

[24] S. M. Ross, *Stochastic Processes , 1983*: John Wiley & Sons.

[25] G. Rubino and B. Sericola, “Interval availability distribution computation,” in *Proc. 23th Int. Symp. Fault-Tolerant Computing FTCS-23*, 1993, pp. 48–55.

[26] ——, “Interval availability analysis using denumerable markov processes: Application to multiprocessor subject to breakdowns and repair,” *IEEE Trans. Computers*, vol. 44, no. 2, pp. 286–291, Feb. 1995.

[27] B. Sericola, “Availability analysis of repairable computer systems and stationarity detection,” *IEEE Trans. Computers*, vol. 48, no. 11, pp. 1166–1172, Nov. 1999.

[28] E. de Souza e Silva and H. R. Gail, “Calculating cumulative operational time distributions of repairable computer systems,” *IEEE Trans. Computers*, vol. 35, pp. 322–332, 1986.

[29] ——, “Calculating availability and performability measures of repairable computer systems using randomization,” *J. ACM*, vol. 34, pp. 179–199, 1989.

[30] ——, “An algorithm to calculate transient distributions of cumulative rate and impulse based reward,” *Communications in Statistics-Stochastic Models*, vol. 14, pp. 509–536, 1998.

Juan A. Carrasco has been “Profesor Titular” at the Dept. d’Enginyeria Electrònica of UPC (Polytechnical Univeristy of Catalonia) since 1988. His research is focused on modeling of digital systems and, more specifically, fault-tolerant systems. He received the Engineer degree in 1981 in Industrial Engineering from UPC; the M.S. in Computer Science in 1987 from Stanford University; and the Doctor Engineer degree in 1987 from UPC. He has been a member of the program committees of several international conferences. His current research interests encompass numerical techniques for the solution of Markov dependability and performability models, bounding methods, fast simulation methods, importance sampling techniques, combinatorial methods for dependability evaluation, and applications. He is the representative at UPC of the CaberNet Network of Excellence in distributed and dependable computing systems, funded by the European Commission.