Transient Analysis of Dependability/Performability Models by Regenerative Randomization with Laplace Transform Inversion*

Juan A. Carrasco

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

Abstract. In this paper we develop a variant of a previously proposed method (the regenerative randomization method) for the transient analysis of dependability/performability models. The variant is obtained by developing a closed-form expression for the solution of the truncated transformed model obtained in regenerative randomization and using a Laplace transform inversion algorithm. Using models of moderate size of a 5-level RAID architecture we compare the new variant with the original randomization method, with randomization with steady-state detection for irreducible models, and with the standard randomization method for transient models (models with absorbing states). The new variant seems to be competitive for models of moderate size.

1 Introduction

Homogeneous continuous time Markov chains (CTMCs) are frequently used for performance, dependability and performability modeling. Commonly used methods for the transient analysis of CTMCs are ODE (ordinary differential equation) solvers and randomization. Good recent reviews of these methods can be found in [6] and [12]. The randomization method (also called uniformization) is attractive because it has guaranteed numerical stability, since it involves additions of positive numbers, and the computation error can be specified in advance. Let Λ be the maximum output rate of the CTMC in consideration. Then, the number of steps required by the method is roughly equal to Λt when Λt is large. For models of repairable fault-tolerant systems the t of interest makes typically Λt very large and, then, randomization is very inefficient.

Several variants of the (standard) randomization method have been proposed to improve its efficiency. Miller has used selective randomization to solve reliability models with detailed representation of error handling activities [7]. Reibman

* This work was supported by the "Comisión Interministerial de Ciencia y Tecnología" (CICYT) under the research grant TAP99-0443-C05.

J. Rolim et al. (Eds.): IPDPS 2000 Workshops, LNCS 1800, pp. 1226-1235, 2000.

and Trivedi [12] have proposed a more general approach based on the multistep concept. However, that method introduces fill-in in the transition probability matrix of the randomized discrete-time Markov chain (DTMC). In adaptive uniformization (randomization) [8] the randomization rate is adapted depending on the states in which the randomized DTMC can be at a given step. Adaptive randomization seems to be faster than the standard randomization method for small and medium mission times. In addition, it can be used to solve models with infinite state spaces and not uniformly bounded transition rates. Recently, it has been proposed the combination of adaptive and standard randomization [9]. Another recent proposal to speed up the randomization method when X is irreducible is steady-state detection [6]. Recently, a method based on steady-state detection which gives error bounds has been developed [14]. Regenerative randomization [1,2] is another recent proposal.

In this paper we develop a variant of the regenerative randomization method described in [1,2]. The state space of the CTMC X is assumed to be $\Omega = S \cup \{f_1, f_2, \ldots, f_A\}$, where f_i are absorbing states and all states in S are strongly connected and have paths to f_i (for $A = 0$ X is irreducible). We will assume $P[X(0) = f_i] = 0$, $1 \leq i \leq A$. In addition, we assume a reward rate structure $r_i \geq 0$, $i \in \Omega$, with different reward rates assigned to the A absorbing states. We consider two measures, the transient reward rate at time t, $TRR(t)$, and the mean reward rate during the interval $[0,t]$, $MRR(t)$. The rest of the paper is organized as follows. Section 2 describes the variant. Section 3 compares the variant with the original randomization method, randomization with steady-state detection, and standard randomization. Section 4 concludes the paper.

2 The New Variant

The regenerative randomization method is described in detail in [1,2]. In that method, a transformed truncated CTMC $V_{K,L}$ (V_K) is obtained in terms of which can be expressed with a preespecified accuracy the measures $TRR(t)$ and $MRR(t)$. Regenerative randomization requires the selection of a regenerative state r and its performance will be good when r is visited often in the DTMC $\widehat{X}$ obtained by randomizing X with rate Λ. Let $\alpha_i = P[X(0) = i]$. Figure 1 illustrates the state transition diagram of $V_{K,L}$, truncated transformed model for the case $\alpha_r < 1$; the initial probability distribution of $V_{K,L}$ is $P[V_{K,L}(0) = s_0] = \alpha_r$, $P[V_{K,L}(0) = s'_0] = 1 - \alpha_r$, $P[V_{K,L}(0) = i] = 0$, $i \neq s_0, s'_0$. The reward rate structure of $V_{K,L}$ is $r_{s_k} = b(k)$, $r_{s'_k} = b'(k)$, $r_a = 0$. Then, the $TRR(t)$ and $MRR(t)$ measures for the original CTMC X can be computed with given error bound $\epsilon/2$ (the remaining $\epsilon/2$ is reserved for the solution of the truncated transformed models) as the $TRR(t)$ and $MRR(t)$ measures of $V_{K,L}$, called $TRR^a_{K,L}(t)$ and $MRR^a_{K,L}(t)$. In the particular case $\alpha_r = 1$, the truncated transformed model, V_K, has identical structure as $V_{K,L}$, except that states s'_k disappear. The approximated values for the desired measures are denoted in this case by $TRR^a_K(t)$ and $MRR^a_K(t)$.

The parameters q_i, w_i, v^j_i, q'_i, w'_i, v'^j_i, $b(k)$ and $b'(k)$ on which the truncated transformed models depend can be obtained by stepping DTMCs of about the

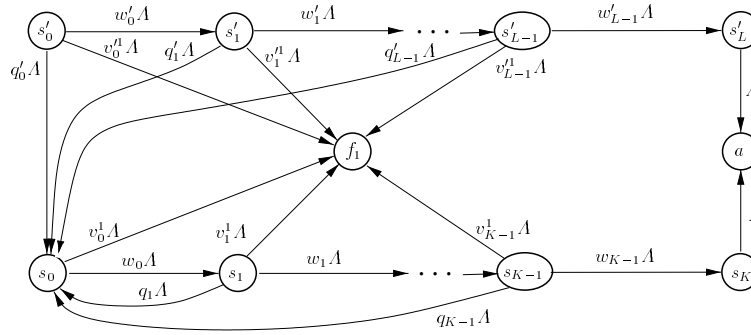

Fig. 1. State transition diagram of the CTMC $V_{K,L}$ for the case $\Lambda = 1$.

same size as $\widehat{X}$ $K + L$ steps when $\alpha_r < 1$ and K steps when $\alpha_r = 1$. The parameters w_k obey the relation $w_k = a(k+1)/a(k)$, where $a(0) = 1$. Similarly, the parameters w'_k satisfy $w'_k = a'(k+1)/a'(k)$, where $a'(0) = 1 - \alpha_r$. The parameters $a(k)$ and $a'(k)$ are easily computable while stepping the same DTMCs as above.

In the regenerative randomization method, $V_{K,L}$ or V_K is solved using standard randomization, which can be relatively expensive when Λt is large and the transformed truncated model is not significantly smaller than X. The new variant proposed here is called regenerative randomization with Laplace transform inversion and combines a closed-form solution of the $V_{K,L}$ (V_K) model in the Laplace transform domain with the usage of a numerical Laplace inversion algorithm.

2.1 Closed form solution in the Laplace transform domain

Given the structure of $V_{K,L}$, it is possible to obtain closed-form expressions for the Laplace transforms of the transient probabilities of the states of the CTMC. Details can be found in [3]. From them, it is possible to find a closed-form expression for the Laplace transform of $TRR^a_{K,L}(t)$. Let $c(k) = a(k)b(k)$ and $c'(k) = a'(k)b'(k)$, we have

$$\begin{aligned}
\widetilde{TRR}^a_{K,L}(s) &= \sum_{k=0}^{K} b(k)\widetilde{p}_k(s) + \sum_{k=0}^{L} b'(k)\widetilde{p}'_k(s) + \sum_{i=1}^{A} r_{f_i}\widetilde{p}_{f_i}(s) \\
&= \left[\sum_{k=0}^{K} c(k)\left(\frac{\Lambda}{s+\Lambda}\right)^k + \frac{\Lambda}{s}\sum_{k=0}^{K-1}\left(\sum_{i=1}^{A} r_{f_i}v^i_k\right)a(k)\left(\frac{\Lambda}{s+\Lambda}\right)^k\right]\widetilde{p}_0(s) \\
&\quad + \sum_{k=0}^{L} c'(k)\frac{\Lambda^k}{(s+\Lambda)^{k+1}} + \sum_{k=0}^{L-1}\left(\sum_{i=1}^{A} r_{f_i}v'^i_k\right)a'(k)\frac{\Lambda^{k+1}}{s(s+\Lambda)^{k+1}},
\end{aligned}$$

where

$$\widetilde{p}_0(s) = \frac{A(s)}{B(s)},$$

$$A(s) = 1 - \frac{s}{s+\Lambda}\sum_{k=0}^{L} a'(k)\left(\frac{\Lambda}{s+\Lambda}\right)^k - \frac{\Lambda}{s+\Lambda}\sum_{k=0}^{L-1}\left(\sum_{i=1}^{A} v_k'^i\right) a'(k)\left(\frac{\Lambda}{s+\Lambda}\right)^k - a'(L)\left(\frac{\Lambda}{s+\Lambda}\right)^{L+1},$$

and

$$B(s) = s\sum_{k=0}^{K} a(k)\left(\frac{\Lambda}{s+\Lambda}\right)^k + \Lambda\sum_{k=0}^{K-1}\left(\sum_{i=1}^{A} v_k^i\right) a(k)\left(\frac{\Lambda}{s+\Lambda}\right)^k + a(K)\Lambda\left(\frac{\Lambda}{s+\Lambda}\right)^K.$$

The Laplace transform of $MRR^a_{K,L}(t)$ can be easily obtained noting that $MRR^a_{K,L}(t) = \int_0^t TRR^a_{K,L}(\tau)\,d\tau/t$. Then defining $C_{K,L}(t) = tMRR^a_{K,L}(t)$ we have $\widetilde{C}_{K,L}(s) = \widetilde{TRR}^a_{K,L}(s)/s$.

In the particular case $\alpha_r = 1$ we have $\widetilde{p}_0(s) = 1/B(s)$,

$$\widetilde{TRR}^a_K(s) = \left[\sum_{k=0}^{K} c(k)\left(\frac{\Lambda}{s+\Lambda}\right)^k + \frac{\Lambda}{s}\sum_{k=0}^{K-1}\left(\sum_{i=1}^{A} r_{f_i} v_k^i\right) a(k)\left(\frac{\Lambda}{s+\Lambda}\right)^k\right]\widetilde{p}_0(s),$$

and defining $C_K(t) = tMRR^a_K(t)$, $\widetilde{C}_K(s) = \widetilde{TRR}^a_K(s)/s$.

2.2 Numerical Laplace inversion

There are several numerical Laplace inversion algorithms. We have experimented with the methods proposed in [4] and [11]. Both are based on Durbin's approximation for $f(t)$ [5]:

$$f_a(t) = \frac{1}{T}e^{at}\left[\frac{\widetilde{f}(a)}{2} + \sum_{k=1}^{\infty}\Re\left\{\widetilde{f}\left(a + \frac{ik\pi}{T}\right)e^{\frac{ik\pi t}{T}}\right\}\right], \tag{1}$$

where $i = \sqrt{-1}$. The approximation error is:

$$f_\epsilon(t) = \sum_{k=1}^{\infty} f(2kT + t)e^{-2akT}.$$

The method described in [4] takes $T = t$ and accelerates the convergence of the series of (1) using the epsilon algorithm. We have found that the method is fast, but it is sometimes unstable. On the other hand, the method described in [11], which only differs from the method described in [4] in that it takes $T = 16t$ is very stable but significantly slower. Thus, we decided to experiment with several choices for T, increasing from $T = t$ to $T = 16t$. We found that $T = 8t$ gave enough stability, and we use that selection of T.

To control the error of the Laplace inversion algorithm we proceed as follows. The total error on $TRR^a_{K,L}(t)$ ($MRR^a_{K,L}(t)$) must be $\leq \epsilon/2$. There are two kinds of errors: the approximation error and the truncation error (resulting from the truncation of the convergent series) and we allocate $\epsilon/4$ to each of them. For the measure $TRR(t)$ we have $TRR^a_{K,L}(t) \leq r_{\max}$, where $r_{\max} = \max_{i\in\Omega} r_i$, and

$$(TRR^a_{K,L})_\epsilon(t) \leq \sum_{k=1}^{\infty} r_{\max} e^{-2akT} = r_{\max} \frac{e^{-2aT}}{1-e^{-2aT}}.$$

To bound the approximation error we take the a satisfying

$$r_{\max} \frac{e^{-2aT}}{1-e^{-2aT}} = \frac{\epsilon}{4},$$

i.e.

$$a = -\frac{1}{2T} \log\left(\frac{1}{1+\frac{4r_{\max}}{\epsilon}}\right).$$

Regarding the truncation error we only have control over the tolerance between consecutive approximations of the (accelerated) convergent series and we decide achieved the convergence when that difference is $\leq \epsilon/100$, i.e. we leave a factor 25 to account for the difference between the tolerance between consecutive values and the actual truncation error.

For the measure $MRR(t)$ we in fact invert $\widetilde{C}_{K,L}(s)$, where $C_{K,L}(t) = tMRR^a_{K,L}(t)$. Then, to have an error $\epsilon/2$ in $MRR^a_{K,L}(t)$ we must require an error $t\epsilon/2$ in the inversion of $\widetilde{C}_{K,L}(s)$. We allocate $t\epsilon/4$ for the approximation error and $t\epsilon/4$ for the truncation error, with a factor 25 as before. We have $C_{K,L}(t) \leq r_{\max} t$ and

$$(C_{K,L})_\epsilon(t) \leq \sum_{k=1}^{\infty} r_{\max}(2kT+t)e^{-2akT} = r_{\max} t \sum_{k=1}^{\infty} e^{-2akT} + 2r_{\max} T \sum_{k=1}^{\infty} k e^{-2akT}$$
$$= r_{\max} t \frac{e^{-eaT}}{1-e^{-2aT}} + 2r_{\max} T \frac{e^{-2aT}}{(1-e^{-2aT})^2} = r_{\max} \frac{(t+2T)e^{-2aT} - te^{-4aT}}{(1-e^{-2aT})^2}.$$

To bound the approximation error we take the a with $e^{-2aT} < 1$ satisfying

$$r_{\max} \frac{(t+2T)e^{-2aT} - te^{-4aT}}{(1-e^{-2aT})^2} = \frac{\epsilon}{4},$$

i.e.

$$a = \frac{1}{2T} \log\left(\frac{1}{x}\right),$$

$$x = \frac{\frac{\epsilon}{2} + (t+2T)r_{\max} - \sqrt{(\frac{\epsilon}{2} + (t+2T)r_{\max})^2 - (\frac{\epsilon}{4} + tr_{\max})\epsilon}}{\frac{\epsilon}{2} + 2tr_{\max}}. \qquad (2)$$

Regarding the truncation error we only have control over the tolerance between consecutive approximations of the (accelerated) convergent series and we decide achieved the convergence when that difference is $\leq \epsilon t/100$. Expression (2) has severe cancellation errors when $y = \sqrt{(\epsilon/4 + tr_{\max})\epsilon}/(\epsilon/2 + (t+2T)r_{\max}) \ll 1$. The problem can be solved by taking the Taylor series on y and use it when y is small, say $y < 10^{-3}$. In that case, the Taylor series gives

$$x \approx \frac{(\frac{\epsilon}{4} + tr_{\max})\epsilon}{(\frac{\epsilon}{2} + (t+2T)r_{\max})(\frac{\epsilon}{2} + 2tr_{\max})}.$$

Up to now we have implicitly consider the case $\alpha_r < 1$. The case $\alpha_r = 1$ is treated identically. Efficient algorithms to compute the Laplace transforms required by the inversion algorithm can be found in [3].

3 Analysis and Comparison

In this section we analyze the performance of the proposed variant of the regenerative randomization method (RRL). For irreducible models ($A = 0$) we will compare RRL with the original regenerative randomization method (RR) and randomization with steady-state detection (RSD) [14]. For models with absorbing states ($A \geq 1$), we will compare RRL with RR and standard randomization (SR). The analysis and comparison will be made using dependability models of a level 5 RAID architecture [10]. The models we will consider are similar to a model described in [13]. Our models consider hot spares for controllers, which were not considered in [13] and encompass availability measures.

Figure 2 shows the architecture of the considered level 5 RAID system. The system includes $G \times N$ disks and N controllers. The disks are organized in G parity groups, each with N disks. Each controller controls a string of G disks. The system also includes C_{H} hot spare controllers and D_{H} hot spare disks. The system is operational if there is access to at least $N-1$ available disks of each parity group. When there is a failed controller all disks of the associated string become unavailable. When a failed disk is replaced by a good one and if all disks of the parity group are available, the parity group starts the reconstruction of data in the replaced disk. The reconstruction process also starts when a disk of a parity group which was not available due to failure of one controller becomes available due to the replacement of the failed controller. All disks of the parity group involved in a reconstruction are "overloaded" and have a higher failure rate. Non-overloaded disks fail with rate λ_{D}. Overloaded disks fail with rate λ_{S}. Controllers fail with rate λ_{C}. The reconstruction process has an exponential duration with rate μ_{DRC}. Failed disks and controllers are replaced, if respective hot spares are available, by a repairman with rates μ_{DRP} and μ_{CRP}, respectively, with priority given to controllers. Lacking spares and failed disks and controllers for which there are not spares are replaced with rate μ_{SR} by an unlimited number of repairmen. A reconstruction process is successful with probability P_{R}. Failure in a reconstruction process causes the failure of the system. Finally, when the system is failed, it is returned to its original state, with all disks and hot spares

available, by a global repair action which has rate μ_G. The exact model gives very large CTMCs for moderate values of G and N. Instead, we will use a pessimistic approximated model giving CTMCs with much smaller size. Unavailable disks are said to be aligned if they belong to the same string. The approximation consists in assuming that if unavailable disks are not aligned, when one of them becomes available the remaining disks would still be unaligned whenever their number is ≥ 2. Using that approximation it is possible to describe the state of the CTMC using the following state variables: NFD (number of failed disks), NDR (number of disks under reconstruction), NWD (number of disks waiting for reconstruction), NSD (number of hot spare disks), AL (a boolean variable which is YES when unavailable disks are aligned and NO otherwise), NFC (number of failed controllers), NSC (number of hot spare controllers), and F (a boolean value which is YES when the system is failed and NO otherwise). We will vary the model parameters G, C_H, and D_H and will fix the other parameters of the model to the values: $N = 5$, $\lambda_D = 10^{-5}$, $\lambda_S = 2 \times 10^{-5}$, $\lambda_C = 5 \times 10^{-5}$, $\mu_{DRC} = 1$, $\mu_{DRP} = 4$, $\mu_{CRP} = 4$, $\mu_{SR} = 0.25$ and $\mu_G = 0.25$, with all rates in h^{-1}. We will consider two measures. The first them is a particular case of $TRR(t)$ when the model is irreducible ($A = 0$), is point unavailability $UA(t)$ and is obtained by assigning a reward rate 0 to the operational states and a reward rate 1 to the failed state. The second of them is a particular case of $TRR(t)$ when the model has absorbing states, is the unreliability $UR(t)$, and is obtained by making the system failed state absorbing (and thus $A = 1$) and assigning a reward rate 1 to the absorbing state and a reward rate 0 to the transient states. For all measures we will assume that the initial state is the state without failed components and all hot spares available, which will be taken as regenerative state for the methods RR and RRL. For all methods we will take $\epsilon = 10^{-12}$, which gives enough accuracy for all measures and values of t.

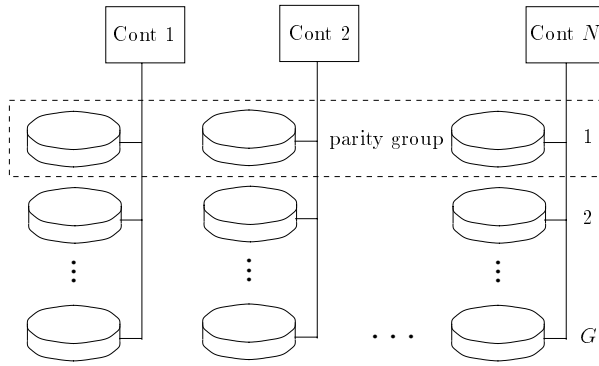

Fig. 2. Architecture of the considered level 5 RAID system.

To analyze the performance of the proposed RRL method and compare it with that of RR, RSD and SR we will use two instances of the parametric model

defined by $C_H = 1$, $D_H = 3$ and $G = 20$ for one instance and $G = 40$ for the second instance. The irreducible models have 3,841 states and 24,785 transitions for $G = 20$ and 14,081 states and 94,405 transitions for $G = 40$. The models with absorbing state have the same number of states and one transition less. We start comparing RRL, RR, and RSD for the measure $UA(t)$. Table 1 gives the number of steps required by RRL and RR (both require the same number of steps) and RSD for several values of t. Figure 3 plots the corresponding CPU times. We can note that RRL and RR require fewer steps than RSD up to a certain value of t. Regarding CPU times, there are crosspoints between RR and RRL in one hand and RSD on the other hand. RRL is about as fast as RSD and significantly faster than RR for large t. The numerical Laplace transform inversion is fast and consumes a very small percentage of the time of the RRL method (about 2 % for the example with $G = 20$ and 1 % for the example with $G = 40$). The number of required abscissae varied from 105 to 329.

Table 1. Number of steps required by RR, RRL and RSD for the measure $UA(t)$ for several values of t.

	$G = 20$		$G = 40$	
t (h)	RR/RRL	RSD	RR/RRL	RSD
1	56	66	86	99
10	323	355	554	594
100	2,234	2,612	4,187	4,823
1,000	2,708	2,612	5,123	4,823
10,000	2,938	2,612	5,549	4,823
100,000	3,157	2,612	5,957	4,823

We next compare RRL, RR and SR using the example with the measure $UR(t)$. Table 2 and Figure 4 give the results. For small t, SR is slightly faster than both RR and RRL. Similarly, for models with $A = 0$ such as the one considered previously, SR should be slightly faster than RRL, RR and RSD. We can note that SR is extremely expensive for large t. For those t, RR performs better than SR and the fastest method is the proposed RRL, which outperforms RR significantly. We note that for the largest t considered ($t = 100,000$ h), $UR(t)$ is 0.50480 for the model with $G = 20$ and 0.74750 for the model with $G = 40$. Thus, the selection $\epsilon = 10^{-12}$ is a very stringent one and translates to require about 14 digits of accuracy to the numerical Laplace inversion algorithm. Thus, that algorithm seems to be very stable.

4 Conclusions

We have proposed a new variant of the regenerative randomization method for the transient analysis of dependability/performability models. For irreducible

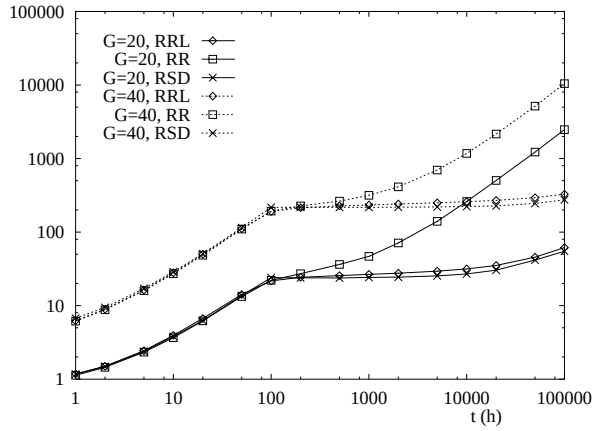

Fig. 3. CPU times in seconds required by RRL, RR and RSD for the measure $UA(t)$ as a function of t.

Table 2. Number of steps required by RR, RRL and SR for the measure $UR(t)$ for several values of t.

	$G = 20$		$G = 40$	
t (h)	RR/RRL	SR	RR/RRL	SR
1	56	65	86	98
10	323	354	554	593
100	2,233	2,726	4,186	4,849
1,000	2,708	24,844	5,122	45,234
10,000	2,937	240,958	5,547	442,203
100,000	3,157	2,386,068	5,955	4,390,141

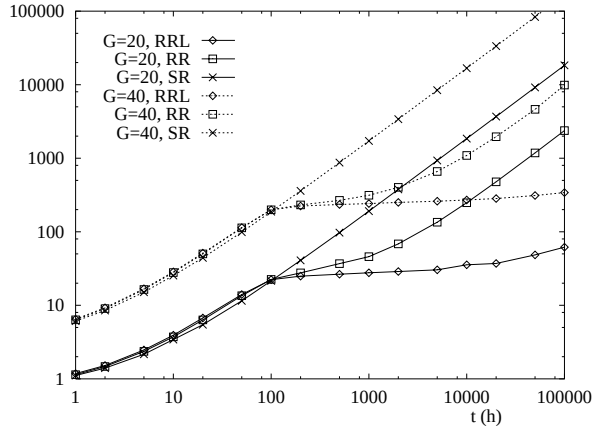

Fig. 4. CPU times in seconds required by RRL, RR and SR for the measure $UR(t)$ as a function of t.

models, the new variant seems to be about as fast as randomization with steady-state detection and, for large t and models of moderate size, significantly faster than the original regenerative randomization. For models with absorbing states and large t, the new variant is much faster than standard randomization and significantly faster than the original regenerative randomization for models of moderate size.

References

1. J. A. Carrasco, "Solving Large Reliability Models of Repairable Fault Tolerant Systems using Regenerative Randomization," Technical Report DMSD_99_2, Universitat Politècnica de Catalunya, February 1999, available at `ftp://ftp-eel.upc.es/techreports`.
2. J. A. Carrasco, "Efficient Computation of Bounds for Transient Performability Measures using Regenerative Randomization," Technical Report DMSD_99_4, Universitat Politècnica de Catalunya, May 1999, available at `ftp://ftp-eel.upc.es/techreports`.
3. J, A. Carrasco, "Transient Analysis of Dependability/Performability Models by Regenerative Randomization with Laplace Transform Inversion," Technical Report DMSD_2000_1, Universitat Politècnica de Catalunya, May 1999, available at `ftp://ftp-eel.upc.es/techreports`.
4. K. S. Crump, "Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation," *Journal of the ACM*, vol. 23, pp. 89–96, 1976.
5. F. Durbin, "Numerical inversion of Laplace transforms: an efficient improvement to Dubner and Abate's method," *Computer Journal*, vol. 17, no. 4, 1974, pp. 371–376.
6. 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, 1994, pp. 1825–1841.
7. D. R. Miller, "Reliability Calculation using Randomization for Markovian Fault-Tolerant Computing Systems," *in Proc. 13th IEEE Int. Symp. on Fault-Tolerant Computing, FTCS-13*, June 1983, pp 284–289.
8. A. P. Moorsel and W. H. Sanders, "Adaptive Uniformization", *Communications in Statistics - Stochastic Models*, vol. 10, no. 3, 1994, pp. 619–647.
9. A. P. Moorsel and W. H. Sanders, "Transient Solution of Markov Models by Combining Adaptive and Standard Uniformization," *IEEE Trans. on Reliability*, vol. 46, no. 3, 1997, pp. 430–440.
10. D. A. Patterson, P. M. Chen, G. A. Gibson, and R. H. Katz, "Introduction to redundant arrays of inexpensive disks (RAID," in *Proc. IEEE Computer Society International Conference (COMPCON)*, 1989, pp. 112–117.
11. R. Piessens and R. Huysmans, "Algorithm 619: Automatic Numerical Inversion of the Laplace Transform [D5]," *ACM Transactions on Mathematical Software*, vol. 10, no. 3, pp. 348–353, September 1984.
12. A. Reibman and K. S. Trivedi, "Numerical Transient Analysis of Markov Models," *Comput. Operations Res.*, vol. 15, 1988, pp. 19–36.
13. V. Santoja, M. Alonso, J. Molero, J. J. Serrano, P. Gil, and R. Ors, "Dependability Models of RAID using Stochastic Activity Networks," *Dependable Computing-EDCC-2*, Lecture Notes in Computer Science, 1996.
14. B. Sericola, "Availability Analysis of Repairable Computer Systems and Stationarity Detection," *IEEE Trans. on Computers*, vol. 48, no. 11, November 1999, pp. 1166–1172.