Pergamon

Computers Math. Applic. Vol. 35, No. 8, pp. 41–51, 1998

0898-1221/98 \$19.00 + 0.00

PII: S0898-1221(98)00043-1

Performability Measure for Acyclic Markovian Models

H. NABLI
Faculté des Sciences de Monastir, Tunisia
and
RSR Laboratory of the Ecole Nationale des Sciences de l'Informatique de Tunis

(Received November 1995; revised and accepted October 1997)

Abstract—Continuous-time Markov processes with a finite-state space are generally considered to model degradable fault-tolerant computer systems. The finite space is partitioned as $\cup_{i=1}^{m} B_i$, where B_i stands for the set of states which corresponds to the configuration where the system has a performance level (or reward rate) equal to r_i. The performability Y_t is defined as the accumulated reward over a mission time $[0, t]$. In this paper, a renewal equation is established for the performability measure and solved for both "standard" and uniform acyclic models. Two closed form expressions for the performability measure are derived for the two types of models. Furthermore, an algorithm with a low polynomial computational complexity is presented and applied to a degradable computer system.

Keywords—Acyclic models, Degradable systems, Markovian models, Performability.

1. INTRODUCTION

As recognized in a large number of studies, the quantitative evaluation of degradable computer systems requires dealing simultaneously with aspects of both performance and reliability. As part of these studies, Meyer [1] introduces a unified measure called performability which combines the two aspects of performance and reliability. Performability is defined as the accomplishment level of the system over a specified time period t. The distribution $\mathbb{P}\{Y_t \in B\}$ is then the probability that the system performs at a level in B, where B is a set of accomplishment levels.

Formally, the system fault behavior is assumed to be modeled by a homogeneous Markov process $(X_s)_{s \geq 0}$ over a finite-state space $E = \{0, 1, \dots, n\}$. A reward rate $\rho(i)$ (or performance level) is associated with each state $i \in E$. This reward measures how well the system performs in the corresponding configuration. Since we consider degradable systems, it must be that $\rho(i) \geq \rho(j)$ if a transition is possible from state i to state j. Therefore, we can number the states so that $i \mapsto \rho(i)$ becomes an increasing function (i.e., $\rho(i) \geq \rho(j)$ if $i \geq j$) and a transition from state i goes only to a state j satisfying $j < i$. Since two different states may have the same reward rate, we denote by $r_m > r_{m-1} > \cdots > r_0$ the $m+1$ different reward rates ($m \leq n$). With the above notations, the performability, or the accumulated reward over the mission time $[0, t]$, is defined by

The author would like to thank B. Sericola and R. Marie for their invaluable suggestions.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

$$Y_t = \int_0^t \rho(X_s)\,ds = \sum_{i=0}^{n} \rho(i) \int_0^t \mathbf{1}_{\{X_s=i\}}\,ds,$$

where $\mathbf{1}_c = 1$, if condition c is true and 0 otherwise.

The random variable Y_t takes its values in the interval $[r_0 t, r_m t]$ and we wish to derive $\mathbb{P}\{Y_t > s\}$. The reward rates r_i are arbitrary real numbers, but we can assume $r_0 = 0$, without loss of generality. This is obtained by replacing r_i by $r_i - r_0$ and s by $s - r_0 t$. Henceforth, we take $r_0 = 0$.

The distribution of performability has been studied in previous papers for acyclic models, which correspond to nonrepairable computer systems, and also for cyclic models, which correspond to repairable computer systems. Several methods have been proposed to compute the probability distribution of performability during an interval of time $[0, t]$. For cumulative operational time (i.e., when the reward rates are either 0 or 1), De Souza e Silva and Gail [2] compute the distribution of interval availability using the uniformization technique [3]. The computational complexity of this method has been improved in [4,5]. In [6], a system of integral equations is established for the interval availability for a semi-Markov process. The distribution of cumulative operational time is then computed for a two-unit system with sequential preventive maintenance by solving the integral equations via a two-point trapezoidal rule. In [7], a closed form expression is obtained for the joint distribution of cumulative operational time and the number of visits to up states during $[0, t]$. Time-discretization is also considered as a technique for the computation of the distribution of the cumulative operational time [8,9].

The distribution of accumulated reward over a finite mission time (with general reward rates) is more complex to obtain. Meyer [10] obtains a closed form expression for the distribution of performability for a degradable computer system with N processors and a buffer with finite capacity. Furchtgott and Meyer [11,12] define i-resolvable vectors to characterize the trajectories of an acyclic semi-Markovian process corresponding to a certain performance level. By enumerating all the possible trajectories of the system, they derive an integral expression for performability which they solve numerically. However, the complexity of such an algorithm is exponential in the number of states of the process. Beaudry [13] gives a method for the computation of performability in a Markovian process until absorption. Ciardo *et al.* [14] generalize Beaudry's approach to a semi-Markov reward process and remove the restriction requiring only the absorbing states to be associated with a zero reward rate. Iyer *et al.* [15] propose an algorithm to compute recursively the moments of the accumulated reward over the mission time, with a polynomial computational complexity in the number of states. In [16], Nabli and Sericola present an algorithm, based on the uniformization technique and the result of [17], to compute this distribution for block acyclic models which are more general than acyclic one's. They determine new truncation steps which improve the execution time of their algorithm. Goyal and Tantawi [18] derive a closed form expression (precisely, a finite sum of exponential functions) for the performability of degradable heterogeneous systems. They also give an algorithm with a polynomial complexity $O(d(n+1)^3)$ in the number $(n+1)$ of states and in the number d of components in the system. They consider at first the homogeneous case (i.e., only transitions between state i to state $i-1$ are allowed). However, their generalization to the nonhomogeneous, or heterogeneous, case is not clear. A method which follows an approach similar to the one used by Goyal and Tantawi will be presented in Section 2. An algorithm to compute the probability distribution of performability, with a low computational complexity in comparison with the algorithm of Goyal and Tantawi, will be derived.

The remainder of this paper is organized as follows. In the next section, we give a solution for computing performability for acyclic Markovian models. We also discuss the computational complexity of the algorithm. Section 3 is dedicated to the performability measure for uniform acyclic models. Such models are characterized by the uniformity of the yield between the output rate $-\lambda_{ii}$ of state i and the reward rate $\rho(i)$ associated to the state i (i.e., for all $i \in E$, $(-\lambda_{ii}/\rho(i))$

are equal). An algorithm with a low polynomial complexity will be presented. A numerical example for a degradable computer system is presented and solved for a given performability measure in Section 4. The main points are summarized in the concluding section.

2. MODEL SOLUTION

We consider a nonrepairable computer system with d types of components. Mathematically, that means that the connectivity degree of the matrix A is equal to d. Each state $i \in \{0, 1, \dots, n\}$ may be described by a vector $(k_1, \dots, k_d)$ where k_l is the number of functioning components of type l, $l = 1, \dots, d$. The sequence of states visited is governed by the transition rates λ_{ij} $(i, j \in E,\ i \neq j)$. The homogeneous Markov process $(X_s)_{s \geq 0}$ is entirely determined by its infinitesimal generator $(\lambda_{ij})_{i,j \in E}$ and its initial state which is assumed to be $X_0 = n$. That means that the system starts in state n which has the largest reward rate. Because the system is nonrepairable, the infinitesimal generator is an upper triangular matrix.

Our approach consists of expressing the distribution of Y_t with a closed form expression. More exactly, we will evaluate the performability measure by means of a finite sum of exponential functions each affected by coefficients calculated by recurrence. We will use the same closed form expression derived by Goyal and Tantawi [18]. In return, we give recurrence formulas which lead to a lower computational complexity (see Section 2.2).

2.1. Renewal Equation

Let us define the following notations:

$$F_n(s,t) = \mathbb{P}\left\{ Y_t > \frac{s}{X_0} = n \right\}, \qquad \text{for } s \in [0, r_m t[,$$

$$\tau_{n,j} = \begin{cases} 0, & \text{if } s - r_j t \leq 0 \text{ or } \rho(n) = r_j, \\ \dfrac{s - r_j t}{\rho(n) - r_j}, & \text{otherwise,} \end{cases}$$

and

$$\lambda_i = -\lambda_{ii} = \sum_{j \neq i} \lambda_{ij} = \sum_{j=0}^{i-1} \lambda_{ij} : \text{ the output rate of state } i.$$

We note that $F_n(s,t) = 1$ if $s < 0$ and $F_n(s,t) = 0$ if $s \geq r_m t$. This remark leads us to consider the distribution only for $s \in [0, r_m t[$. It is well known (see [18], for example) that the distribution $F_n(s,t)$ satisfies the following renewal equation:

$$F_n(s,t) = e^{-(\lambda_n/\rho(n))s} + \sum_{i=1}^{n-1} \int_0^{s/(\rho(n))} e^{-\lambda_n u} \lambda_{n,i} F_i\left(s - \rho(n)u, t - u\right) du. \tag{1}$$

Equation (1) gives a recursive relation upon the index n. Our analysis consists of obtaining a recursive relationship not only upon index n but also upon index j by considering the partition $[0, r_m t[= \cup_{j=1}^{m} [r_{j-1}t, r_j t[$. For this purpose, we define

$$F_n^{(j)}(s,t) = F_n(s,t) \mathbf{1}_{\{r_{j-1}t \leq s < r_j t\}}.$$

Seeing that two different states may have the same reward rate, we define, for each $j \in \{1, \dots, m\}$, the index $\phi_j = \min\{k \in E / \rho(k) = r_j\}$. ϕ_j stands for the first state for which the reward rate is r_j. We also denote by ω_l the index satisfying $\rho(l) = r_{\omega_l}$. With these notations, we have the following lemma.

LEMMA 2.1. *For all* $j_0 \in \{1,\dots,m\}$ *and for all* $s \in [r_{j_0-1}t, r_{j_0}t[$, *we have*

$$F_n^{(j_0)}(s,t) = e^{-(\lambda_n/\rho(n))s} + \sum_{j=1}^{j_0} \int_{\tau_{n,j}}^{\tau_{n,j-1}} e^{-\lambda_n u} \sum_{i=\phi_j}^{n-1} \lambda_{n,i} F_i^{(j)}(s-\rho(n)u, t-u)\, du.$$

PROOF. According to equation (1) and the definition of $\tau_{n,j}$, we have

$$F_n(s,t) = e^{-(\lambda_n/\rho(n))s} + \sum_{i=1}^{n-1} \int_0^{\tau_{n,0}} e^{-\lambda_n u} \lambda_{n,i} F_i\left(s-\rho(n)u, t-u\right) du.$$

Our goal is to express this equation in accordance with $F_i^{(j)}(.,.)$ instead of $F_i(.,.)$. So, since $F_i(s-\rho(n)u, t-u) = 0$ when $s-\rho(n)u > r_j(t-u)$ and $j \geq \omega_i$, we obtain

$$\begin{aligned} F_i\left(s-\rho(n)u, t-u\right) &= F_i\left(s-\rho(n)u, t-u\right) \sum_{j=1}^{\omega_i} \mathbf{1}_{\{r_{j-1}(t-u) \leq s-\rho(n)u < r_j(t-u)\}} \\ &= \sum_{j=1}^{\omega_i} F_i^{(j)}\left(s-\rho(n)u, t-u\right) \\ &= \sum_{j=1}^{\omega_i} F_i^{(j)}\left(s-\rho(n)u, t-u\right) \mathbf{1}_{\{r_{j-1}(t-u) \leq s-\rho(n)u < r_j(t-u)\}}. \end{aligned}$$

The indicator function above verifies the following equality:

$$\mathbf{1}_{\{r_{j-1}(t-u) \leq s-\rho(n)u < r_j(t-u)\}} = \mathbf{1}_{\{\tau_{n,j} < u \leq \tau_{n,j-1}\}}.$$

Now we use the hypothesis: s is in the interval $[r_{j_0-1}t, r_{j_0}t[$. According to the strict monotony of sequence $(r_j)_{j\in E}$, the coefficient $\tau_{n,j-1}$ becomes strictly negative once j over-steps j_0+1. So, we obtain

$$\mathbf{1}_{\{\tau_{n,j} < u \leq \tau_{n,j-1}\}} \mathbf{1}_{\{u>0\}} = 0, \qquad \text{for } j \geq j_0 + 1,$$

and therefore

$$F_i^{(j)}(s - r_n u, t-u) \mathbf{1}_{\{u>0\}} = 0, \qquad \text{for } j \geq j_0 + 1.$$

Hence,

$$F_i\left(s-\rho(n)u, t-u\right) = \sum_{j=1}^{\min(\omega_i, j_0)} F_i^{(j)}\left(s-\rho(n)u, t-u\right) \mathbf{1}_{\{\tau_{n,j} < u \leq \tau_{n,j-1}\}}.$$

So equation (1) becomes as follows:

$$F_n^{j_0}(s,t) = e^{-(\lambda_n/\rho(n))s} + \sum_{i=1}^{n-1} \sum_{j=1}^{\min(\omega_i, j_0)} \int_{\tau_{n,j}}^{\tau_{n,j-1}} e^{-\lambda_n u} \lambda_{n,i} F_i^{(j)}\left(s-\rho(n)u, t-u\right) du. \tag{2}$$

On the other hand, we can prove that $\sum_{i=1}^{n-1} \sum_{j=1}^{\min(\omega_i, j_0)}(\dots) = \sum_{j=1}^{j_0} \sum_{i=\phi_j}^{n-1}(\dots)$. The lemma follows by taking into account the last equality in equation (2). ∎

By considering the initial condition $F_1^{(1)}(s,t) = e^{-\lambda_1 s/r_1} \mathbf{1}_{\{0 \leq s < r_1 t\}}$, we can prove by recurrence that the renewal equation in Lemma 2.1 allows only one solution. Theorem 2.2 will give a simple closed form expression for $F_n^{(j_0)}(s,t)$.

2.2. Solution and Algorithmical Aspects

Before stating the solution of our model, we propose to give some notations

$$\alpha(k,j) = -\frac{\lambda_k - \lambda_j}{\rho(k) - \rho(j)},$$

and

$$\beta(k,j) = \frac{\lambda_k \rho(j) - \lambda_j \rho(k)}{\rho(k) - \rho(j)}, \qquad \text{for all } k, j \text{ such that } \rho(k) \neq \rho(j).$$

We remark that $\alpha(k,0) = -\lambda_k/r_k$ and $\beta(k,0) = 0$ since $\rho(0) = r_0 = 0$ and $\lambda_0 = 0$ (the state 0 is an absorbing state). Without loss of generality, we suppose that 0 is the unique state i which satisfies $\rho(i) = 0$.

Our goal is to evaluate the probability distribution of the performability over $[0,t]$. Theorem 2.2 states a closed form expression for the distribution of Y_t.

THEOREM 2.2. *Under the condition $(\lambda_i - \lambda_j)/(\rho(i) - \rho(j)) \neq (\lambda_k - \lambda_j)/(\rho(k) - \rho(j))$ for all $0 \leq j \leq \phi_{m-1}$ and for all $\phi_{\omega_j+1} < k < i \leq n$, we have: for all $j_0 \in \{1, \dots, m\}$ and for all $s \in [r_{j_0-1}t, r_{j_0}t[$,*

$$F_n^{(j_0)}(s,t) = \sum_{j=0}^{\phi_{j_0}-1} \sum_{k=\phi_{j_0}}^{n} b^{(j)}(n,k) \exp\left(\alpha(k,j)s + \beta(k,j)t\right),$$

where $b^{(j)}(n,k)$ are real numbers given by the following set of recursive expressions: for all $0 \leq j \leq \phi_{j_0} - 1$, we have

for $k = \phi_{j_0}, \dots, n-1$,

$$b^{(j)}(n,k) = \frac{1}{\lambda_n + \rho(n)\alpha(k,j) + \beta(k,j)} \sum_{i=k}^{n-1} \lambda_{n,i} b^{(j)}(i,k),$$

for $k = n$,

$$b^{(j)}(n,n) = \mathbf{1}_{\{j=0\}} + \sum_{l=0}^{\phi_{\omega_j-1}} b^{(l)}(n,j) - \sum_{k=\phi_{\omega_j+1}}^{n-1} b^{(j)}(n,k).$$

PROOF. The condition done on the rate transitions λ_i and the reward rates $\rho(n)$ in the beginning of Theorem 2.2 permits us to have a sense to the fraction $1/(\lambda_n + \rho(n)\alpha(k,j) + \beta(k,j))$. Moreover, according to Lemma 2.1, we have

$$\begin{aligned}
F_n^{(j_0)}(s,t) &= e^{-(\lambda_n/\rho(n))s} + \sum_{j=1}^{j_0} \int_{\tau_{n,j}}^{\tau_{n,j-1}} e^{-\lambda_n u} \sum_{i=\phi_j}^{n-1} \lambda_{n,i} F_i^{(j)}\left(s - \rho(n)u, t-u\right) du \\
&= e^{-(\lambda_n/\rho(n))s} + \sum_{j=1}^{j_0} \int_{\tau_{n,j}}^{\tau_{n,j-1}} e^{-\lambda_n u} \sum_{i=\phi_j}^{n-1} \lambda_{n,i} \\
&\quad \times \left[\sum_{l=0}^{\phi_j-1} \sum_{k=\phi_j}^{i} b^{(l)}(i,k) \exp(\alpha(k,l)\left(s - \rho(n)u\right) + \beta(k,l)(t-u))\right] du.
\end{aligned}$$

The indexes j, i, l, and k verify the following inequation systems:

$$\begin{cases} 1 \leq j \leq j_0 \\ \phi_j \leq i \leq n-1 \\ 0 \leq l \leq \phi_j - 1 \\ \phi_j \leq k \leq i \end{cases} \Leftrightarrow \begin{cases} 0 \leq l \leq \phi_{j_0} - 1 \\ \phi_{\omega_l+1} \leq k \leq n-1 \\ k \leq i \leq n-1 \\ \omega_l + 1 \leq j \leq \inf(j_0, \omega_k) \end{cases}$$

$$\Leftrightarrow \begin{cases} 0 \leq l \leq \phi_{j_0} - 1 \\ \phi_{\omega_l+1} \leq k \leq \phi_{j_0} - 1 \\ k \leq i \leq n-1 \\ \omega_l + 1 \leq j \leq \omega_k \end{cases} \text{ or } \begin{cases} 0 \leq l \leq \phi_{j_0} - 1 \\ \phi_{j_0} \leq k \leq n-1 \\ k \leq i \leq n-1 \\ \omega_l + 1 \leq j \leq j_0. \end{cases}$$

So, the distribution $F_n^{(jo)}(s,t)$ becomes equal to

$$\begin{aligned}
F_n^{(jo)}(s,t) &= \exp\left(\alpha(n,0)s+\beta(n,0)t\right), \text{ since } \alpha(n,0)=\frac{-\lambda_n}{\rho(n)} \text{ and } \beta(n,0)=0 \\
&\quad+\sum_{l=0}^{\phi_{j_O}-1}\sum_{k=\phi_{\omega_l+1}}^{\phi_{j_O}-1}\sum_{i=k}^{n-1}\lambda_{n,i}b^{(l)}(i,k)\exp\left(\alpha(k,l)s+\beta(k,l)t\right) \\
&\quad\times\sum_{j=\omega_l+1}^{\omega_k}\int_{\tau_{n,j}}^{\tau_{n,j-1}}\exp\left[-\left(\lambda_n+\rho(n)\alpha(k,l)+\beta(k,l)\right)u\right]du \\
&\quad+\sum_{l=0}^{\phi_{j_O}-1}\sum_{k=\phi_{j_O}}^{n-1}\sum_{i=k}^{n-1}\lambda_{n,i}b^{(l)}(i,k)\exp\left(\alpha(k,l)s+\beta(k,l)t\right) \\
&\quad\times\sum_{j=\omega_l+1}^{j_O}\int_{\tau_{n,j}}^{\tau_{n,j-1}}\exp\left[-(\lambda_n+\rho(n)\alpha(k,l)+\beta(k,l))u\right]du \\
&=\exp(\alpha(n,0)s+\beta(n,0)t) \\
&\quad+\sum_{l=0}^{\phi_{j_O}-1}\sum_{k=\phi_{\omega_l+1}}^{\phi_{j_O}-1}\sum_{i=k}^{n-1}\lambda_{n,i}b^{(l)}(i,k)\exp\left(\alpha(k,l)s+\beta(k,l)t\right) \\
&\quad\times\int_{\tau_{n,\omega_k}}^{\tau_{n,\omega_l}}\exp\left[-(\lambda_n+\rho(n)\alpha(k,l)+\beta(k,l))u\right]du \\
&\quad+\sum_{l=0}^{\phi_{j_O}-1}\sum_{k=\phi_{j_O}}^{n-1}\sum_{i=k}^{n-1}\lambda_{n,i}b^{(l)}(i,k)\exp\left(\alpha(k,l)s+\beta(k,l)t\right) \\
&\quad\times\int_{0}^{\tau_{n,\omega_l}}\exp\left[-(\lambda_n+\rho(n)\alpha(k,l)+\beta(k,l))u\right]du, \qquad \text{since } \tau_{n,j_O}=0.
\end{aligned}$$

On the other hand, we can easily prove that

$$\alpha(k,l)s+\beta(k,l)t-\left(\lambda_m+r_m\alpha(k,l)+\beta(k,l)\right)\tau_{m,k}=\alpha(m,k)s+\beta(m,k)t$$

and

$$\alpha(k,l)s+\beta(k,l)t-\left(\lambda_m+r_m\alpha(k,l)+\beta(k,l)\right)\tau_{m,l}=\alpha(m,l)s+\beta(m,l)t.$$

By taking into account of these two equalities in the last expression of $F_n^{(jo)}(s,t)$, we obtain

$$\begin{aligned}
F_n^{(jo)}(s,t) &= \exp\left(\alpha(n,0)s+\beta(n,0)t\right) \\
&\quad+\sum_{l=0}^{\phi_{j_O}-1}\sum_{k=\phi_{\omega_l+1}}^{\phi_{j_O}-1}\sum_{i=k}^{n-1}\frac{\lambda_{n,i}}{\lambda_n+\rho(n)\alpha(k,l)+\beta(k,l)}b^{(l)}(i,k)\exp(\alpha\,(n,k)s+\beta(n,k)t) \\
&\quad-\sum_{l=0}^{\phi_{j_O}-1}\sum_{k=\phi_{\omega_l+1}}^{n-1}\sum_{i=k}^{n-1}\frac{\lambda_{n,i}}{\lambda_n+\rho(n)\alpha(k,l)+\beta(k,l)}b^{(l)}(i,k)\exp\left(\alpha(n,l)s+\beta(n,l)t\right) \\
&\quad+\sum_{l=0}^{\phi_{j_O}-1}\sum_{k=\phi_{j_O}}^{n-1}\sum_{i=k}^{n-1}\frac{\lambda_{n,i}}{\lambda_n+\rho(n)\alpha(k,l)+\beta(k,l)}b^{(l)}(i,k)\exp\left(\alpha(k,l)s+\beta(k,l)t\right).
\end{aligned}$$

Moreover, it is obvious that $\sum_{l=0}^{\phi_{j_O}-1}\sum_{k=\phi_{\omega_l+1}}^{\phi_{j_O}-1}(\dots)=\sum_{k=1}^{\phi_{j_O}-1}\sum_{l=0}^{\phi_{\omega_k}-1}(\dots)$. Therefore, the func tion $F_n^{(jo)}(s,t)$ will be equal to

$$F_n^{(jo)}(s,t) = \exp\left(\alpha(n,0)s + \beta(n,0)t\right)$$
$$+ \sum_{l=0}^{\phi_{jo}-1} \sum_{k=\phi_{jo}}^{n-1} \left[\sum_{i=k}^{n-1} \frac{\lambda_{n,i}}{\lambda_n + \rho(n)\alpha(k,l) + \beta(k,l)} b^{(l)}(i,k)\right] \exp\left(\alpha(k,l)s + \beta(k,l)t\right)$$
$$+ \sum_{k=1}^{\phi_{jo}-1} \left[\sum_{l=0}^{\phi_{\omega_k}-1} \sum_{i=k}^{n-1} \frac{\lambda_{n,i}}{\lambda_n + \rho(n)\alpha(k,l) + \beta(k,l)} b^{(l)}(i,k)\right] \exp\left(\alpha(n,k)s + \beta(n,k)t\right)$$
$$- \sum_{l=0}^{\phi_{jo}-1} \left[\sum_{k=\phi_{\omega_l}+1}^{n-1} \sum_{i=k}^{n-1} \frac{\lambda_{n,i}}{\lambda_n + \rho(n)\alpha(k,l) + \beta(k,l)} b^{(l)}(i,k)\right] \exp\left(\alpha(n,l)s + \beta(n,l)t\right).$$

If we identify now this relation with the expression of $F_n^{(jo)}(s,t)$ in Theorem 2.2, we get the following recurrent expressions for the coefficients $b^{(j)}(n,k)$:

for $k = \phi_{jo}, \dots, n-1$,

$$b^{(j)}(n,k) = \frac{1}{\lambda_n + \rho(n)\alpha(k,j) + \beta(k,j)} \sum_{i=k}^{n-1} \lambda_{n,i} b^{(j)}(i,k),$$

for $k = n$,

$$b^{(j)}(n,n) = \mathbf{1}_{\{j=0\}} + \sum_{l=0}^{\phi_{\omega_j}-1} \frac{1}{\lambda_n + \rho(n)\alpha(j,l) + \beta(j,l)} \sum_{i=j}^{n-1} \lambda_{n,i} b^{(l)}(i,j)$$
$$- \sum_{k=\phi_{\omega_j}+1}^{n-1} \frac{1}{\lambda_n + \rho(n)\alpha(k,j) + \beta(k,j)} \sum_{i=k}^{n-1} \lambda_{n,i} b^{(l)}(i,k),$$

which is equivalent to

$$b^{(j)}(n,n) = \mathbf{1}_{\{j=0\}} + \sum_{l=0}^{\phi_{\omega_j}-1} b^{(l)}(n,j) - \sum_{k=\phi_{\omega_j}+1}^{m-1} b^{(j)}(n,k).$$

The proof of this theorem is then completed. ∎

In case the system is degradable homogeneous: when there is only one possible transition out of any state i, namely, to state $i-1$ (i.e., it contains $d = 1$ type of component), the above recurrence formulas will be simplified since $\lambda_{il} = 0$ for $l < i-1$. So, the sequence $b^{(j)}(n,k)$ becomes as follows:

$$b^{(j)}(n,k) = \frac{\lambda_{n,n-1}}{\lambda_{n,n-1} + \rho(n)\alpha(k,j) + \beta(k,j)} b^{(j)}(n-1,k), \qquad \text{for } \phi_j \le k \le n-1,$$

and

$$b^{(j)}(n,n) = \mathbf{1}_{\{j=0\}} + \sum_{l=0}^{\phi_{\omega_j}-1} b^{(l)}(n,j) - \sum_{k=\phi_{\omega_j}+1}^{n-1} b^{(j)}(n,k).$$

We observe that these formulas are the same obtained by Goyal and Tantawi. However, their generalization to nonhomogeneous systems is not very clear. Furthermore, the coefficients which they evaluate by recurrence depend on four indexes instead of three indexes, which generates a greater computational cost.

The computation of the distribution of Y_t mainly involves computing the coefficients $b^{(j)}(i,k)$. Since indexes j, i, and k vary, respectively, in $\{0,\dots,\phi_{j_0}-1\}$, $\{1,\dots,n\}$, and the set $\{\phi_{\omega_j+1},\dots,i-1\}$, the total number of required terms is

$$\sum_{j=0}^{\phi_{j_0}-1} \frac{(n-\phi_{\omega_j+1}+1)(n-\phi_{\omega_j+1}+2)}{2} \le \sum_{j=0}^{\phi_{j_0}-1} \frac{(n-j)(n-j+1)}{2} = \frac{n\phi_{j_0}(n-\phi_{j_0}+2)}{2} + \frac{\phi_{j_0}(\phi_{j_0}-1)(\phi_{j_0}-2)}{6}.$$

Moreover, the computation of each vector $b^{(j)}(i,k)$, $k \neq n$, requires at most $O(d)$ operations (see equations in Theorem 2.2). Therefore, the total computational effort to evaluate the distribution of performability for this kind of model is

$$O\left(d\phi_{j_0}\left(\frac{n(n-\phi_{j_0}+2)}{2} + \frac{(\phi_{j_0}-1)(\phi_{j_0}-2)}{6}\right)\right).$$

Theorem 2.2 gives, for instance, a new method to compute the distribution of performability for acyclic homogeneous Markov models. The main advantage supplied by our approach, in comparison to Goyal and Tantawi's method, is that the computational complexity of our algorithm depends on the value of s. The total effort is as low as the value of s is close to $r_1 t$ (s "close to" $r_1 t \Leftrightarrow j_0$ "small"). On the other hand, the complexity of the method developed in [18] is $O(d(n+1)^3)$. The $b^{(j)}(n,k)$ recurrence depends only on three indexes, however in [18], the authors proposed a set of recurrence formulas based on four indexes. Moreover, these formulas, which come from the homogeneous case (only transitions between state i and state $i-1$ are allowed), are not very clear when they are generalized to heterogeneous systems. This fact prevented us from comparing the time complexity of each method.

3. UNIFORM ACYCLIC MODELS

The method established in the previous section is valid under a certain condition between transition rates λ_i and reward rates r_i. The condition $(\lambda_i - \lambda_j)/(\rho(i)-\rho(j)) \neq (\lambda_k - \lambda_j)/(\rho(k)-\rho(j))$ was necessary[1] to define the coefficients $b^{(j)}(i,k)$. This property led us to study the performability measure for acyclic models satisfying the condition $(\lambda_i - \lambda_j)/(\rho(i)-\rho(j)) = (\lambda_k - \lambda_j)/(\rho(k)-\rho(j))$. We limit our study to the case where two different states cannot have the same reward rate.[2] These two conditions are equivalent to $(\lambda_i/r_i) = (\lambda_j/r_j)$ and $\rho(i) = r_i$ for all $i,j \in E\backslash\{0\}$.

DEFINITION 3.1. *An acyclic performability model is said to be uniform if we have*

$$\frac{\lambda_i}{\rho(i)} = \frac{\lambda_j}{\rho(j)}, \qquad \text{for all } i,j \in E\backslash\{0\}.$$

The state 0 is the absorbing state.

Note that $\rho(i)/\lambda_i$ represents the mean reward in the state i. Therefore, the "uniform" condition means that the mean reward for all states is equal.

3.1. Model Solution

In this section, we present a closed form expression for $F_m^{(j_0)}(s,t)$ for this kind of model. The solution follows from the renewal equation established in Lemma 2.1. It appears as a finite sum of polynomial functions affected by coefficients which are evaluated by recurrence.

[1]The denominator of the fraction $1/[\lambda_i + \rho(i)\alpha(k,j) + \beta(k,j)]$ is not equal to zero if and only if $(\lambda_i - \lambda_j)/(\rho(i)-\rho(j)) \neq (\lambda_k - \lambda_j)/(\rho(k)-\rho(j))$.

[2]For the general case (i.e., $\exists\, i \neq j \in \{0,1,\dots,n\}$ such that $\rho(i) = \rho(j)$), see [16].

THEOREM 3.2. *For all $j_0 \in \{1,\dots,m\}$ and all $s \in [r_{j_0-1}t, r_{j_0}t[$, we have*

$$F_m^{(j_0)}(s,t) = \exp\left(-\lambda_m \frac{s}{r_m}\right)\left[\sum_{k=0}^{m-j_0} a^{(j_0)}(m,k)\frac{(s/r_1)^k}{k!}\right.$$
$$+ \sum_{l=1}^{j_0-1}\sum_{k=1}^{m-1} c_l^{(j_0)}(m,k)\frac{((r_m t - s)/(r_m - r_l))^k}{k!}$$
$$\left.+ \sum_{l=1}^{j_0-1}\sum_{j=j_0}^{m-1}\sum_{n=2}^{m-1}\sum_{k=1}^{n} b_m^{j,l}(n,k)\frac{(s - r_l t)/(r_j - r_l)^k (r_j t - s)/(r_j - r_l)^{n-k}}{k!(n-k)!}\right].$$

The coefficients $a^{(j_0)}(m,k)$, $c_l^{(j_0)}(m,k)$, and $b_m^{j,l}(n,k)$ are real numbers given by the following set of recursive expressions:

for $k = 1,\dots,m$,

$$a^{(j_0)}(m,0) = 1,$$
$$a^{(j_0)}(m,k) = \frac{r_1}{r_m}\sum_{i=k+j_0-1}^{m-1} \lambda_{mi} a^{(j_0)}(i,k-1),$$

for $l = 1,\dots,j_0-1$, $j = l+2,\dots,j_0$, **and** $k = 2,\dots,m-1$,

$$c_l^{(j)}(m,1) = \frac{r_l}{r_m}\lambda_{ml},$$
$$c_l^{(l+1)}(m,k) = \frac{r_l}{r_m}\sum_{i=k+l-1}^{m-1} \lambda_{mi} c_l^{(l+1)}(i,k-1),$$
$$c_l^{(j)}(m,k) = c_l^{(j-1)}(m,k) + \left(\frac{r_m - r_l}{r_m - r_j}\right)^k b_m^{j-1,l}(k,k),$$

for $n = 2,\dots,m-1$ **and** $k = 2,\dots,n$,

$$b_m^{j,l}(n,1) = \lambda_{mj}\frac{r_j - r_l}{r_m - r_l}\left[c_l^{(l+1)}(j,n-1) + \sum_{h=1}^{l-1} b_j^{l,h}(n-1,n-1)\right],$$
$$b_m^{j,l}(n,k) = \frac{r_m - r_j}{r_m - r_l} b_m^{j,l}(n,k-1) + \frac{r_j - r_l}{r_m - r_l}\sum_{i=\max(n,j+1)}^{m-1} \lambda_{mi} b_i^{j,l}(n-1,k-1).$$

PROOF. (The proof of this theorem is too long. For more details, see [19].) ∎

We assume here that a sum $\sum_{p=x}^{y}(\dots)$ is equal to zero if $x > y$. So if, for example, $j_0 = 1$, $F_m^{(1)}(s,t)$ mainly involves computing the coefficients $a^{(1)}(m,k)$, for $k = 0,\dots,m-1$. The number of cells $a^{(j_0)}(m,k)$, $c_l^{(j_0)}(m,k)$, and $b_i^{j,l}(n,k)$ required in this method depends on j_0 and therefore on the value of s. Moreover, according to the strict monotonicity of the sequence $(r_i)_{i\in E}$, it is obvious that the three sequences $a^{(j_0)}(m,k)$, $c_l^{(j_0)}(m,k)$, and $b_i^{j,l}(n,k)$ are positive.

4. NUMERICAL EXAMPLE

In this section, we present an application of the previous algorithm, related to acyclic uniform models, to a simple degradable computer system. It consists of a multiprocessor with n processors, each is subject to a random failure exponentially distributed with rate $\lambda = 10^{-6}$/sec. We suppose that each processor is self-testing, and in the presence of a single faulty processor, the system is able to recover, with a coverage rate $c = 0.999$, to an $n-1$ processor configuration, provided that

$n \geq 2$. The coverage rate is the probability that the system successfully recover after a processor failure. When the system performs with a single processor (i.e., the $n-1$ remaining processors are down), fault recovery is no longer possible. The system is considered to be down when no processor is available. Our goal is to calculate the distribution of the average number of available processors over a given mission time $[0, t]$.

The Markov process which describes the behavior of the system is shown in Figure 1 when the number of processors is $n = 3$. Each state $0 \leq i \leq n$ corresponds to the number of available processors in the system. The failure rate associated to each state i is $i\lambda$, which is decomposed into $ic\lambda$, when the recovery is well functioning, and $i(1-c)\lambda$, when not. Since we are interested in the average number of available processors, we take the function $r_i = i$ as reward rate. This model still satisfies the condition $\lambda_i/r_i = \lambda_j/r_j$. In fact, for all $i \in \{1, \dots, n\}$, we have

$$\frac{\lambda_i}{r_i} = \frac{ic\lambda + i(1-c)\lambda}{r_i} = \frac{i\lambda}{r_i} = \frac{i\lambda}{i} = \lambda.$$

The accumulated reward averaged over t (i.e., Y_t/t) represents, therefore, the average number of available processors over the mission time $[0, t]$.

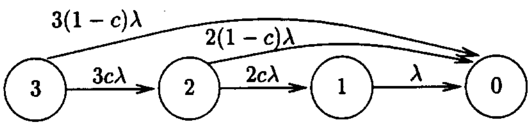

Figure 1. Transition rate graph.

Figure 2 shows the probability that the average number of available processors over $[0, t]$ is greater than 85% of the number of processors in the system. These curves are shown for many values of t and n.

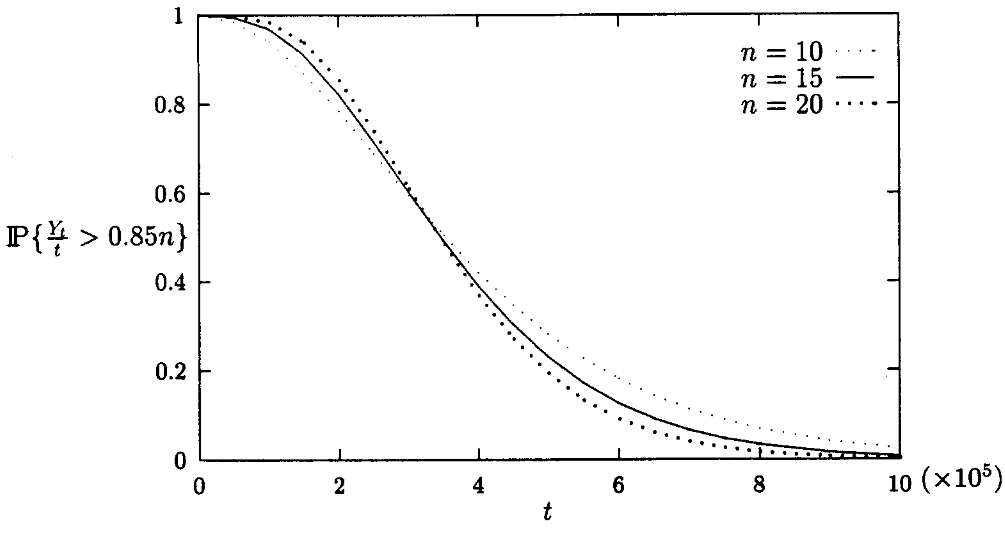

Figure 2. Distribution of Y_t/t.

We observe from the three curves of Figure 2 that the probability $\mathbb{P}\{Y_t/t > 0.85n\}$ increases with respect to n for $t < 300000$ sec and decreases with respect to n for $t > 350000$ sec. That means that, for $n = 10, 15, 20$, the probability of the average number of available processors in the system over $[0, t]$ increases with n so long as t is less than 3×10^5 sec, and conversely for $t > 3.5 \times 10^5$ sec. For $n = 10$, we obtain $j_0 = n - 1$ and for $n = 15, 20$, we obtain $j_0 = n - 2$.

5. CONCLUSION

We have presented, in this paper, two methods to evaluate the performability distribution for degradable computer systems. The first method follows the same approach developed by Goyal and Tantawi [18] and leads to a new algorithm with a low polynomial computational complexity. Its main advantage is a computational complexity which depends on the minimum expected accomplishment level.[3] The second method concerns a new class of acyclic models which we call uniform acyclic models. This method leads to a simple closed form for the transient distribution of performability.

REFERENCES

1. J.F. Meyer, On evaluating the performability of degradable computing systems, *IEEE Transactions on Computers* **C-29** (8), 720–731, (August 1980).
2. E. de Souza e Silva and H.R. Gail, Calculating cumulative operational time distributions of repairable computer systems, *IEEE Transactions on Computers* **C-35** (4), 322–332, (April 1986).
3. A. Jensen, Markoff chains as an aid in the study of Markoff processes, *Skandinavsk Aktuarietidskrift* **36**, (1953).
4. G. Rubino and B. Sericola, Interval availability distibution computation, In *Proceedings IEEE 23rd Fault-Tolerant Computing Symposium*, pp. 49–55, Toulouse, France, (June 1993).
5. G. Rubino and B. Sericola, Interval availability analysis using denumbrable Markov processes: Application to multiprocessor subject to breakdowns and repair, *IEEE Transactions on Computers* **44** (2), 286–294, (February 1995).
6. A. Csenki, Transient analysis of interval availability for repairable sytems modelled by finite semi-Markov processes, *IMA Journal of Mathematics Applied in Business and Industry* **6**, 267–281, (1995).
7. A. Csenki, A dependability measure for Markov models of repairable systems: Solution by randomization and computational experience, *Computers Math. Applic.* **30** (2), 95–110, (1995).
8. A. Goyal and A.N. Tantawi, A measure of guaranteed availability and its numerical evaluation, *IEEE Transactions on Computers* **C-37** (1), 25–32, (January 1988).
9. A. Csenki, Cumulative operational time analysis of finite semi-Markov reliability models, *Reliability Engineering and System Safety* **44**, 17–25, (1994).
10. J.F. Meyer, Closed-form solutions of performability, *IEEE Transactions on Computers* **C-31** (7), 648–657, (July 1982).
11. D.G. Furchtgott, Performability models solutions, Ph.D. Thesis, University of Michigan, (January 1984).
12. J.F. Meyer, D.G. Furchtgott and L.T. Wu, Performability evaluation of the SIFT computer, *IEEE Transactions on Computers* **C-29** (6), 501–509, (June 1980).
13. M.D. Beaudry, Performance-related reliability measures for computing systems, *IEEE Transactions on Computers* **C-27**, 540–547, (June 1978).
14. G. Ciardo, R. Marie, B. Sericola and K.S. Trivedi, Performability analysis using semi-Markov reward processes, *IEEE Transactions on Computers* **C-39**, 1251–1264, (October 1990).
15. B.R. Iyer, L. Donatiello and P. Heidelberger, Analysis of performability for stochastic models of fault-tolerant systems, *IEEE Transactions on Computers* **C-35** (10), 902–907, (October 1986).
16. H. Nabli and B. Sericola, Performability analysis for degradable computer systems, Technical Report 2602, INRIA, Campus de Beaulieu, Rennes, France, July 1995, `ftp://ftp.inria.fr/INRIA/tech-reports/publi-ps-gz/RR/RR-2602.ps.gz`.
17. H. Nabli and B. Sericola, Performability analysis: A new algorithm, *IEEE Transactions on Computers* **45** (4), 491–494, (April 1996).
18. A. Goyal and A.N. Tantawi, Evaluation of performability for degradable computer systems, *IEEE Transactions on Computers* **C-36** (6), 738–744, (June 1987).
19. H. Nabli, Mesure de performabilité sur des processus de Markov homogènes à espace d'états fini, Thèse de Doctorat, *Université de Rennes I* **1378**, 136–150, (September 1995); `http://www.irisa.fr/model/theses.html`.

[3]In this paper, the minimum expected accomplishment level is denoted by s.