Performability analysis using semi-markov reward process

Gianfranco Ciardo, Raymond Marie, Bruno Sericola, Kishor Strivedi

INRIA

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France

Tél. (1) 39 63 55 11

Rapports de Recherche

N° 859

PERFORMABILITY ANALYSIS USING SEMI-MARKOV REWARD PROCESS

Gianfranco CIARDO
Raymond MARIE
Bruno SERICOLA
Kishor S. TRIVEDI

JUIN 1988

Performability analysis using semi-Markov reward process

Gianfranco Ciardo †, Raymond Marie ‡, Bruno Sericola ‡, Kishor S. Trivedi †

†Duke University - Durham, NC 27706, USA

‡IRISA Campus de Beaulieu - 35042 Rennes Cedex, FRANCE

May 9, 1988
Publication Interne n° 408
38 pages

abstract

With the increasing complexity of multiprocessor and distributed systems, the need to develop efficient and accurate modelling method is evident. Fault-tolerance and degradable performance of such systems has given rise to considerable interest in models for the combined evaluation of performance and reliability [1,2]. Most of these models are based upon Markov or semi-Markov reward processes. Beaudry [1] proposed a simple method for computing the distribution of performability in a Markov reward process. We present two extensions of Beaudry's approach. First, we generalize the method to a semi-Markov reward process. Second, we remove this restriction requiring the association of zero reward to absorbing states only. Such reward models can be used to evaluate the effectiveness of degradable fault-tolerant systems. We illustrate the use of this approch with three interesting applications.

Index Terms – Computer performance, computer reliability, graceful degradation, Markov models, semi-Markov models, reward processes.

---

[0]This work was supported in part by the U.S. Air Force Office of Scientific Research under Grant AFSOR-84-0132 and by the NASA Langley Research Center under Grant NAG-1-70.

[1]This work also appeared as a Duke Computer Science Technical Report CS-1988-9.

Analyse de performabilité par les processus semi-markoviens à taux de récompense

Résumé

Avec la complexité croissante des systèmes distribués et des systèmes multiprocesseurs, le besoin de developper des méthodes de modélisation efficaces et précises est évident. La tolérance aux pannes et la dégradation des performances de tels systèmes ont engendré un intérêt considérable dans les modèles permettant l'évaluation combinée des aspects de performance et de fiabilité [1,2]. La plupart de ces modèles sont basés sur les processus markoviens ou semi-markoviens à taux de récompense. Beaudry [1] proposa une méthode pour calculer la distribution de la performabilité dans un processus markovien à taux de récompense. On présente deux extensions de l'approche de Beaudry. D'abord, on généralise la méthode à un processus semi-markovien à taux de récompense. Puis, on lève la restriction consistant à associer des taux de récompense nuls seulement aux états absorbants. De tels modèles à taux de récompense peuvent être utilisés pour évaluer l'efficacité des systèmes dégradables tolérant les pannes. Enfin, l'utilisation de cette approche est illustrée par trois applications intéressantes.

Mots-clés – Performance des systèmes, fiabilité des systèmes, performabilité, modèles markoviens, modèles semi-markoviens, processus à taux de récompense.

1 Introduction

Multiprocessors and distributed processing systems can provide higher performance and higher reliability/availability over single-processor systems. In order to properly assess the effectiveness of such systems, measures that combine performance and reliability are needed. For the purpose, Meyer [2] developed a conceptual framework of *performability*. Markov and semi-Markov reward models are used in the performability evaluation of computer or communication systems.

This paper presents a new algorithm for the computation of the distribution of accumulated reward until absorption in a semi-Markov reward process. Failures and repairs of system resources are modeled by a semi-Markov process, called the structure-state process [2]. A performance level, or reward rate, is associated with each state of the structure-state process. The resulting semi-Markov reward process is then able to capture not only failure and repair of system components, but degradable performance as well. Application examples used to illustrate our method are fairly diverse and interesting. The first example is a repairable system with a limited number of repairs [3]. The second example is the M/M/2 queueing system subject to failure and repair. The final example is based on a measurement-based performability model of a large IBM system [4].

In [1], Beaudry proposed an algorithm for computing the accumulated reward until absorption in a continuous-time Markov chain. We extend her result in two directions. First, we consider semi-Markov reward processes, thus removing the restriction of exponentially distributed sojourn times. We have observed in [4] that the use of an exponential distribution as an approximation to a non-exponential distribution can cause considerable errors in the results. Second, we allow non-absorbing states with zero reward rate, which do not occur in practical models [4]. In Beaudry's method, these states would have caused a "division by zero" in the computations or would have to required to approximate their zero reward rate using a small positive quantity instead, but we caution the reader that the numérical errors due to such an approximation can be large (as will be shown in one of the examples of Section 6).

In Section 2, the problem of computing the distribution of accumulated reward for a semi-Markov process is introduced; in Section 3, an informal approach to the solution of this problem is given; in Section 4, a transformation is defined, to obtain a new semi-Markov process whose lifetime is distributed as the accumulated reward of the original process. The efficient construction

of the new semi-Markov process is discussed in Section 5, while examples of the application of our method to the modeling of computer systems are presented in Section 6. Section 7 contains some considerations for further extensions. Proof of important results are given in the Appendix.

2 The semi-Markov Reward Process

Assume that $\{X(t), t \geq 0\}$ is a right continuous semi-Markov process with state space $S \subseteq \mathbb{N} = \{0, 1, 2, \ldots\}$ and with probability of being eventually absorbed equal to 1. Let T_k be the time of the k-th transition ($T_0 = 0$) and define $V_k = T_{k+1} - T_k$, the sojourn time in the $(k+1)$-th visited state ($k \in \mathbb{N}$). Let $X_k = X(T_k)$, the state reached after the k-th transition. Let S_A denote the set of absorbing states. Define the kernel, the transition probability matrix, and the initial probability vector respectively as

$$Q(t) = [Q_{i,j}(t)] = [pr\{X_{k+1} = j, V_k \leq t | X_k = i\}]$$

$$P = [P_{i,j}] = [pr\{X_{k+1} = j | X_k = i\}] = [Q_{i,j}(\infty)] = Q(\infty)$$

$$\underline{\pi} = [\pi_i] = [pr\{X(0) = i\}]$$

Assuming for simplicity that: $\forall i \in S_A$, $Q_{i,i}(t) = \delta(t-1)$, where

$$\delta(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

and note that $\forall i \in S_A$, $Q_{i,j} = 0$ for $i \neq j$.

$\{X_k, k \in \mathbb{N}\}$ is a discrete-time Markov chain over the state space S with transition matrix probability P (see [5]). Eventually the process gets absorbed, so with probability 1 there exists a finite index K such that $\forall k' > K \quad X_{k'} = X_K$.

A real-valued reward rate r_i is associated to each state $i \in S$. Assume that the reward rates are non-negative, and, in particular, zero for the absorbing states:

$$\forall i \in S \quad (\, r_i \in \mathbb{R},\, r_i \geq 0\,) \;\wedge\; (\, i \in S_A \Rightarrow r_i = 0\,).$$

Define the accumulated reward earned up to time t by

$$Y(t) = \int_0^t r_{X(\tau)} d\tau.$$

The measure considered in this paper are the distribution of $Y(\infty)$, representing the accumulated reward up to absorption, conditioned on the initial state i:

$$C_i(x) = pr\{Y(\infty) \leq x \mid X_0 = i\} = pr\left\{\int_0^\infty r_{X(t)}dt \leq x \mid X_0 = i\right\}.$$

The problem of computing the distribution of $Y(t)$ for a finite t in a Markov reward process is considered elsewhere [6]. The distribution of $Y(t)$ in a semi-Markov reward process is discussed in [7,8].

The following semi-Markov reward process will be used as a running example throughout the paper. Errors arise in a system (initially in an *up* state) according to a Poisson process with rate λ. When an error is detected, the system goes to a *recover* state, where it tries to handle the error. The error handling time is assumed to be a given constant T. If a second error arises before the recovery completes (within T time units from the occurrence of a previous error), the system fails, going to a *down* state. This event has probability $1 - c = 1 - e^{-\lambda T}$. Figure 1 shows the transitions between states. Using the given definitions, $S = \{up, recover, down\}$, $S_A = \{down\}$, $\underline{\pi} = [1, 0, 0]$, and

$$Q(t) = \begin{bmatrix} 0 & 1 - e^{-\lambda t} & 0 \\ c\,\delta(t-T) & 0 & 1 - e^{-\lambda \min\{t,T\}} \\ 0 & 0 & \delta(t-1) \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 1 & 0 \\ c & 0 & (1-c) \\ 0 & 0 & 1 \end{bmatrix}$$

The reward rates are $r_{up} = 1$, $r_{recover} = \gamma$, $r_{down} = 0$, where $0 \leq \gamma \leq 1$. If $\gamma = 0$, then $Y(\infty)$ is the total system uptime until failure and hence $C(x)$ will be the distribution of system uptime. If $\gamma = 1$, then $Y(\infty)$ will be the time to system failure.

3 A Change of Pace

The stochastic process $\{X(t), t \geq 0\}$ defined in the previous section is a semi-Markov reward process, since reward rates are associated with the state of the semi-Markov process. A method

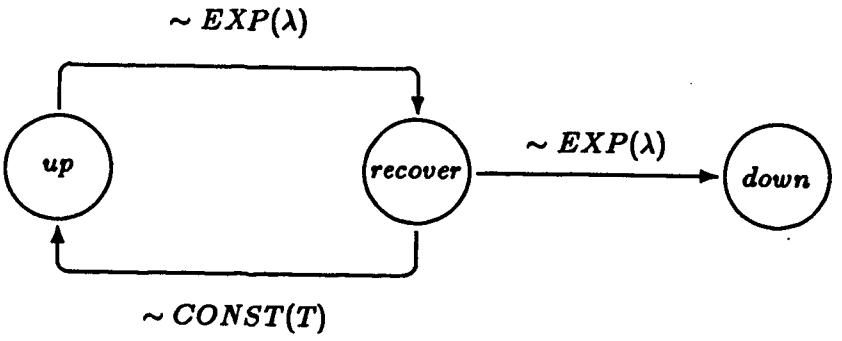


Figure 1: A simple example

of computing the distribution of accumulated reward until absorption when $\{X(t), t \geq 0\}$ is a Markov chain was given in [1], where transient states were restricted to strictly positive reward rates. More recently, in [9], this restriction is removed. A similar approach can be used in the semi-Markov setting. Beaudry's idea in [1] is to define a new Markov chain $\{\hat{X}(x), x \geq 0\}$ by dividing the transition rates out of state i by r_i, so that the distribution of time to absorption of the new process is the same as that of $Y(\infty)$ in the original process. This is true because the transformation takes into account the earned reward in the "time" index of the stochastic process. The sojourn in state i is slowed down or accelerated according to the value of r_i, being larger or smaller than 1. Then, for state i, sojourn times of length τ in $\{X(t), t \geq 0\}$ and of length τr_i in $\{\hat{X}(x), x \geq 0\}$ are equivalent. Another way of thinking about this transformation is a substitution of the index variable, from time to reward: a sojourn of t time units corresponds to a gain of tr_i reward units in state i (and a gain of x reward units in state i corresponds to a sojourn of x/r_i time units). Figure 2 illustrates this correspondence.

Following the example of figure 1, the sojourn time in state *up* is left unchanged, and so is the one in state *down* (which is irrelevant since state *down* is absorbing), but the sojourn time in state *recover* is changed. The transition toward state *down* now has rate λ/γ, and a successful recovery takes $T\gamma$ "time" (or, better, accrues $T\gamma$ reward).

There is a difficulty with the above approach: the reward rate of a transient state could be zero. No reward is earned during the sojourn in such a state, however the system will eventually move either to another transient state, possibly with positive reward rate, or to an absorbing state.

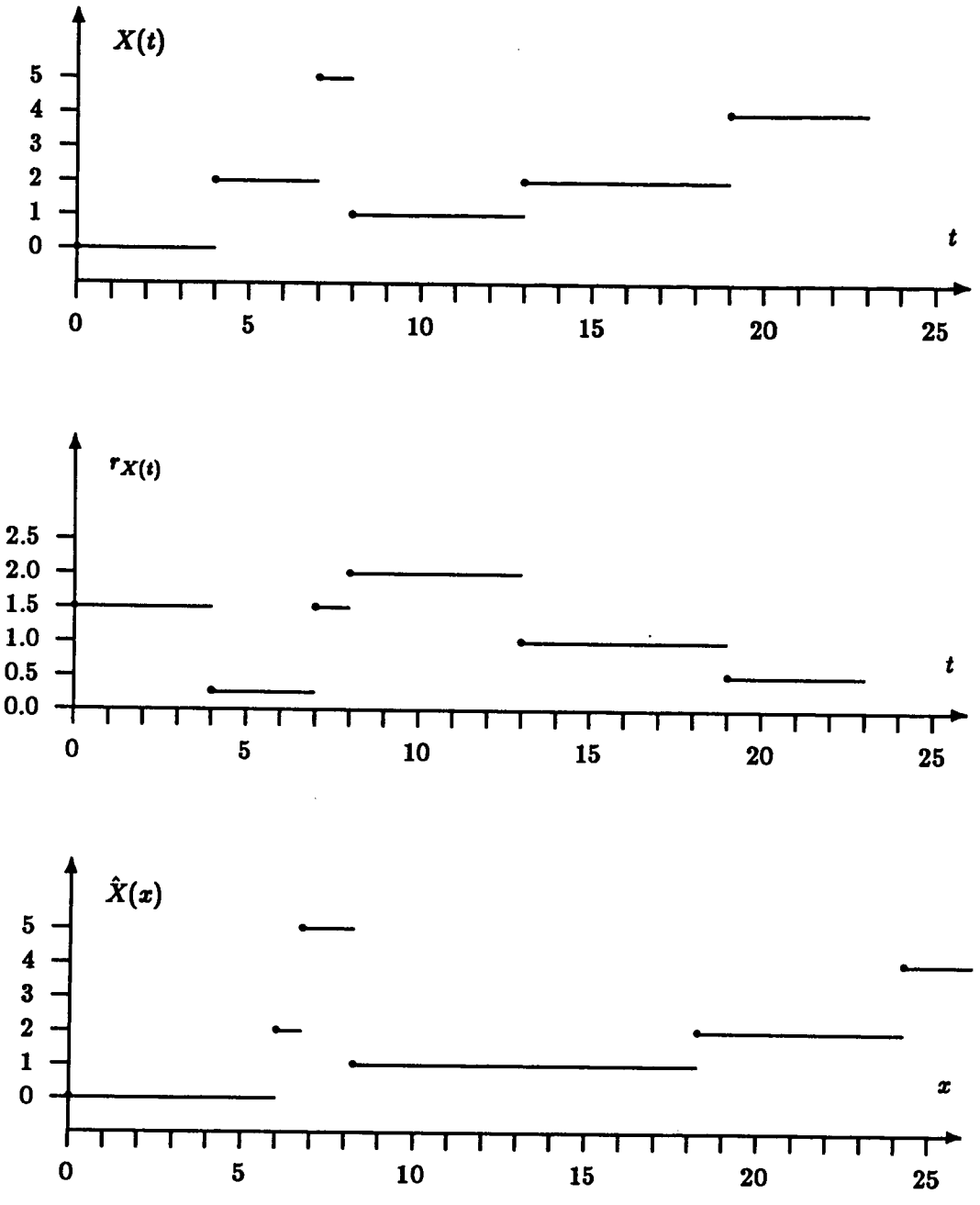


Figure 2: Transformation on the index variable

Intuitively, zero reward rates correspond to instantaneous transitions in $\{\hat{X}(x), x \geq 0\}$, since it spends no time ($\{X(t), t \geq 0\}$ gains no reward) in transient states with zero reward rate. Kulkarni [9] showed how to eliminate these states if $\{X(t), t \geq 0\}$ is a Markov chain. A similar approach had already been used in the Generalized Stochastic Petri Net (GSPN) framework [10], where infinite rates arise directly from the definition, not from the presence of zero reward rates. In the GSPN, "timed" transitions represent activities whose time to completion is exponentially distributed, while "immediate" transitions represent logic events taking no time. The markings (states) of the GSPN are partitioned into "tangible" and "vanishing", the latter corresponding to markings where at least one immediate transition is enabled. Though vanishing markings may describe logical (but instantaneous) conditions of relevance, they must be eliminated before the underlying stochastic process can be studied: a GSPN is hence transformed into a smaller Markov chain whose states are the tangible markings only.

Yet another related effort is reported in [11], where groups of "fast" states are approximated to be groups of vanishing states. In [12], the distribution of sojourn time until absorption for a given subset of states in a Markov process is calculated.

In figure 1, state *recover* becomes vanishing if γ is 0. This would amount to saying that no useful work is performed when a recovery is under way, a reasonable assumption.

4 The Definition of the New Process

In this section a new semi-Markov process $\{\hat{X}(x), x \geq 0\}$ is defined, whose state space is a subset of S, and whose time to absorption has the same distribution as $Y(\infty)$. This is done by eliminating the vanishing states, and by scaling the sojourn time in each state.

Let S_0 be the set of transient states with zero reward rates, S_T the set of transient states with positive reward rates, and define $S_1 = S_T \cup S_A = S \setminus S_0$, so that $S = S_0 \cup S_T \cup S_A = S_0 \cup S_1$. If the state are reordered according to these subsets, the one-step transition probability matrix P and the initial probability vector $\underline{\pi}$ can be partitioned as

$$P = \begin{bmatrix} p[00] & p[01] \\ p[10] & p[11] \end{bmatrix} = \begin{bmatrix} p[00] & p[0T] & p[0A] \\ p[T0] & p[TT] & p[TA] \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{\pi} = \left[\underline{\pi}^{[0]}, \underline{\pi}^{[1]}\right] = \left[\underline{\pi}^{[0]}, \underline{\pi}^{[T]}, \underline{\pi}^{[A]}\right]$$

The quantity

$$M = [M_{i,j}] = (I - P^{[00]})^{-1} P^{[01]} = P^{[01]} + P^{[00]}P^{[01]} + P^{[00]^2}P^{[01]} + \ldots =$$

$$\left[M^{[0T]}, M^{0A]}\right] = \left[(I - P^{[00]})^{-1} P^{[0T]}, (I - P^{[00]})^{-1} P^{[0A]}\right]$$

will be often used in the following derivations. $(I - P^{[00]})^{-1}$ is the fundamental matrix of the discrete-time Markov chain $\{X_k, k \in \mathbb{N}\}$ when all the states in S_1 are considered absorbing. Hence the entries of M satisfy [13]:

$$\forall i \in S_0, \ \ \forall j \in S_1, \quad M_{i,j} = \sum_{k=1}^{\infty} pr\{X_k = j, \forall n \ \ 0 < n < k \ \ X_n \in S_0 \mid X_0 = i\}$$

that is, $M_{i,j}$ is the probability that the first state visited in S_1 will be j, after an arbitrary number of transitions within S_0, given that the process $\{X_k, k \in \mathbb{N}\}$ is in state $i \in S_0$.

The state space of the new process $\{\hat{X}(x), x \geq 0\}$ is $\hat{S} = S_1$. Let $\hat{T}_i$ be the "time" of the i^{th} transition for $\{\hat{X}(x), x \geq 0\}$ and let $\hat{T}_0 = 0$, $\hat{V}_i = \hat{T}_{i+1} - \hat{T}_i$, and $\hat{X}_i = \hat{X}(\hat{T}_i)$. The kernel

$$\hat{Q}(x) = \left[\hat{Q}_{i,j}(x)\right] = \left[pr\{\hat{X}_{k+1} = j, \hat{V}_k \leq x \mid \hat{X}_k = i\}\right]$$

is given by

$$\hat{Q}_{i,j}(x) = \begin{cases} Q_{i,j}\left(\frac{x}{r_i}\right) + \sum_{l \in S_0} Q_{i,l}\left(\frac{x}{r_i}\right) M_{l,j} & \text{if } i \in S_T \\ Q_{i,j}(x) & \text{if } i \in S_A \end{cases}$$

If state $i \in S_T$ does not reach any state in S_0, this transformation preserves the type of the $Q_{i,j}(\cdot)$ distribution, it merely stretches or compresses its shape on the horizontal axis, from "time" to "reward". If such a state in S_0 does indeed exist, $\hat{Q}_{i,j}(\cdot)$ is a mixture distribution, possibly completely different from $Q_{i,j}(\cdot)$, which is just a component in mixture ($Q_{i,j}(\cdot)$ could even be identically equal 0).

The matrix

$$\hat{P} = \hat{Q}(\infty) = \left[\hat{P}_{i,j}\right] = \left[pr\{\hat{X}_1 = j | \hat{X}_0 = i\}\right] = \begin{bmatrix} \hat{P}^{[TT]} & \hat{P}^{[TA]} \\ 0 & I \end{bmatrix}$$

is related to P by the following relation:

$$\hat{P} = P^{[11]} + P^{[10]}M = P^{[11]} + P^{[10]}P^{[01]} + P^{[10]}P^{[00]}P^{[01]} + P^{[10]}P^{[00]^2}P^{[01]} + \ldots \qquad (1)$$

$\hat{P}$ defines the DTMC obtained from $\{X_k, k \in \mathbb{N}\}$ when only transitions into states of $\hat{S}$ are considered:

$$\forall i,j \in \hat{S},\ \hat{P}_{i,j} = \sum_{k=1}^{\infty} pr\{X_k = j, \forall n\ 0 < n < k\ X_n \in S_0 \mid X_0 = i\}$$

That is, a single step from $i \in \hat{S}$ to $j \in \hat{S}$ in $\{\hat{X}_k, k \geq 0\}$ can be seen as a sequence of steps in $\{X_k, k \geq 0\}$, starting from i, ending in j, and such that the visited states, apart from the first and the last, belong to S_0.

The initial probability vector

$$\underline{\hat{\pi}} = [\hat{\pi}_i] = \left[pr\{\hat{X}(0) = i\}\right]$$

of the transformed process is given by

$$\underline{\hat{\pi}} = \underline{\pi}^{[1]} + \underline{\pi}^{[0]}M = \left[\underline{\hat{\pi}}^{[T]}, \underline{\hat{\pi}}^{[A]}\right] = \left[\underline{\pi}^{[T]} + \underline{\pi}^{[0]}M^{[0T]}, \underline{\pi}^{[A]} + \underline{\pi}^{[0]}M^{[0A]}\right] \tag{2}$$

That is, the probability that the initial state for $\{\hat{X}_k, k \geq 0\}$ is $i \in S_1$ is computed as the sum of the probability that the initial state for $\{X_k, k \geq 0\}$ is i plus the probability that the initial state for $\{X_k, k \geq 0\}$ belongs to S_0 but the first state visited in S_1, after an arbitrary number of transitions in S_0, is i.

Define $N(n)$ and $\hat{N}(n)$ as the number of states in S_T visited in the first n steps by $\{X_k, k \geq 0\}$ and $\{\hat{X}_k, k \geq 0\}$ respectively, including the initial state. Using the definition $E_{(\alpha)} = 1$ if event α is true, $E_{(\alpha)} = 0$ otherwise:

$$N(n) = \sum_{k=0}^{n} E_{(X_k \in S_T)} \qquad \hat{N}(k) = \sum_{k=0}^{n} E_{(\hat{X}_k \in S_T)}.$$

$N(\infty)$ and $\hat{N}(\infty)$ are the number of states in S_T visited before absorption by the two processes respectively, including the initial state. We obtain the following results:

$$pr\{\hat{N}(\infty) = 0\} = pr\{\hat{X}_0 \in S_A\} = 1 - \underline{\hat{\pi}}^{[T]}\underline{1} = \underline{\hat{\pi}}^{[A]}\underline{1}$$

$$\forall n \geq 1, \quad pr\{\hat{N}(\infty) = n\} = \underline{\hat{\pi}}^{[T]}\hat{P}^{[TT]^{n-1}}P^{[TA]}\underline{1}$$

where $\underline{1}$ is the column vector with all entries equal to 1. The following results can be found in [12]:

$$\forall i \in S_1,\ \forall n \geq 0,\ pr\{N(\infty) = n \mid X_0 = i\} = pr\{\hat{N}(\infty) = n \mid \hat{X}_0 = i\}$$

$$\forall i \in S_0,\ \forall n \geq 0,\ \ pr\{N(\infty) = n \mid X_0 = i\} = \sum_{j \in S_1} M_{i,j} pr\{N(\infty) = n \mid X_0 = j\}$$

$$\forall n \geq 0,\ \ pr\{N(\infty) = n\} = pr\{\hat{N}(\infty) = n\}$$

Using the above definitions and results, the following lemmas are proved (in the appendix):

Lemma 1: (dealing with the case $X_0 \in S_0$) $\forall i \in S_0,\ \forall n \geq 0,$

$$pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = i\} = \sum_{j \in S_1} M_{i,j} pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = j\}$$

$$\forall i \in S_0, \quad C_i(x) = \sum_{j \in S_1} M_{i,j} C_j(x)$$

Lemma 2: (relating the two processes, given the number of steps in S_T before absorption) $\forall i \in S_T,\ \forall x \geq 0,\ \forall n \geq 0,$

$$pr\{\hat{X}(x) \in S_A\ ,\ \hat{N}(\infty) = n \mid \hat{X}_0 = i\} = pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = i\}$$

Lemma 3: (eliminating the condition on the number of steps to absorption from the previous lemma)

$$C_i(x) = \begin{cases} \hat{F}_i(x) & \text{if } i \in S_1 \\ \sum_{j \in S_1} M_{i,j} \hat{F}_l(x) & \text{if } i \in S_0 \end{cases}$$

where

$$\forall j \in S_1,\ \ \hat{F}_j(x) = pr\{\hat{X}(x) \in S_A \mid \hat{X}_0 = j\}$$

Removing the dependence from the initial state in Lemma 3, we obtain:

Theorem 1:

$$\begin{aligned}
C(x) &\stackrel{\text{def}}{=} \sum_{i \in S} C_i(x)\pi_i \\
&= \sum_{i \in S_1} C_i(x)\pi_i + \sum_{i \in S_0} C_i(x)\pi_i \\
&= \sum_{i \in S_1} \hat{F}_i(x)\pi_i + \sum_{i \in S_0} \left(\sum_{j \in S_1} M_{i,j} \hat{F}_j(x) \right) \pi_i \quad \text{using Lemma 3} \\
&= \sum_{i \in S_1} \hat{F}_i(x)\pi_i + \sum_{j \in S_1} \hat{F}_j(x) \left(\sum_{i \in S_0} \pi_i M_{i,j} \right) \\
&= \sum_{i \in S_1} \hat{F}_i(x)\pi_i + \sum_{j \in S_1} \hat{F}_j(x) \left(\hat{\pi}_j - \pi_j \right) \quad \text{using Equation (2)} \\
&= \sum_{j \in S_1} \hat{F}_j(x)\hat{\pi}_j
\end{aligned}$$

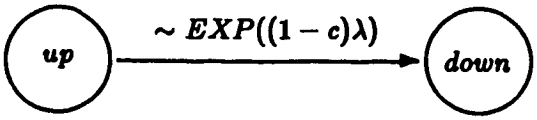


Figure 3: The equivalent Markov chain

where the last quantity can be defined as $\hat{F}(x)$, the unconditional distribution of the time to absorption for process $\{\hat{X}(x), x \geq 0\}$.

The transformation of this section, applied to the example of figure 1 when γ, the reward of state *recover*, is zero, gives $\hat{S} = \{up, down\}$,

$$\hat{Q}(x) = \begin{bmatrix} c(1-e^{-\lambda x}) & (1-c)(1-e^{-\lambda x}) \\ 0 & \delta(x-1) \end{bmatrix}$$

$$\hat{P} = \begin{bmatrix} c & (1-c) \\ 0 & 1 \end{bmatrix}$$

Thus the system stays in state *up* for a number of periods, until it jumps to state *down*, where it gets absorbed. Each period is exponentially distributed with parameter λ (the semi-Markov process is independent, since the distribution of the period does not depend on whether the destination state is *up* or *down*) and the number of periods up to absorption has a geometric distribution with parameter $(1-c)$. The sum of the lenghts of all the periods is exponentially distributed with parameter $\lambda(1-c)$, so the process is equivalent, as far as absorption time is concerned, to a Markov chain with state space $\{up, down\}$, where the transition from state *up* to state *down* has rate $\lambda(1-c)$, and where state *down* is absorbing (see figure 3). Then

$$pr\{Y(\infty) \leq x\} = pr\{\hat{X}(x) = down\} = 1 - e^{-\lambda(1-c)x}$$

5 Elimination of the vanishing states and computation of $C(x)$

The first part of this section discusses the cost of performing the elimination of the vanishing states when the state space S is finite. Various methods to compute the functions $\hat{F}_i(x)$, needed to obtain $C(x)$ using Theorem 1, are discussed at the end of the section.

In the GSPN, the presence of a recurrent subset of vanishing states implies that the process becomes stochastically discontinuous. By contrast, during the transformation of the process $\{X(t), t \geq 0\}$ into $\{\hat{X}(x), x \geq 0\}$, this situation cannot arise, since recurrent subsets are neither present initially nor generated during the construction (slowing down or accelerating the sojourn time in a state does not change the relative transition probabilities from a state to the other states). The absence of recurrent subset of vanishing states guarantees the existence of the inverse $(I - P^{[00]})^{-1}$.

Define G as the subgraph obtained from the state transition diagram when only the vanishing states are considered. G can contain *self-loops* and *transient loops.*

A self-loop is a transition from a vanishing state to itself ($P_{i,i} \neq 0$ for some $i \in S_0$), and it can be simply ignored: $P_{i,i}$ is set to 0 and the other non-zero entries in row i for $P^{[00]}$ and $P^{[01]}$ are renormalized (the new value of $P_{i,j}$ is $P_{i,j}/(1 - P_{i,i})$, for $j \in S, j \neq i$). Since state i is vanishing, this modification does not change the final result (only the final destination is relevant, not the number of jumps into state i before finally leaving it). The diagonal entries of $P^{[00]}$ from now on can be assumed zero.

A transient loop is a set of mutually reachable transient states, a strongly connected component in G. The presence of transient loops is not a problem in principle, but it can negatively affect the performance of the algorithm for the elimination of the vanishing states. Let n_0 and n_1 be the number of states in S_0 and S_1 respectively, let $x^{[ij]}$ be the number of non-zero entries in $P^{[ij]}, i, j \in \{0, 1\}$, and assume that a sparse storage technique is used (for simplicity of notation assume $x^{[ij]}$ larger than both n_i and n_j).

If no transient loop is present, a partial ordering can be defined on the states of S_0 (nodes of G); from it, a compatible total ordering can be found in time $O(x^{[00]})$, by means of a topological sort [14], so that, if state i follows state j, then j cannot be reached from i in G.

If the total ordering is applied to S_0, $P^{[00]}$ becomes strictly upper triangular:

$$\left(P^{[00]}\right)^{n_0} = 0 \quad \text{and} \quad \left(I - P^{[00]}\right)^{-1} = \sum_{k=0}^{n_0-1} \left(P^{[00]}\right)^k$$

hence $\left(I - P^{[00]}\right)^{-1}$ is upper triangular with diagonal entries equal to 1. Although $\left(I - P^{[00]}\right)^{-1}$ is never explicitely computed in practice, the triangular nature of $P^{[00]}$ allows great efficiency; the required steps are listed in the following, together with the asymptotic execution time requirements:

- solve for M in $(I - P^{[00]})M = P^{[01]}$ $n_1 O(x^{[00]})$
- compute W using $W = P^{[10]}M$ $n_1 O(x^{[10]})$
- compute $\hat{P}$ using $\hat{P} = P^{[11]} + W$ $O(n_0 n_1 + x^{[11]})$

The total time requirement to compute $\hat{P}$ is then (given the assumption $x^{[10]} \geq n_1$ and $x^{[11]} \geq n_1$)

$$O((x^{[00]} + x^{[10]})n_1) \tag{3}$$

If transient loops are present, the worst-case bound on the time requirement is higher, an LU-factorization is involved, and the obtained matrices L and U can have a large fill-in, possibly $O(n_0^2)$:

- factorize $P^{[00]}$ into L and U $O(n_0^3)$
- solve for M in $(I - P^{[00]})M = P^{[01]}$ using L and U $n_1 O(n_0^2)$
- compute W using $W = P^{[10]}M$ $n_1 O(x^{[10]})$
- compute $\hat{P}$ using $\hat{P} = P^{[11]} + W$ $O(n_0 n_1 + x^{[11]})$

So the total time requirement is

$$O(n_0^3 + (x^{[10]} + n_0^2)n_1) \tag{4}$$

Comparing (3) with (4), the term $O(x^{[00]}n_1)$ is substituted by the term $O(n_0^3 + n_0^2 n_1)$. The two terms are equivalent only if $x^{[00]} = O(n_0^2)$ and if $n_0 = O(n_1)$.

Finally, if transient loops are present, but their size can be considered "small" with respect to n_0, the topological sort idea can still be used to improve the performance. A partial ordering is now implicitly defined among the transient loops (strongly connected components in G), not among the states (nodes of G), if each state not in a transient loop is considered as a transient loop in itself. The strongly connected components can be found using a modification of the depth-first search algorithm [15], so the number of operations to find a total ordering on the states of S_0 is still $O(x^{[00]})$ as before. The total ordering defined this way is such that if state i follows state j, then either state j cannot be reached from state i in G, or states i and j are mutually reachable in G (they are in the same strongly connected component).

Assume for simplicity that there are exactly n_0/k transient loops of size k each: matrix $P^{[00]}$ can then be put in upper block triangular form, where each block $P^{[00]}_{i,j}, i \leq j$ is of size $k \times k$. Also, denote with $M_{i,j}$ and $P^{[01]}_{i,j}$ the $k \times 1$ blocks forming M and $P^{[01]}$ respectively (a small abuse of notation, since we already used the same symbols to indicate *elements* of a matrix, not *blocks*). Each diagonal block must then be LU-factorized, introducing fill-in only in the block itself, then a block-oriented back-substitution procedure can be used to compute M, k rows at a time, without any fill-in [16, page 161]. The required steps are:

- for $i = 1$ upto n_0/k do $\qquad n_0/k \cdots$
  factorize $P^{[00]}_{i,i}$ into L_i and U_i $\qquad \cdots O(k_0^3)$
- for $i = n_0/k$ downto 1 do $\qquad n_0/k \cdots$
  for $j = 1$ upto n_1 do $\qquad \cdots n_1 \cdots$
  compute A using $A = \sum_{l=i+1}^{n_0/k} P^{[00]}_{i,l} M_{l,j}$ $\qquad O(\alpha)$
  solve for $M_{i,j}$ in $(I - P^{[00]}_{i,i}) M_{i,j} = P^{[01]}_{i,j} - A$ using L_i and U_i $\qquad O(k^2)$
- compute W using $W = P^{[10]} M$ $\qquad n_1 O(x^{[10]})$
- compute $\hat{P}$ using $\hat{P} = P^{[11]} + W$ $\qquad O(n_0 n_1 + x^{[11]})$

The expression $O(\alpha)$ depends on the number of non-zero entries in each $P^{[00]}_{i,l}$, but, summing over all the values of l and i it gives $O(x^{[00]})$, to be summed over all n_1 values of j, so the total time requirement is

$$O(k^2 n_0 + k n_0 n_1 + (x^{[00]} + x^{[10]}) n_1) \tag{5}$$

If $k = O(n_0)$, there is a small (constant) number of large loops: expressions (5) and (4) coincide. If $k = O(1)$, there is a large number of small (constant size) loops: expressions (5) and (3) coincide. Finally, if k is a sublinear function of n_0, for example $k = O(\sqrt{n_0})$, the dominant term in (5) will be determined by the relative value of n_0, n_1, $x^{[00]}$, and $x^{[10]}$, but the running time of the algorithm will be in any case improved by exploiting this ordering, especially considering that the determination of the ordering itself is an inexpensive operation in comparison with the other steps.

So far the discussion has been on the computation of $\hat{P}$. It should be clear how the same bounds hold when considering the computation of $\hat{Q}$, although, in general, the analogous expression

$\hat{Q}(x) = Q^{[11]}(x/r) + Q^{[10]}(x/r)M$ (where $Q^{[11]}$ and $Q^{[10]}$ have the obvious meaning) involves sums of *real functions*, not of *real numbers*. The bounds are nonetheless valid even in this case, for example the reader could imagine that they represent the time to output $\hat{Q}$ in symbolic sparse format.

The elimination of vanishing states helps the subsequent numerical solution both by reducing the overall size of the problem, especially when n_0 is large in comparison to n_1, and by avoiding the introduction of distributions having extremely small averages, to approximate vanishing states. As it will be pointed out in the section on the examples, the existence of "fast" states puts an excessive burden on the numerical solution. Furthermore, the incorporation of the reward information into the rate allows us to reduce the problem of computing the accumulated reward to that of computing the time to absorption. Once matrix $\hat{Q}$ has been obtained, the problem of computing $C(x)$ (by computing $\hat{F}_i(x)$ first) for the relevant value(s) of x or, even better, as a function of x, remains. Unfortunately this problem is generally harder than the elimination of the vanishing states itself. Viable methods include numerical integration of a system of coupled Volterra equations [17], the use of Laplace transforms, and, if all $Q_{i,j}(\cdot)$ are phase-type distributions, transformation of the semi-Markov process into a CTMC and solution using approximate numerical methods. Regardless of the method employed, additional advantages can be obtained by lumping all the absorbing states together, an operation always advisable unless detailed information on the particular failure state (reason or "mode" of the failure) is required. In the following it is assumed that this lumping is performed, and the absorbing macro-state is L.

For simplicity we will drop the "$\hat{}$" notation, since the problem is now the computation of the time to absorption for a semi-Markov process, and the notion that this process was obtained from yet another process is irrelevant. The goal is the computation of the time-dependent state probability for state L (or the first-passage time distribution to state L), conditioned on the initial state i, for each state $i \in S \setminus S_A$ such that $\pi_i > 0$.

Conceptually only $O(n_1)$ distribution functions are needed, or even less, for example only one function is needed if the initial state is certain, that is, if $\exists i \in S \setminus S_A$ such that $\pi_i = 1$. Unfortunately, in general, additional information must be computed to obtain the required distribution(s), so that the solution can be very expensive in terms of execution time, memory requirements, or both.

If each distribution is approximated using a phase-type expansion, a Markov chain is obtained.

A survey of numerical methods to solve for the time-dependent state probabilities is presented in [18]. There is a tradeoff between the number of stages used in the expansions and how well the distributions are fitted, but the first and main tradeoff is between size and simplicity: the state space is increased to allow a (simpler) Markovian analysis.

A direct approach for the transient analysis of the semi-Markov process involves solving the system of integral equations

$$\forall i \in S, \ \forall j \in S, \ \ \Pi_{i,j}(x) = E_{(i=j)}(1 - H_i(x)) + \sum_{l \in S} \int_0^x \Pi_{l,j}(x-u) dQ_{i,l}(u)$$

where

$$\Pi = [\Pi_{i,j}(x)] = [pr\{X(x) = j | X_0 = i\}]$$

is the the matrix expressing the time-dependent probability of being in state j conditioned on i being the initial state and

$$H(x) = [H_i(x)] = [pr\{V_k \leq x | X_k = i\}] = \left[\sum_{l \in S} Q_{i,l}(x)\right]$$

is the vector of the holding time distributions in each state, independently of the destination state. Further discussion on numerical methods for the above integration may be found in [17].

Laplace transforms can also be employed, as done in [5, page 320]. Define the k-fold convolution of Q with itself as

$$Q^{(k)}(x) = [Q_{i,j}^{(k)}(x)] = pr\{X_k = j, T_k \leq x | X_0 = i\}$$

the expected number of visits to state j in $[0, x]$ conditioned on i being the initial state as

$$R(x) = [R_{i,j}(x)] = \left[\sum_{k=0}^{\infty} Q_{i,j}^{(k)}(x)\right]$$

and the distribution of first-passage time in each state conditioned on the initial state as

$$G(x) = [G_{i,j}(x)] = [pr\{\min_{u \geq 0}\{X(u) = j\} \leq x | X_0 = i\}] \ (\mathrm{i} \neq \mathrm{j})$$

and the distribution of the time between two successive occurrences of j as

$$G_{j,j}(x) = pr\{\min_{u \geq T_1}\{X(u) = j\} \leq x | X_0 = j\}$$

Let $Q^*(s)$, $R^*(s)$, and $G^*(s)$ be respectively the matrices representing the Laplace transforms of the above. Then Çinlar shows that

$$R^*(s) = (I - Q^*(s))^{-1}$$

$$G^*_{j,j}(s) = 1 - \frac{1}{R^*_{j,j}(s)} \qquad G^*_{i,j}(s) = \frac{R^*_{i,j}(s)}{R^*_{j,j}(s)} \quad (i \neq j)$$

If $Q^*(s)$ is partitioned as

$$Q^*(s) = \begin{bmatrix} Q^{[TT]*}(s) & Q^{[TL]*}(s) \\ 0 & Q^*_{L,L}(s) \end{bmatrix}$$

then $R^*(s)$ and $G^*(s)$ can accordingly be written as

$$R^*(s) = \begin{bmatrix} (I - Q^{[TT]*}(s))^{-1} & R^{[TL]*}(s) \\ 0 & (1 - Q^*_{L,L}(s))^{-1} \end{bmatrix}$$

$$G^*(s) = \begin{bmatrix} G^{[TT]*}(s) & F^*(s) \\ 0 & Q^*_{L,L}(s) \end{bmatrix}$$

Only the vector $F^*(s)$ is needed, since its elements are the Laplace transform of $F_i(x)$, for $i \in S \backslash S_A$. The following steps, not present in Çinlar, show how $F^*(s)$ can be computed directly, without obtaining $R^*(s)$. They also show how the answer does not depend on the holding time distribution in the absorbing state L, as it should be, since the process is studied *up to* absorption.

Since $R^*(s)(I - Q^*(s)) = I = (I - Q^*(s))R^*(s)$, $R^{[TL]*}(s)$ can be computed as the solution of

$$(I - Q^{[TT]*}(s))R^{[TL]*}(s) + Q^{[TL]*}(s)(1 - Q^*_{L,L}(s))^{-1} = 0$$

or

$$(I - Q^{[TT]*}(s))R^{[TL]*}(s)(1 - Q^*_{L,L}(s)) = -Q^{[TL]*}(s)$$

but, observing that $R^{[TL]*}(s)(1 - Q^*_{L,L}(s)) = F^*(s)$, the above can be rewritten as

$$(I - Q^{[TT]*}(s))F^*(s) = -Q^{[TL]*}(s) \tag{6}$$

eliminating the dependency on $Q^*_{L,L}(s)$. In oreder, to obtain $F(t)$, we use a two-phase approach. We first solve the matrix equation (6) in s-domain and subsequently use numerical inversion of the vector $F^*(s)$ into $F(t)$; see [19] for a discussion on inversion of Laplace transforms.

6 Examples

Limited number of repairs [3]

Consider a system experiencing failures. When a failure occurs, the system is repaired, but only up to $k-1$ times. At the k^{th} failure, the system will not be repaired any more, it will remain in the

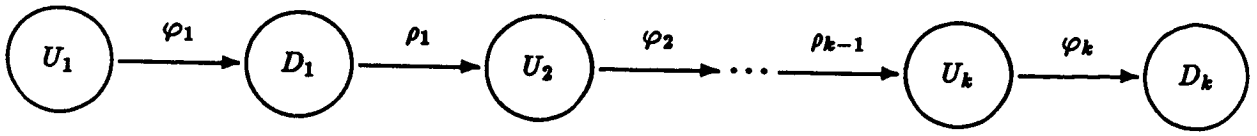


Figure 4: Limited repairs

failed state (figure 4). The state space is $S = \{U_1, D_1, U_2, \ldots, U_k, D_k\}$, there is only one absorbing state, $S_A = \{D_k\}$, and $\underline{\pi}(0) = [1, 0, 0, \ldots, 0]$. Transitions are from state U_i to state D_i $(1 \leq i \leq k)$ and from state D_i to state U_{i+1} $(1 \leq i \leq k-1)$.

$$Q_{U_i,D_i}(t) = \varphi_i(t) \qquad \text{(distribution of } i^{th} \text{ up period)}$$
$$Q_{D_i,U_{i+1}}(t) = \rho_i(t) \qquad \text{(distribution of } i^{th} \text{ repair period)}$$

If the reward rate in state U_i is $r_i > 0$, and in state D_i is 0 $(1 \leq i \leq k)$, then $D_1, \ldots, D_{k-1}$ are vanishing states. Applying the transformation, $\hat{S} = \{U_1, U_2, \ldots, U_k, D_k\}$ and

$$\hat{Q}_{U_i,U_{i+1}}(x) = \hat{\varphi}_i(x) = \varphi_i\left(\frac{x}{r_i}\right) \qquad (1 \leq i \leq k-1)$$
$$\hat{Q}_{U_k,D_k}(x) = \hat{\varphi}_k(x) = \varphi_k\left(\frac{x}{r_k}\right)$$

The distribution of the accumulated reward up to absorption is

$$\begin{aligned} C_{U_1}(x) &= (\hat{\varphi}_1 \star \cdots \star \hat{\varphi}_k)(x) \\ &= \int_0^x \int_0^{x-x_1} \cdots \int_0^{x-x_1-\cdots-x_{k-2}} \hat{\varphi}_k(x - x_1 - \cdots - x_{k-1})\, d\hat{\varphi}_{k-1}(x_{k-1}) \cdots d\hat{\varphi}_2(x_2)\, d\hat{\varphi}_1(x_1) \\ &= \int_0^{\frac{x}{r_1}} \int_0^{\frac{x-y_1 r_1}{r_2}} \cdots \int_0^{\frac{x-y_1 r_1-\cdots-y_{k-2} r_{k-2}}{r_{k-1}}} \varphi_k\left(\frac{x - y_1 r_1 - \cdots - y_{k-1} r_{k-1}}{r_k}\right) \\ &\quad d\varphi_{k-1}(y_{k-1}) \cdots d\varphi_2(y_2)\, d\varphi_1(y_1) \end{aligned}$$

where $\star$ denotes the convolution operator. The expression for the LST transform is simpler:

$$\tilde{C}_{U_1}(s) = \int_0^\infty e^{-sx}\, dC_{U_1}(x) = \prod_{i=1}^{k} \tilde{\varphi}_i\left(\frac{s}{r_i}\right) = \prod_{i=1}^{k} \int_0^\infty e^{-\frac{s}{r_i}t}\, d\varphi_i(t)$$

Note that if $r_i = 1$ for all the non-absorbing states, then $C_{U_1}(x)$ is simply the distribution of total uptime until failure. Interesting cases are:

- $Q_{U_i,D_i}(\cdot) \sim EXPO(\lambda_i)$, for $1 \leq i \leq k$, then $C_{U_1}(\cdot) \sim HYPO(\lambda_1/r_1, \ldots, \lambda_k/r_k)$, assuming that all the fractions λ_i/r_i are different. Note the independence of the result from the distributions $Q_{D_i,U_{i+1}}(\cdot)$. Independence from the sojourn time distribution of the states with zero reward is a general property of this method, further simplifying the analysis.

- $Q_{U_i,D_i}(\cdot) \sim EXPO(\lambda)$ and $r_i = 1$, for $1 \leq i \leq k$, then $C_{U_1}(\cdot) \sim ERL(\lambda, k)$. This could correspond to the simple case of a uniprocessor system, which can be either up or down.

- $Q_{U_i,D_i}(\cdot) \sim EXPO((k+1-i)\lambda)$ and $r_i = k+1-i$, for $1 \leq i \leq k$, then $C_{U_1}(\cdot) \sim ERL(\lambda, k)$, as before, interestingly enough, since this could correspond, for example, to a k processor system without repair (ρ_i is related to the time to reconfigure the system after the i-th failure, not to the time to repair it)

Cumulative computation of an $M/M/2/K$ system before failure

Consider a queueing system with Poisson arrivals, two servers, exponentially distributed service times, and finite capacity K. Let $1/\lambda$ and $1/\mu$ be the average interarrival and service times respectively. Life-time and repair-time of each server are independent and exponentially distributed with average $1/\phi$ and $1/\rho$ respectively. For example the reader might imagine a system performing a certain set of tasks using two processors. Tasks are sorted according to their priority, and low priority ones may be discarded if there is no waiting room available. When only one processor is available, tasks are more likely to be discarded (especially if with lower priority), but the system is still able to survive. If the second processor also fails, while the first is being repaired, no task can be performed, and the system is considered down. The measure of interest is the cumulative computation performed up to the first failure of the system.

The state of the system is denoted by a pair (i, j) where $i = 0, 1, 2$ is the number of non-failed servers, and j is the number of jobs in the system (j is absent if $i = 0$). The system is initially in state $(2, 0)$ with probability 1. The cumulative computation for this system is represented by $Y(\infty)$ for the CTMC in figure 5 when the rewards are defined as

$$r_{(2,0)} = 0, \quad r_{(2,1)} = \mu, \quad \forall j \geq 2, \quad r_{(2,j)} = 2\mu,$$

$$r_{(1,0)} = 0, \qquad \forall j \geq 1, \quad r_{(1,j)} = \mu, \quad \forall j \geq 0, \quad r_{(0)} = 0.$$

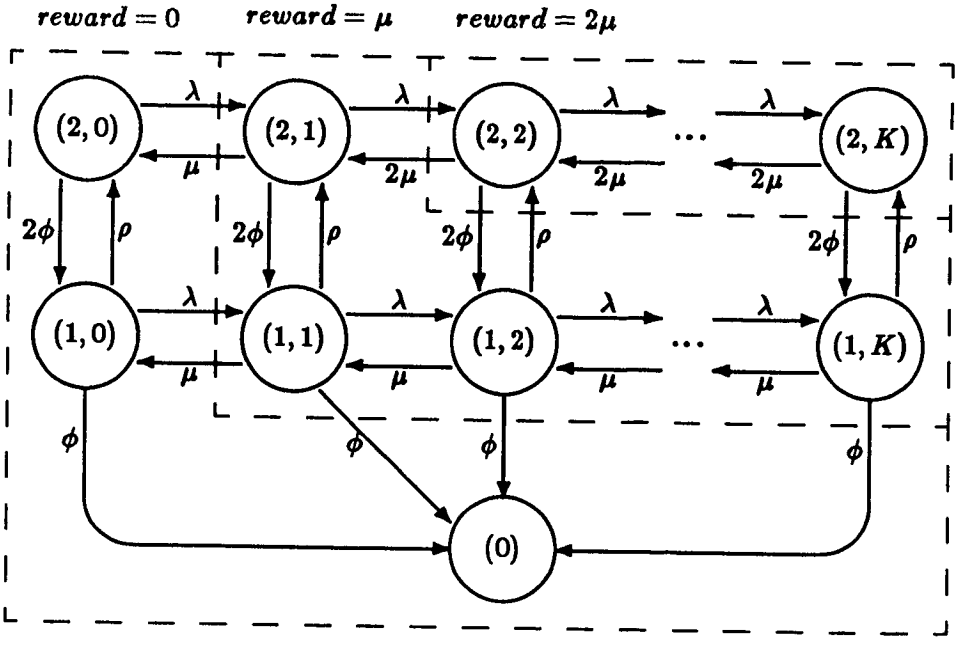
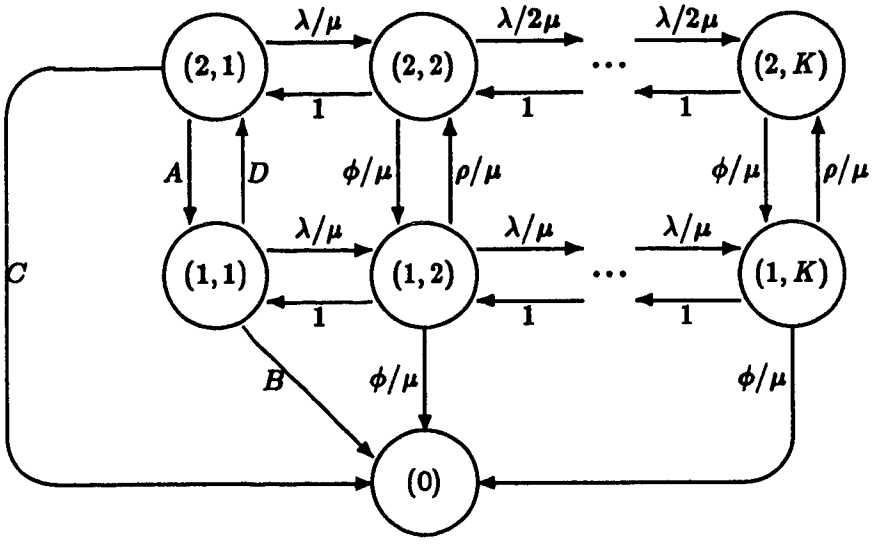


Figure 5: The Markov chain corresponding to the $M/M/2/K$ system.

The SHARPE package [20] was used to analyze $C(x)$ for the system, both with an "approximate" approach (the reward for states $(2,0)$ and $(1,0)$ is defined as 0.000001) and with the "exact" method proposed in this paper (states $(2,0)$ and $(1,0)$ are eliminated because vanishing). The transformed CTMC is in figure 6. The initial probability vector, given the choice $\pi_{(2,0)} = 1$, is

$$\hat{\pi}_{(2,1)} = \frac{\lambda(\phi+\lambda+\rho)}{\rho\lambda+(2\phi+\lambda)(\phi+\lambda)} \quad \hat{\pi}_{(1,1)} = \frac{2\lambda\phi}{\rho\lambda+(2\phi+\lambda)(\phi+\lambda)} \quad \hat{\pi}_{(0)} = \frac{2\phi^2}{\rho\lambda+(2\phi+\lambda)(\phi+\lambda)}$$

The values assigned to the parameters are $K = 10$, $\mu = 1.0$, $\lambda = 0.6$, $\phi = 0.00001$, and $\rho = 0.005$. The distribution of $C(x)$ is plotted in Figure 7. The difference between the exact and the approximate values of $C(x)$ is plotted in figure 8 (note the logarithmic scale on the x axis). As expected, the approximate solution is somewhat more optimistic than the exact one (shown in figure 7), since it assumes that some computation is performed in the two states that would otherwise be vanishing. Unfortunately, the presence of extremely small rewards (six orders of magnitude smaller than the rewards for the other states) causes numerical problems in SHARPE. The values of $C(x)$ obtained with the approximate method appear to be correct only up to $x = 10^3$, then they become unstable, getting larger than the one obtained using the exact method.



$$A = 2\phi\left[\frac{1}{\mu} + \frac{\lambda}{\rho\lambda + (2\phi+\lambda)(\phi+\lambda)}\right] \quad B = \phi\left[\frac{1}{\mu} + \frac{2\phi+\lambda}{\rho\lambda + (2\phi+\lambda)(\phi+\lambda)}\right]$$

$$C = \frac{2\phi^2}{\rho\lambda + (2\phi+\lambda)(\phi+\lambda)} \quad D = \rho\left[\frac{1}{\mu} + \frac{\lambda}{\rho\lambda + (2\phi+\lambda)(\phi+\lambda)}\right]$$

Figure 6: The transformed Markov chain corresponding to the $M/M/2/K$ system.

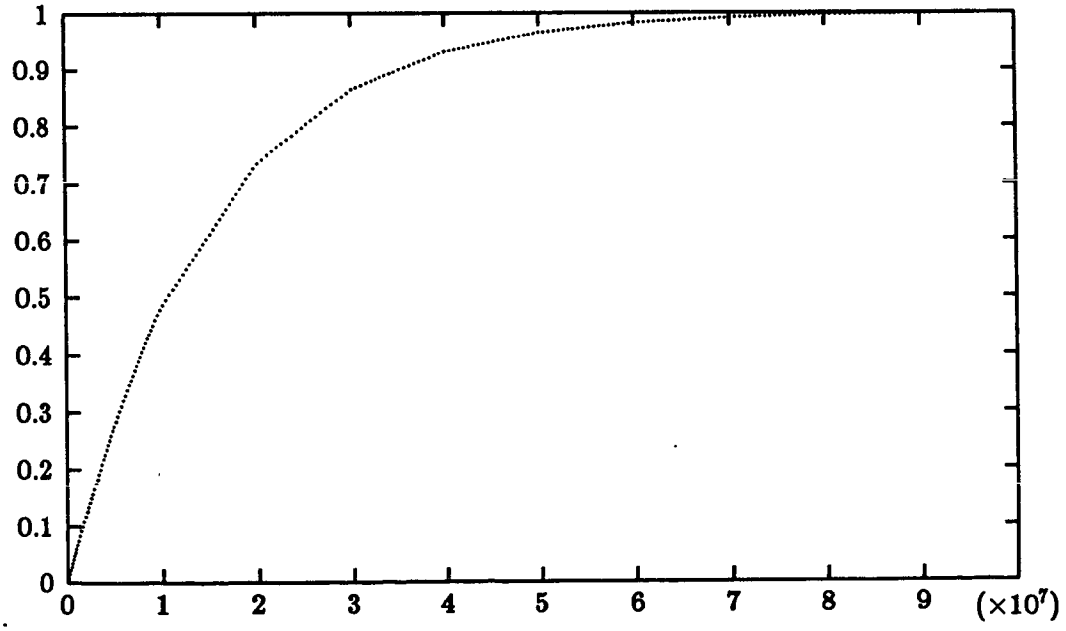


Figure 7: $C(x)$ of the $M/M/2/K$ system.

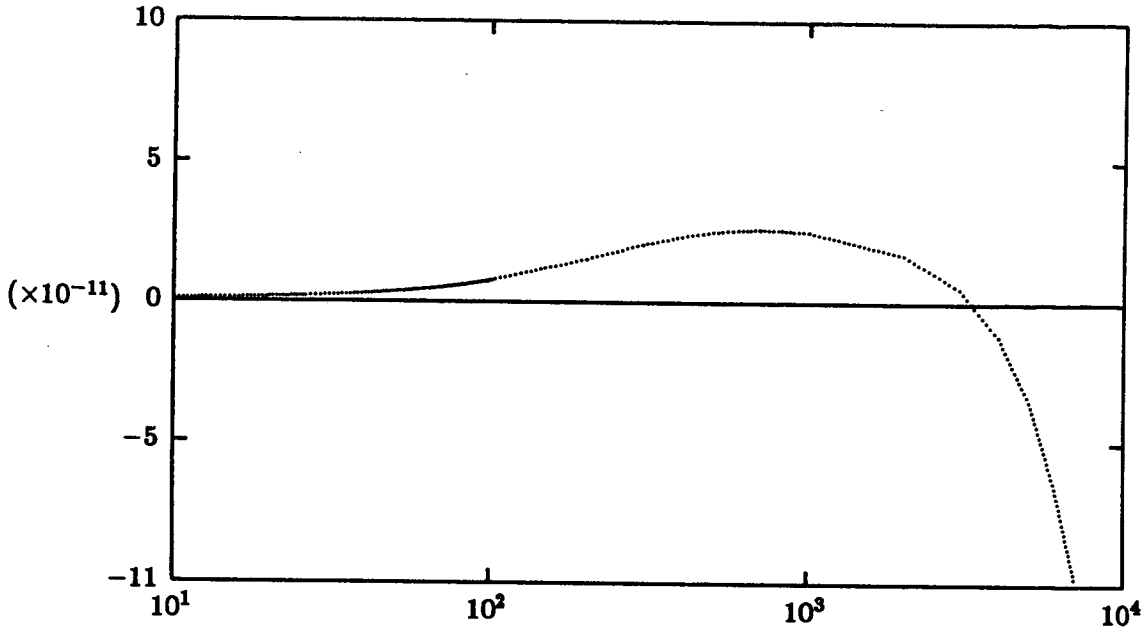


Figure 8: Exact $C(x)$ minus approximate $C(x)$, for $10 \leq x \leq 10000$

Measurement-based performability of a multiprocessor system

In [21], a performability study of an IBM 3081 system is reported. The analysis involves three aspects: performance, failure, and recovery behavior. The working states are classified into clusters, according to the utilization of resources (CPU, channel, direct access storage). States 1 trough 7 correspond to the clusters. When a failure arises in a working state, it is logged as a channel (c), software (e), or direct access storage (d) error. Furthermore, a failure affecting several components is classified as a multiple (m) error. When the system enters state c, e, d, or m, a recovery process tries to restore the state of the system. Hardware (h) and software (s) recovery can be attempted on line. If they fail, an offline (o) recovery is necessary and the system is considered down. It is observed that the failure behavior is dependent on the particular working state. An extra state, 0, is added, to signify the periods of no data collection. A transition from state 0 to state x means that the state is x at the beginning of an observation period. A transition from state y to state 0 means that an observation period ended while the state was y. The state space is then $\{0, 1, 2, 3, 4, 5, 6, 7, c, e, d, m, h, s, o\}$.

The process is assumed to be independent semi-Markovian, and the holding time distributions

$$
\begin{aligned}
H_0 &\sim EXP(10000.0) \\
H_1 &\sim .29EXP(.003) + .71[EXP(.001039) \star EXP(.0047974)] \\
H_2 &\sim .205EXP(.006) + .54588EXP(.000995) + .24912EXP(.0004539) \\
H_3 &\sim .385EXP(.004) + .615[EXP(.002193) \star EXP(.0059258)] \\
H_4 &\sim .073EXP(.002) + .76988EXP(.001030) + .15712EXP(.0002102) \\
H_5 &\sim .195EXP(.01) + .5238EXP(.001030) + .2812EXP(.0005528) \\
H_6 &\sim .33EXP(.004) + .67[EXP(.000998) \star EXP(.014442)] \\
H_7 &\sim .35EXP(.005) + .49119EXP(.002163) + .15881EXP(.0006993) \\
H_c &\sim EXP(.19689) \\
H_e &\sim .925039EXP(.044518) + .074961EXP(.0036075) \\
H_d &\sim .25EXP(.15) + .07EXP(.1) + .59296EXP(.014562) + .08704EXP(.0021377) \\
H_m &\sim EXP(.003817) \star EXP(.0301092) \\
H_h &\sim EXP(2.0) \\
H_s &\sim EXP(1.0)
\end{aligned}
$$

Figure 9: Holding time distributions

for the transient states are fitted to the stage-type distributions shown in figure 9 (the holding time is independent of the next state reached):

$$\forall i \in S, \;\; H_i(t) = pr\{V_k \leq t \mid X_k = i\}$$

The transition probability matrix P resulting from the observations is given in figure 10. Assume $\pi_0 = 1$. The following reward rates for each state are defined:

$$r_0 = \cdots = r_7 = 1; \;\; r_c = .9946; \;\; r_e = .2736; \;\; r_d = .5708; \;\; r_m = .2777; \;\; r_h = r_s = .0001; \;\; r_o = 0$$

Also this example was analyzed using SHARPE. Each stage-type distribution was expanded, increasing the state space, but obtaining a CTMC as a result.

An observation of the transition and reward rates suggests that *this model has two types of vanishing states*. The reward rate for states h and s is almost 0, it would be exactly 0 if SHARPE had a provision for zero reward transient states (the system gains no reward in them). The remaining

	0	1	2	3	4	5	6	7	*c*	*e*	*d*	*m*	*h*	*s*	*o*
0	–	.135	.216	–	.108	.351	.027	.163	–	–	–	–	–	–	–
1	.023	–	.046	.015	.038	.045	.098	–	.015	.227	.387	.106	–	–	–
2	.033	.054	–	–	.067	.034	.087	.013	.007	.262	.383	.060	–	–	–
3	.091	–	.091	–	.091	–	–	–	–	.091	.545	.091	–	–	–
4	.093	.004	.015	–	–	–	.063	–	.007	.232	.504	.082	–	–	–
5	.019	.170	.113	–	.019	–	.132	.057	.019	.151	.226	.094	–	–	–
6	.040	.022	.051	–	.069	.011	–	–	.026	.208	.482	.091	–	–	–
7	.143	.143	.047	.048	–	.047	.048	–	–	.143	.381	–	–	–	–
c	–	–	–	–	–	–	–	–	–	–	–	–	1	–	–
e	–	–	–	–	–	–	–	–	–	–	–	–	.5	.5	–
d	–	–	–	–	–	–	–	–	–	–	–	–	1	–	–
m	–	–	–	–	–	–	–	–	–	–	–	–	.645	.355	–
h	.018	.131	.142	.013	.322	.036	.307	.014	–	–	–	–	–	.017	–
s	.006	.159	.182	.007	.304	.046	.285	.008	–	–	–	–	–	–	.003
o	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1

Figure 10: P, the transition probability matrix of the original process

	1	2	3	4	5	6	7	*c*	*e*	*d*	*m*	*o*
1	.0031	.0510	.0150	.0405	.0531	.0986	.0037	.0150	.2270	.3870	.1060	–
2	.0585	.0071	–	.0706	.0456	.0879	.0184	.0070	.2620	.3830	.0600	–
3	.0123	.1107	–	.1008	.0319	.0025	.0148	–	.0910	.5450	.0910	–
4	.0166	.0351	–	.0100	.0326	.0655	.0152	.0070	.2320	.5040	.0820	–
5	.1726	.1171	–	.0211	.0067	.1325	.0601	.0190	.1510	.2260	.0940	–
6	.0274	.0596	–	.0733	.0250	.0011	.0065	.0260	.2080	.4820	.0910	–
7	.1623	.0779	.0480	.0154	.0972	.0519	.0233	–	.1430	.3810	–	–
c	.1361	.1490	.0131	.3291	.0431	.3123	.0171	–	–	–	–	.00005
e	.1480	.1662	.0101	.3169	.0456	.2987	.0130	–	–	–	–	.0015
d	.1361	.1490	.0131	.3291	.0431	.3123	.0171	–	–	–	–	.00005
m	.1445	.1612	.0109	.3204	.0449	.3027	.0142	–	–	–	–	.0011
o	–	–	–	–	–	–	–	–	–	–	–	1

Figure 11: $\hat{P}$, the transition probability matrix of the new process (entries are rounded)

states have a non-negligible reward, but the holding time in state 0 is negligible in comparison to the one in the other states. State 0 is indeed introduced as a device to represent the interruption of observation epochs, it simply provides a ***probabilistic switch*** from the states having a transition to it $(1,\cdots,7,h,s)$ to the states reached from it $(1,2,4,\cdots,7)$. State 0 is vanishing because the original process does not spend time in it (state 0 is really an abstraction), not because its reward rate is zero, but the final effect is the same: ideally the process does not gain any reward in states 0, h, or s.

Using the notation previously introduced, $\hat{S} = \{1,2,3,4,5,6,7,c,e,d,m,o\}$. Since the original process is independent, also the new process is independent, and, in addition, the holding time distributions for its states are obtained by stretching the corresponding original distributions on the horizontal axis (no mixture distribution is generated):

$$\forall i \in \hat{S} \setminus \{o\}, \quad \hat{H}_i(x) = H_i(\frac{x}{r_i}) \quad \text{and} \quad \hat{H}_o(x) = H_o(x)$$

The transition probability matrix $\hat{P}$ (figure 11) completes the specification of the new process. $\hat{P}$ was obtained applying (1) with the help of the MATLAB package [22].

The analysis of the new process is also performed using SHARPE, expanding the stage-type distributions, as was done for the original process, but now matrix $\hat{P}$ contains non-zero diagonal entries and this leads, in the expanded CTMC, to arcs having coinciding source and destination. When this happens, the correct CTMC is obtained by simply ignoring these arcs (the holding time in the state or stage is increased).

The system was modeled with the original description method using several sets of data, differing in the choice of H_0, r_h, and r_s, to observe how the results were affected (all the other specifications were as in figure 9). The choices were $H_0 \sim EXP(10^n)$ and $r_h = r_s = 10^{-n}$, for n = 1,2,4,6, and 8 (in figure 9, n is 5). Let $C^{[n]}(x)$ be the accumulated reward up to absorption computed this way, for n = 1,2,4,6, and 8.

The system was then modeled with the exact method, where states 0, h, and s are absent, obtaining a distribution of the accumulated reward consistently more pessimistic than the previous ones. The exact accumulated reward, $C^{[\infty]}(x)$, is shown in figure 12).

The absolute difference between $C^{[\infty]}(x)$ and $C^{[1]}(x)$, $C^{[2]}(x)$, and $C^{[4]}(x)$ respectively are reported in figure 13 (for small values of x, from 10 to 100 reward units) and in figure 14 (for large

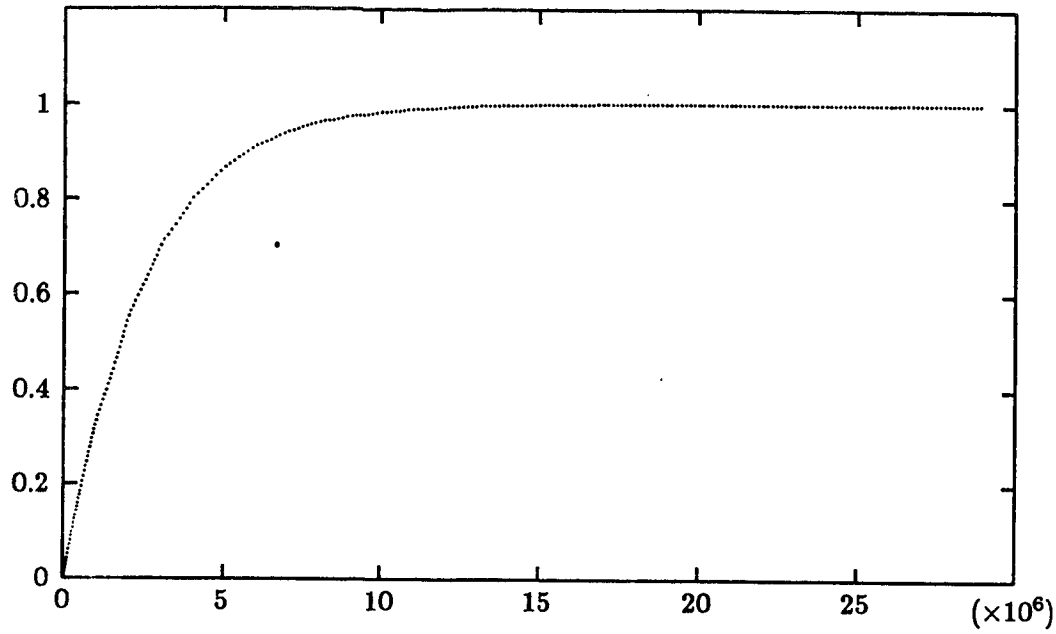


Figure 12: $C^{[\infty]}(x)$ for $0 \leq x \leq 3 \cdot 10^7$

values of x, up to $3 \cdot 10^7$ reward units). As expected, the differences tend to be smaller (closer to 0) as n increases, but raising n too much ($n \geq 6$) causes numerical problems in SHARPE. Interestingly enough, $C^{[4]}(x)$, whose computation is numerically correct, is still consistently apart from $C^{[\infty]}(x)$. The cause of this behavior is the large number of visits to states 0,h, and s before absorption: state 0 has an extremely short holding time, but its reward rate is 1; and states h and s have an extremely small reward rate, but their holding times are not negligible. To obtain an almost perfect match with $C^{[\infty]}(x)$ without numerical problems, both the reward rate for state 0 and the sojourn times for states h and s had to be reduced. In other words, it may be difficult to obtain exact results using a "limiting" operation that will stress, if not compromise, most numerical methods: it is safer, and generally faster, to explicitly eliminate the vanishing states.

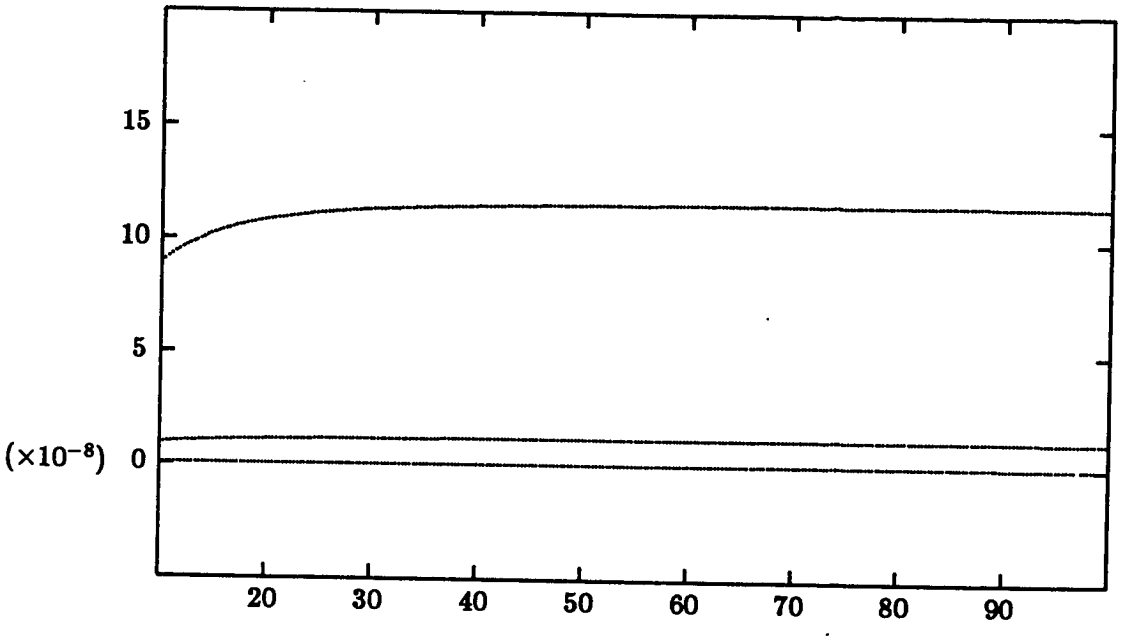


Figure 13: From top to bottom, $C^{[\infty]}(x) - C^{[1]}(x)$, $C^{[\infty]}(x) - C^{[2]}(x)$, and $C^{[\infty]}(x) - C^{[4]}(x)$, for $10 \leq x \leq 100$.

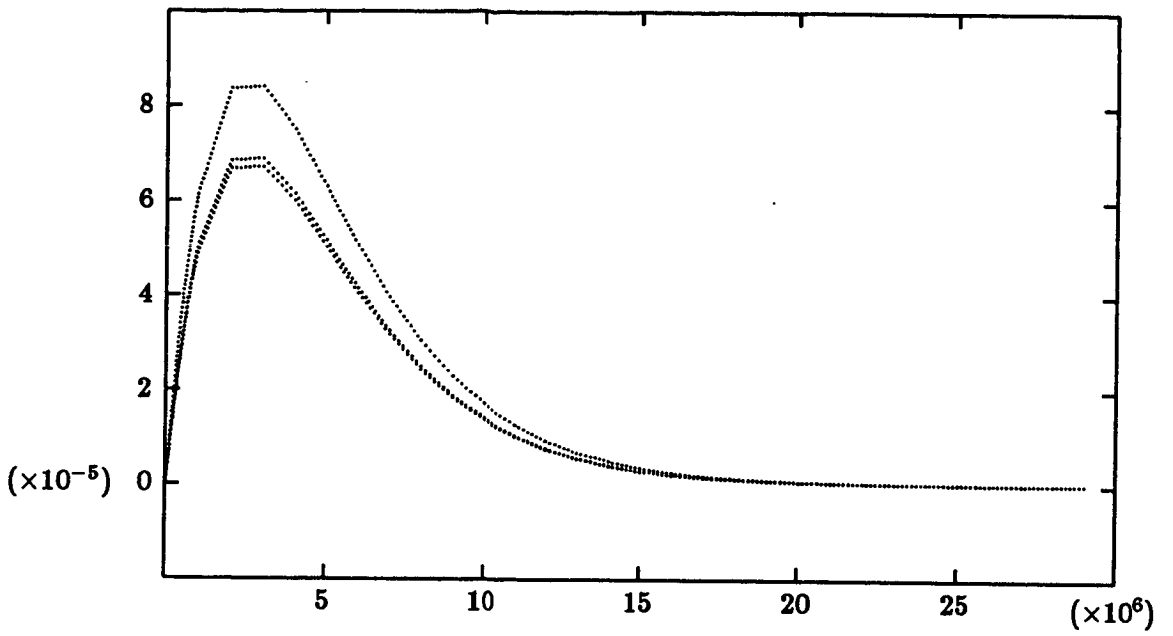


Figure 14: From top to bottom, $C^{[\infty]}(x) - C^{[1]}(x)$, $C^{[\infty]}(x) - C^{[2]}(x)$, and $C^{[\infty]}(x) - C^{[4]}(x)$, for $0 \leq x \leq 3 \cdot 10^7$.

7 Observations

Cumulative time spent in a subset of states up to absorption

If the reward rates for the transient states are restricted to be either 0 or 1, and the reward rates for the absorbing states are 0, and if the set $B \subset S \setminus S_A$ is defined as $B = \{i \in S \setminus S_A \wedge r_i = 1\}$, then $C(x)$ represents the probability that no more than x time units are globally spent in the states of the subset B before absorption. By changing the set of states having reward rate 1, different transformed processes can be obtained, and different aspects of the same system can be analyzed.

Extensions of the class of semi-Markov processes considered

The proposed transformation has been applied to semi-Markov processes with probability of absorption equal 1, and with state space S partitioned into two subsets, $S \setminus S_A$, composed of transient

states with non-negative reward rate, and S_A, composed of absorbing states with zero reward rate. A more general class of semi-Markov processes could be considered, partitioning the state space S into two subsets, S_B, composed of transient states with positive reward rates, and $S \setminus S_B$, composed of states with zero reward rates. States in $S \setminus S_B$ may be recurrent, transient, or absorbing, as long as the states of S_B are transient. It is easy to see, though, that this is a trivial generalization, where a recurrent subset of states with zero reward can be substituted for a single absorbing state with null reward rate.

Another class of semi-Markov process that could fit the proposed approach is exemplified by the process $\{X(t), t \geq 0\}$ having $S = \mathbb{N}$, $\forall i, j \in \mathbb{N}$ $Q_{i,j}(t) \sim EXP(2^i)$, if $j = i + 1$, $Q_{i,j}(t) = 0$ otherwise, $\pi_0 = 1$, a classical example of non-regular CTMC [23]. The properties of the process obtained applying the transformation will be dependent on the reward rates assigned to each state. If $\forall i \in \mathbb{N}$ $r_i = 1$, the transformed process $\{\hat{X}(x), x \geq 0\}$ will be non-regular as well: its lifetime, and $Y(\infty)$, will be finite with probability 1. If $\forall i \in \mathbb{N}$ $r_i = 2^i$, the transformed process has a sojourn time $\sim EXP(1)$ in each state, hence it is regular: its lifetime, and $Y(\infty)$, will be ∞ with probability 1.

Analogously, consider an initial process as before, but with $Q_{i,i+1}(t) \sim EXP(1)$, an example of regular CTMC where all states are transient. If $\forall i \in \mathbb{N}$ $r_i = 1$, the transformed process is regular and its lifetime, and $Y(\infty)$, are ∞ with probability 1. If instead $\forall i \in \mathbb{N}$ $r_i = 2^{-i}$, the transformed process has a sojourn time $\sim EXP(2^i)$ in state i, hence it is irregular, and its lifetime, and $Y(\infty)$, will be finite with probability 1.

Absorbing states with positive reward rate

If a positive reward rate is assigned to an absorbing state reachable from the initial state i, then $E[Y(\infty) \mid X_0 = i] = +\infty$. Furthermore, if the absorbing states S_A are partitioned into S_{A^0} and S_{A^+}, the former containing states with zero reward rate and the latter containing states with positive reward rate, then the value of

$$\alpha_i = pr\{Y(\infty) = +\infty \mid X_0 = i\} = pr\{\hat{X}(t) \in S_{A^+} \mid X_0 = i\} = pr\{X(\infty) \in S_{A^+} \mid X_0 = i\}$$

will influence the subsequent analysis. If $\alpha_i < 1$, the conditional distribution of $Y(\infty)$, given that the process is absorbed in a state with zero reward rate and the initial state i, may be computed,

$$pr\{Y(\infty) \leq x \mid X_0 = i, X(\infty) \in S_{A^0}\}$$

as well as its expectation,

$$E[Y(\infty) \mid X_0 = i, X(\infty) \in S_{A^0}]$$

Once α_i has been computed, the problem is reduced to the one previously discussed, and it can be solved with the same technique (α_i is simply the mass at ∞ for a defective distribution).

Negative rewards

A further complication arises in the presence of negative reward rates, often present, for example, when costs are considered. If some absorbing state has a negative reward rate, then S_A must be partitioned into S_{A^-}, S_{A^0}, and S_{A^+}. α_i^-, α_i^0, and α_i^+ can be defined as the probabilities of being absorbed in each subset respectively. If $\alpha_i^0 > 0$, the conditional distribution of $Y(\infty)$ can be defined as done in the previous paragraph, but its computation is complicated by transient states with negative reward rate. If there is a lower bound $-b$ to the value of these rates, the reward rates of the transient states can be shifted by adding b to them, so that they become non-negative. In particular, if S is finite, b always exists.

Defining $\mathcal{Y}(\infty)$ as the cumulative reward computed using the shifted reward rates, the following expression holds:

$$E[Y(\infty) \mid X_0 = i] = E[\mathcal{Y}(\infty) \mid X_0 = i] - b\, MTTF_i$$

where $MTTF_i$, the expected lifetime starting in state i, is obviously independent of the reward rates, so it can be computed for the original process. Unfortunately the expression cannot be generalized to the distribution of $Y(\infty)$, because the cumulative reward and the lifetime are not stochastically independent.

8 Conclusion

In this paper we have presented an algorithm for the computation of the accumulated reward in a semi-Markov reward process. This is an extension of an algorithm proposed by Beaudry. The

algorithm is robust, since it allows for zero-reward states that are non-absorbing. The algorithm proceeds by replacing zero-reward absorbing states by a probabilistic switch, thus it is related to the elimination of "vanishing" states from the reachability graph of a Generalized Stochastic Petri Net [10] and to the elimination of fast transient states in a decomposition approach to stiff Markov chains [11]. The usefulness of the algorithm is illustrated with several examples.

Appendix

Proof of Lemma 1:

$\forall i \in S_0$, $\forall n \geq 0$,

$$\begin{aligned} & pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = i\} \\ = & \sum_{j \in S_0 \cup S_1} P_{i,j} pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = j\} \\ = & \sum_{j \in S_0} P_{i,j} pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = j\} \\ & + \sum_{j \in S_1} P_{i,j} pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = j\} \end{aligned}$$

Define the column vectors

$$U_0 = (pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = i\}) \quad i \in S_0$$

$$U_1 = (pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = i\}) \quad i \in S_1$$

we obtain

$$U_0 = (I - P^{[00]})^{-1} P^{[01]} U_1 = M U_1$$

and the first result follows. The second result is obtained summing over n.

Proof of Lemma 2:

For $n = 0$, the two quantities are trivially equal to 0. For $n \geq 1$, the proof is made by induction on n. Let $i \in S_T$, then, for $n = 1$,

$$pr\{\hat{X}(x) \in S_A\ ,\ \hat{N}(\infty) = 1 \mid \hat{X}_0 = i\} = \sum_{j \in S_A} \hat{Q}_{i,j}(x)$$

and $\quad pr\{Y(\infty) \le x\ ,\ N(\infty) = 1 \mid X_0 = i\}$

$$= \sum_{h=1}^{+\infty} pr\{X_h \in S_A\ ,\ \forall n\ 0 < n < h\ X_n \in S_0\ ,\ V_0 \le \frac{x}{r_i} \mid X_0 = i\}$$

$$= pr\{X_1 \in S_A\ ,\ V_0 \le \frac{x}{r_i} \mid X_0 = i\}$$
$$+ \sum_{h=2}^{+\infty} pr\{X_h \in S_A\ ,\ \forall n\ 0 < n < h\ X_n \in S_0\ ,\ V_0 \le \frac{x}{r_i} \mid X_0 = i\}$$

$$= \sum_{j \in S_A} Q_{i,j}(\frac{x}{r_i})$$
$$+ \sum_{l \in S_0} \sum_{h=2}^{+\infty} pr\{X_h \in S_A\ ,\ \forall n\ 1 < n < h\ X_n \in S_0\ ,\ X_1 = l\ ,\ V_0 \le \frac{x}{r_i} \mid X_0 = i\}$$

$$= \sum_{j \in S_A} Q_{i,j}(\frac{x}{r_i})$$
$$+ \sum_{l \in S_0} \sum_{h=2}^{+\infty} pr\{X_h \in S_A\ ,\ \forall n\ 1 < n < h\ X_n \in S_0 \mid X_1 = l\ ,\ V_0 \le \frac{x}{r_i}\ ,\ X_0 = i\}$$
$$pr\{X_1 = l\ ,\ V_0 \le \frac{x}{r_i} \mid X_0 = i\}$$

$$= \sum_{j \in S_A} Q_{i,j}(\frac{x}{r_i})$$
$$+ \sum_{l \in S_0} Q_{i,l}(\frac{x}{r_i}) \sum_{h=2}^{+\infty} pr\{X_h \in S_A\ ,\ \forall n\ 1 < n < h\ X_n \in S_0 \mid X_1 = l\}$$

$$= \sum_{j \in S_A} Q_{i,j}(\frac{x}{r_i})$$
$$+ \sum_{l \in S_0} Q_{i,l}(\frac{x}{r_i}) \sum_{h=1}^{+\infty} pr\{X_h \in S_A\ ,\ \forall n\ 0 < n < h\ X_n \in S_0 \mid X_0 = l\}$$

$$= \sum_{j \in S_A} Q_{i,j}(\frac{x}{r_i}) + \sum_{j \in S_A} \sum_{l \in S_0} Q_{i,l}(\frac{x}{r_i}) M_{l,j}$$

$$= \sum_{j \in S_A} \hat{Q}_{i,j}(x)$$

Suppose now the lemma true for $1, 2, \cdots, n-1$ and let $n > 1$.

$$\begin{aligned}
& pr\{\hat{X}(x) \in S_A\,,\ \hat{N}(\infty) = n \mid \hat{X}_0 = i\} \\
= & \sum_{j \in S_1} \int_0^x pr\{\hat{X}(x-u) \in S_A\,,\ \hat{N}(\infty) = n-1 \mid \hat{X}_0 = j\} d\hat{Q}_{i,j}(u) \\
= & \sum_{j \in S_T} \int_0^x pr\{Y(\infty) \le x-u\,,\ N(\infty) = n-1 \mid X_0 = j\} d\hat{Q}_{i,j}(u) \\
= & \sum_{j \in S_T} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,j}(u) \\
& + \sum_{j \in S_T} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} \sum_{l \in S_0} M_{l,j} dQ_{i,l}(u) \\
= & \sum_{j \in S_T} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,j}(u) \\
& + \sum_{j \in S_T} \sum_{l \in S_0} M_{l,j} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,l}(u)
\end{aligned}$$

$$\begin{aligned}
& pr\{Y(\infty) \le x\,,\ N(\infty) = n \mid X_0 = i\} \\
= & \sum_{j \in S_1} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,j}(u) \\
& + \sum_{j \in S_0} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,j}(u) \\
= & \sum_{j \in S_T} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,j}(u) \\
& + \sum_{j \in S_0} \int_0^{\frac{x}{r_i}} \sum_{l \in S_T} M_{j,l} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = l\} dQ_{i,j}(u) \quad \text{using Lemma 1} \\
= & \sum_{j \in S_T} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,j}(u) \\
& + \sum_{j \in S_0} \sum_{l \in S_T} M_{j,l} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = l\} dQ_{i,j}(u) \\
= & \sum_{j \in S_T} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,j}(u) \\
& + \sum_{j \in S_T} \sum_{l \in S_0} M_{l,j} \int_0^{\frac{x}{r_i}} pr\{Y(\infty) \le x - r_i u\,,\ N(\infty) = n-1 \mid X_0 = j\} dQ_{i,l}(u)
\end{aligned}$$

Proof of Lemma 3: Distinguish three cases:

If $i \in S_T$,

$$\begin{aligned} C_i(x) &= \sum_{n=0}^{+\infty} pr\{Y(\infty) \leq x\ ,\ N(\infty) = n \mid X_0 = i\} \\ &= \sum_{n=0}^{+\infty} pr\{\hat{X}(x) \in S_A\ ,\ \hat{N}(\infty) = n \mid \hat{X}_0 = i\} \quad \text{using Lemma 2} \\ &= \hat{F}_i(x) \end{aligned}$$

If $i \in S_A$, $C_i(x) = \hat{F}_i(x) = 1$

If $i \in S_0$,

$$\begin{aligned} C_i(x) &= \sum_{l \in S_1} M_{i,l} C_l(x) \quad \text{using Lemma 1} \\ &= \sum_{l \in S_1} M_{i,l} \hat{F}_l(x) \end{aligned}$$

References

[1] M. D. Beaudry, "Performance-related reliability measures for computing systems," *IEEE Transactions on Computers*, vol. C-27, pp. 540–547, June 1978.

[2] J. F. Meyer, "On evaluating the performability of degradable computing systems," *IEEE Transactions on Computers*, vol. C.29, pp. 720–731, August 1980.

[3] A. Goyal, V. F. Nicola, A. N. Tantawi, and K. S. Trivedi, "Reliability of systems with limited repairs," *IEEE Transactions on Reliability*, vol. R-36, p. , June 1987.

[4] M. C. Hsueh, R. K. Iyer, and K. S. Trivedi, "Performability modeling based on real data: a case study," *IEEE Transactions on Computers*, vol. 37, pp. 478–484, April 1988.

[5] E. Cinlar, *Introduction to stochastic Processes.* New-Jersey: Prentice Hall, 1975.

[6] R. M. Smith, K. S. Trivedi, and A. V. Ramesh, "Performability analysis: measures, an algorithm, and a case study," *IEEE Transactions on Computers*, vol. 37, pp. 406–417, April 1988.

[7] R. A. McLean and M. F. Neuts, "The integral of a step function defined on a semi-Markov process," *SIAM Journal of Applied Mathematics*, vol. 15, p. , May 1967.

[8] V. G. Kulkarni, V. F. Nicola, and K. S. Trivedi, "The completion time of a job on multi-mode systems," *Advances in Applied Probability*, vol. , p. , December 1987.

[9] V. G. Kulkarni, "A new class of multivariate phase type distributions," Tech. Rep., University of North Carolina, Chapel Hill, NC, USA, May 1986.

[10] M. Ajmone Marsan, G. Balbo, and G. Conte, "A class of Generalized Stochastic Petri Nets for the performance evaluation of multiprocessor systems," *ACM Transactions on Computer Systems*, vol. 2, pp. 93–122, May 1984.

[11] A. Bobbio and K. S. Trivedi, "An aggregation technique for the transient analysis of stiff Markov chains," *IEEE Transactions on Computers*, vol. C-35, pp. 803–814, September 1986.

[12] R. Marie and B. Sericola, "Distribution du temps total de séjour dans un sous-ensemble d'états transitoires d'un processus markovien homogène à espace d'état fini," Tech. Rep. 585, I.N.R.I.A. - I.R.I.S.A. Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France, November 1986.

[13] U. N. Bhat, *Elements of Applied Stochastic Processes.* : John Wiley & Sons, 1984.

[14] D. E. Knuth, *The Art of Computer Programming, Vol.1.* : Addison-Wesley, 1973.

[15] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms.* : Menlo Park, CA, USA: Addison-Wesley Publishing Company, 1974.

[16] S. Pissanetzky, *Sparse Matrix Technology.* : Orlando, FL, USA: Academic Press, Inc., 1984.

[17] R. Geist, M. Smotherman, K. S. Trivedi, and J. Bechta Dugan, "The reliability of life-critical computer systems," *Acta Informatica*, vol. 23, pp. 621–642, 1986.

[18] A. Reibman and K. S. Trivedi, "Numerical transient analysis of Markov models," *Computers and Operations Research*, vol. , p. , 1987.

[19] D. L. Jagerman, "An inversion technique for the Laplace transforms," *Bell System Technical Journal*, vol. 61, pp. 1995–2002, September 1982.

[20] R. A. Sahner and K. S. Trivedi, "SHARPE: symbolic hierarchical automated reliability and performance evaluator," Tech. Rep., Duke University, Durham, NC, USA, September 1986.

[21] M. C. Hsueh, R. K. Iyer, and K. S. Trivedi, "A measurement-based performability model for a multiprocessor system," in *Proceedings of the 2nd International Workshop on Applied Mathematics and Performance/Reliability Models of Computer/Communication Systems, (Rome, Italy)*, p. , May 1987.

[22] C. Moler, "MATLAB User's Guide," Tech. Rep., Univeristy of New Mexico, 1981.

[23] S. Ross, *Stochastic Processes.* John Wiley and Sons, 1983.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique