European Journal of Operational Research 40 (1989) 257–267
North-Holland

Theory and Methodology

Markov and Markov reward model transient analysis: An overview of numerical approaches *

Andrew REIBMAN **, Roger SMITH † and Kishor TRIVEDI
Department of Computer Science, Duke University, Durham, NC 27706, USA

Abstract: The advent of fault-tolerant, distributed systems has led to increased interest in analytic techniques for the prediction of reliability, availability, and combined performance and reliability measures. Markov and Markov reward models are common tools for fault-tolerant system reliability prediction. In this paper, we first derive instantaneous and cumulative measures of Markov and Markov reward model behavior. We then compare the complexity of several competing algorithms for the computation of these measures. Better approaches for Markov model solution should lead to more effective techniques for fault-tolerant system modeling.

Keywords: Markov chains, Markov reward models, transient analysis, performability models, sensitivity analysis, performance modeling, reliability modeling

1. Introduction

In recent years, multiple processor and distributed computer systems have grown in size and complexity. Computer system designers face the task of evaluating large numbers of design alternatives. When evaluating a system, one can consider many measures including reliability, availability, performance, and cost. Our research focuses on reliability and similar measures.

There are three general techniques for predicting system reliability: measurement, simulation, and analytical modeling. Extensive measurement is most accurate, but it is expensive. Furthermore, selecting an architectural alternative during system design precludes the use of actual measurements. Simulation models can also be expensive, and they may require a large amount of computation time to yield statistically significant results. Analytic modeling is an attractive alternative to measurement and simulation, particularly if many different designs need to be evaluated. Analytic reliability models include combinatorial models (e.g., Chapters 1–5 of [29]), Markov chains [29], semi-Markov and Markov renewal processes [23], and Markov reward models [13].

Combinatorial models such as fault-trees and reliability block diagrams can be used to analyze many systems. However, important interdependencies and dynamic relationships among system components are easily lost when using these models. Markov models can capture more complex attributes of system behavior. This paper provides an overview of some recent work on solving Markov models.

* This work was supported in part by the Army Research Office under contract No. DAAG29-84-0045, and by the Air Force Office of Scientific Research under grant No. AFOSR-84-0132.
** Now with AT&T Bell Laboratories, Holmdel, NJ 07733.
† Now with Computer Science Department, Yale University, New Haven, CT.
Received September 1987; revised June 1988

1.1. Research goals and paper organization

Important goals in Markov reliability modeling research include:

- Ease of use—Make Markov models easy for designers to use.
- Choice of metrics—Identify and compute the right measures for an application.
- Minimize computation costs—Avoid large computation and storage costs.
- Error analysis—Analyze errors caused by modeling assumptions, inexact parameter values, and numerical approximation.
- Reliability bottleneck analysis—Help identify where to spend additional effort improving a design.

The goal of ease of use is best met with computer software packages that either automatically convert system design specifications into a model for analysis, or provide a powerful user interface for model specification. (A discussion of the implementation of these software packages is beyond the scope of this paper. For examples of model specification techniques, the interested reader is referred to [24] and [30].)

Once a Markov or Markov reward model of a system is specified, many different measures can be analyzed. Performance models are usually evaluated using measures based on steady-state (equilibrium) probabilities. In contrast, reliability modeling requires the computation of transient state probabilities or cumulative state probabilities. We define several different reliability measures of interest in Section 1.2.

Choosing an appropriate numerical solution algorithm helps achieve the goal of efficient, accurate model solution. In Section 2, we discuss methods for computing the state probability vector of a continuous-time Markov chain. The main focus in this discussion is the influence of model size and stiffness on the required computation time.

Other measures may be derived from integrals of the state probabilities. The computation of these 'integral' or 'cumulative' measures will be considered in Section 3. In Section 4, we extend the class of 'cumulative' measures further. We consider computing the distribution of accumulated reward. This distribution cannot be obtained as the weighted sum of state probabilities or as the weighted sum of integrals of state probabilities. In Section 5, we consider the computation of the distribution of interval availability, an important special case of the general problem of computing the distribution of accumulated reward.

In Section 6, we discuss parametric sensitivity analysis, a technique that is useful both for the analysis of modeling errors and for system weak link identification.

In section 7, we give some concluding remarks.

1.2. Mathematical preliminaries

Before proceeding with the discussion of numerical algorithms, we introduce some definitions and notation. A system's state at time t can be described by a continuous-time Markov chain (CTMC), $\{X(t), t \geq 0\}$, with discrete state space Ω. We will assume that the state space is finite and of size N. We let q_{ij}, $i \neq j$, be the rate of transition from state i to state j. Define $q_{ii} = -\sum_{j \neq i} q_{ij}$. The matrix $\boldsymbol{Q} = [q_{ij}]$ is the 'infinitesimal generator' of the CTMC. We let η denote the number of nonzero entries in $\boldsymbol{Q}$. If $P_i(t)$ is the probability the CTMC is in state i at time t, the row vector $\boldsymbol{P}(t)$ is called the 'state probability vector' of the CTMC. The system of Kolmogorov differential equations describes the behavior of the state probability vector as a function of t:

$$\dot{\boldsymbol{P}}(t) = \boldsymbol{P}(t)\boldsymbol{Q}, \quad \boldsymbol{P}(0) = \boldsymbol{P}_0. \tag{1}$$

Some measures of system behavior can be derived using weighted sums of state probabilities. As an example consider a queueing system. If state i corresponds to having i jobs in the system, then the average number of jobs in the system at time t is $\sum_i i P_i(t)$. For an availability model, we partition the set of states, Ω, into Ω_0, the set of operational system states, and Ω_f, the set of failed or down states. We define the indicator random variable

$$I(t) = \begin{cases} 1 & \text{if } X(t) \in \Omega_0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The probability that the system is operational at time t is the 'point (instantaneous) availability', defined as

$$\mathrm{PAV}(t) = \Pr[I(t) = 1] = E[I(t)] = \sum_{i \in \Omega_0} P_i(t). \tag{3}$$

'System reliability' can be defined as

$$R(t) = \Pr[I(u) = 1, \forall u \in [0, t]]. \quad (4)$$

If we treat Ω_f as a class of absorbing states, $R(t)$ is $1 - \Pr$[absorption during $[0, t)$]. This quantity can be computed by deleting all arcs leading from states in Ω_f to states in Ω_0, and then computing $R(t) = \sum_{i \in \Omega_0} P_i(t)$.

2. Computation of state probabilities

In this section, we discuss numerical techniques for computing the transient state probability vector of a CTMC. In computer system modeling, large, sparse Markov chains are common. Models need to be reduced as much as possible before model solution begins. Large models can be handled by using efficient solution methods and sparse matrix techniques. Some special classes of models (e.g., acyclic models) may be solved by particularly efficient algorithms.

Regardless of its size, a Markov chain model may be stiff. A stiff model includes events that occur rapidly relative to the length of the solution interval; stiffness depends not only on the transition rates of the Markov chain, but also on length of time that is considered [21]. One indicator of stiffness is the product of the length of the solution interval and the size of the largest total exit rate from any state of the model. Let $q = \max_i |q_{ii}|$ and let t be the time interval for the transient solution. We call qt the 'index of stiffness' of a CTMC problem.

The general solution of the Kolmogorov system of differential equations (1) is given by

$$\boldsymbol{P}(t) = \boldsymbol{P}_0 \, e^{\boldsymbol{Q}t}, \quad (5)$$

where $e^{\boldsymbol{Q}t}$ is the well-known 'matrix exponential' and is defined by the infinite series

$$e^{\boldsymbol{Q}t} = \boldsymbol{I} + \boldsymbol{Q}t + \boldsymbol{Q}^2t^2/2! + \boldsymbol{Q}^3t^3/3! + \cdots. \quad (6)$$

For state i, the general solution can also be written in the form

$$P_i(t) = \sum_{j=1}^{d} \sum_{k=0}^{m_j - 1} a_{ijk} t^k \, e^{-\lambda_j t}, \quad (7)$$

where d is the number of distinct eigenvalues of $\boldsymbol{Q}$, λ_j is the j-th distinct eigenvalue of $\boldsymbol{Q}$, m_j is the multiplicity of λ_j, and a_{ijk} is a constant.

First, consider the special case of acyclic Markov models. Here, the eigenvalues are obtainable by inspection; they are precisely the diagonal elements of the generator matrix $\boldsymbol{Q}$. Based on the convolution integration approach [29], an $O(N^2)$ algorithm has been developed to solve the chain [17]. In the ACE package, we further exploit sparseness of the $\boldsymbol{Q}$ matrix to obtain an $O(\eta)$ algorithm [17], where η is the number of non-zero entries in the matrix $\boldsymbol{Q}$.

Provided we determine all the eigenvalues of $\boldsymbol{Q}$, the closed-form solution (7) can be used to solve cyclic Markov chains. The QR algorithm [32] will determine all the eigenvalues in $O(N^3)$ time. Subsequently, the coefficients a_{ijk} can be determined by the solution of a linear system of equations. By organizing the computation appropriately, an algorithm with an overall execution time of $O(N^3)$ has been derived [28].

The advantage of the two solution methods described above is that they both determine a closed-form expression for state probabilities as functions of the time variable t. These methods are used in the SHARPE package, where closed-form solutions are required [24]. A drawback of the closed-form method for cyclic Markov chains is that the computation of the coefficients a_{ijk} in equation (7) requires the use of a full matrix. Consequently, methods that use (7) to solve large (≥ 400 states) problems encounter severe space and run-time limitations. We now discuss methods that exploit sparsity and hence are suitable for solving large cyclic Markov models. These methods provide a numerical solution, not a closed-form solution.

For non-stiff Markov models, uniformization (used in the SAVE package [8]) and Runge–Kutta–Fehlberg (used in the HARP package [30]) provide an accurate numerical solution [21]. Runge–Kutta is a fourth-order, explicit ordinary differential equation solution technique. Uniformization is a series technique based on the Taylor series expansion for the matrix exponential in (6). However, uniformization first transforms $\boldsymbol{Q}$ to $\boldsymbol{Q}^* = \boldsymbol{Q}/q + \boldsymbol{I}$. The solution is then given by the infinite series

$$\boldsymbol{P}(t) = \sum_{i=1}^{\infty} \boldsymbol{P}(0)(\boldsymbol{Q}^*)^i \, e^{-qt} \frac{(qt)^i}{i!}$$

$$= \sum_{i=0}^{\infty} \boldsymbol{\Pi}(i) \, e^{-qt} \frac{(qt)^i}{i!}, \quad (8)$$

where

$$\Pi(i+1) = \Pi(i)\boldsymbol{Q}^*, \quad \Pi(0) = \boldsymbol{P}(0). \tag{9}$$

Because the matrix $\boldsymbol{Q}^*$ is non-negative, this aproch requires no subtractions. In a practical implementation, the series will be truncated after term k. We can choose k to satisfy a specified truncation error bound:

$$\varepsilon(k) \leqslant 1.0 - \sum_{i=0}^{k} e^{-qt}\frac{(qt)^i}{i!}. \tag{10}$$

The right hand sum above is a Poisson cumulative distribution function with mean qt. So, the normal approximation to the Poisson distribution with standard deviation $\sqrt{qt}$ implies that, for a fixed ε and large qt, $k = qt + c\sqrt{qt} = O(qt)$. In a full matrix implementation, each term of the series in (8) requires $O(N^2)$ operations. The total computation time of uniformization is $O(N^2qt)$. With a sparse matrix implementation, $O(\eta qt)$ is the total computation time for uniformization.

Explicit integration methods have a similar dependence on qt. The dependence on t comes from the need to perform $O(t/\Delta t)$ integration steps each of length Δt. The size of Δt must be inversely proportional to the largest eigenvalue of Q in order for an explicit method to be stable. The upper bound of largest eigenvalue of Q is directly related to q by the Gerschgorin circle theorem. Thus sparse implementations of explicit integration methods in the worst case have computation times of $O(\eta t/\Delta t) = O(\eta qt)$.

For stiff models, special techniques are required to obtain numerically stable algorithms. An implicit integration method that uses the Trapezoid Rule and a backward difference formula (TR-BDF2) is discussed in [21]. In particular, while conventional techniques (e.g., uniformization or Runge–Kutta) markedly degrade as the mission time or the magnitude of the largest transition rate increases, stable numerical integration techniques suffer little or no performance degradation. Because there is no lower bound on the step size from stability requirements, the step size can be changed adaptively without regard for model stiffness. The execution-time requirement of such a stiff solver is given by $O(N^2)$ in the full matrix case and $O(\eta)$ in the sparse matrix case. It should be noted that the overhead required by stiff solvers is only justified for stiff problems ($qt \geqslant 1000$) [21].

Another technique for the transient analysis of stiff Markov chains is the decomposition/aggregation method proposed by Bobbio and Trivedi [3]. This technique both reduces the size of the problem by aggregation *and* eliminates stiffness. The approximation technique works best on systems with to two well-separated sets of transition rate values. This situation is common in fault-tolerant computer systems, where failure rates are typically orders of magnitude smaller than repair rates. So, the rates are classified as fast or slow rates. States of the Markov chain are also classified into fast and slow states. A state is fast if at least one outgoing transition is fast; otherwise, the state is a slow state. The decomposition of the matrix is based on the classification of the fast states into nearly-completely decomposable subsets and into a nearly-transient (or unilaterally coupled) subset of states. An appropriate aggregation algorithm is separately applied to each subset so that the final transient approximate solution is obtained by integrating a smaller, non-stiff set of linear differential equations. Let α be the ratio of the smallest fast rate to the largest slow rate. Then the execution time of the aggregation method is given by $O(N^2qt/\alpha)$.

Table 1 gives a summary of the execution time behavior of the algorithms discussed in this section. The '—' denotes the full-matrix nature of

Table 1
State probability vector computation

	Method	Package	Full storage	Sparse storage	Reference
Acyclic Markov chains	Convolution Integration	ACE	$O(N^2)$	$O(\eta)$	[17]
Cyclic Markov chains	Closed-form solution	SHARPE	$O(N^3)$	–	[28]
	Explicit integration	HARP	$O(N^2qt)$	$O(\eta qt)$	[21]
	Uniformization	SAVE	$O(N^2qt)$	$O(\eta qt)$	[21]
	Implicit integration	TR-BDF2	$O(N^2)$	$O(\eta)$	[21]
	Aggregation method	BT	$O(N^2qt/\alpha)$	?	[3]

closed-form solution, while the '?' denotes a case that has not yet been investigated.

3. Computation of cumulative measures

We now provide detailed discussion of 'cumulative measures' such as 'expected interval availability' [8] and 'expected accumulated reward' [15,17,26]. In general, expected values of cumulative measures can be computed using approaches similar to those presented for the state probability problem [21].

Of interest is the cumulative amount of operational time (up time) of the system during $(0, t)$, defined as

$$O(t) = \int_0^t I(u)\,\mathrm{d}u. \tag{11}$$

From this we derive the 'interval availability', the fraction of time the system was in operation in $(0, t)$, computed as

$$\mathrm{IAV}(t) = \frac{O(t)}{t}.$$

Therefore, the 'expected interval availability' $E[\mathrm{IAV}(t)]$ is given by

$$E[\mathrm{IAV}(t)] = \frac{1}{t}\sum_{i\in\Omega_0}\int_0^t P_i(u)\,\mathrm{d}u. \tag{12}$$

If we further associate with each state i a reward rate r_i, the 'accumulated reward' in the interval $(0, t)$ is

$$Y(t) = \int_0^t r_{X(u)}\,\mathrm{d}u. \tag{13}$$

Thus the 'expected accumulated reward' in the interval $(0, t)$ is given by

$$E[Y(t)] = \sum_{i\in\Omega}\int_0^t r_i P_i(u)\,\mathrm{d}u. \tag{14}$$

The cumulative state probabilities can be computed using uniformization [8, 22]. If we define $\boldsymbol{L}(t) = \int_0^t \boldsymbol{P}(u)\,\mathrm{d}u$, then integrating (8) yields

$$\boldsymbol{L}(t) = \frac{1}{q}\sum_{i=0}^{\infty}\Pi(i)\sum_{j=i+1}^{\infty}\mathrm{e}^{-qt}\frac{(qt)^j}{j!}. \tag{15}$$

This series can be used to compute expected uptime, expected accumulated reward, and, with the addition of a $1/t$ factor, time-averaged cumulative measures.

Alternatively, we can compute cumulative state probabilities with numerical differential equation solution. A direct integration of (1) yields a new set of differential equations:

$$\dot{\boldsymbol{L}}(t) = \boldsymbol{L}(t)\boldsymbol{Q} + \boldsymbol{P}_0, \quad \boldsymbol{L}(0) = \boldsymbol{0}. \tag{16}$$

We can compute the desired cumulative measures using

$$E[\mathrm{IAV}(t)] = \frac{1}{t}\sum_{i\in\Omega_0} L_i(t), \tag{17}$$

and

$$E[Y(t)] = \sum_{i\in\Omega} r_i L_i(t). \tag{18}$$

Contrasting (1) with (16) we see that (16) is inhomogeneous because of the forcing function, $\boldsymbol{P}_0$. However, the forcing function is time-independent, and hence does not greatly complicate finding the solution. Methods of solution for (16) are similar to those discussed for (1). In general, the run-times for cumulative measure analysis are the same as those given in Table 1 for the instantaneous measure case. For further details in the acyclic case see [17], and for the cyclic case see [22].

4. Computation of the distribution of accumulated reward

In a Markov model, each model state corresponds to a system state. Markov reward models also associate non-negative real-valued reward rate with each state. Often a state is characterized by a set of operational components, while the reward rate is a performance level associated with the given set of operational components. The CTMC, $X(t)$, will be referred to as the structure-state process. The vector of rewards, $\boldsymbol{r}$, associated with the states in $X(t)$ are called the reward structure.

Under general assumptions about the stochastic process $\{X(t),\ t \geqslant 0\}$ and the reward structure $\boldsymbol{r}$, Howard [13] studied the expected accumulated reward $E[Y(t)]$ for finite intervals of time and the expected time-averaged accumulated reward $E[Y(t)/t]$ over an infinite time interval. Iyer et al. [14] developed a method to determine the central

moments of the distribution of accumulated reward for cyclic Markov reward models. If all the eigenvalues of the generator matrix $\boldsymbol{Q}$ are distinct and the first k moments of the distribution of accumulated reward are computed, the time complexity of their algorithm is $O(N^4k^2)$.

Often, we are interested in the behavior of $\mathscr{Y}(x, t)$ far from the mean. For example, when a system is required to have a high probability of delivering a specific total reward, the central moments do not provide accurate information. As an alternative, we consider computing the distribution of $Y(t)$. We denote the complementary distribution of accumulated reward at time t, evaluated at x, as

$$\mathscr{Y}(x, t) \equiv \Pr[Y(t) \geqslant x].$$

Computing this distribution can illuminate effects that are not detected by steady-state values, instantaneous measures, or expected values of cumulative measures [25, 26].

Two special cases of this problem have been studied. In one case, the structure-state process is restricted, and in the other the reward structure is constrained. The first special case includes acyclic CTMCs. Beaudry [1] considered the limiting distribution for acyclic CTMC, while others [6, 10, 19] treated the acyclic problem for finite time intervals. Both [6] and [10] require $O(N^3)$ time. An example of the second special case of problems is the computation of the distribution of interval availability. Here the reward structure is specialized to the '0–1' case. In the next section, we discuss this special case in more detail. In the rest of this section, we will describe algorithms developed to evaluate the distribution of accumulated reward for cyclic and acyclic Markov reward models. Three computational approaches are compared: uniformization, Laplace transform inversion, and partial differential equation solution.

The uniformization method, recently generalized in [4], requires exponential time in the number of distinct reward rates. Nevertheless, uniformization may still be an attractive solution method for the interval availability problem, which has only two reward rates. We discuss the computation of interval availability in the next section.

For a general CTMC and arbitrary reward structure, [20] derived a linear system in the double Laplace transform of the distribution of accumulated reward. The numerical solution of the double transform system is a difficult numerical problem that we consider in [15] and [26]. We have developed an $O(N^3)$ algorithm to evaluate the distribution of accumulated reward for cyclic and acyclic Markov reward models.

We use a two-phase transform inversion of the linear system derived in [20] and [16]:

$$(s\boldsymbol{I} + u\boldsymbol{R} - \boldsymbol{Q})\tilde{\mathscr{Y}}^*(u, s) = \boldsymbol{e}. \tag{19}$$

$\tilde{\mathscr{Y}}^*(u, s)$ is $\mathscr{Y}(x, t)$ with a Laplace–Stieltjes Transform ($\sim$) taken with respect of x, followed by a Laplace Transform ($*$) taken with respect to t. The matrix of reward rates is

$$\boldsymbol{R} = \text{diag}\,[r_1, r_2, \ldots, r_i, \ldots, r_n].$$

$\boldsymbol{Q}$ is the generator matrix of the CTMC, and $\boldsymbol{e}$ is a column vector of size N with all elements equal to 1. Using a partial-fraction expansion and the eigenvalues of $\boldsymbol{Q} - u\boldsymbol{R}$, the first phase is to invert analytically with respect to the time transform variable, s. Then rational approximations of a Fourier series are used to invert numerically with respect to the reward transform variable, u. An improvement over the other polynomial-time method [15] for solving both cyclic and acyclic models is obtained in the first phase of the algorithm by re-using work already obtained during the computation of the eigenvalues. The computational cost of the method is $O(mN^3)$ where m is the number of approximating terms needed for the second phase (numerical inversion) [26].

The density of accumulated reward $y_i(x, t) = \mathrm{d}/\mathrm{d}x\{\mathscr{Y}_i(x, t)\}$ has been derived as [25],

$$\boldsymbol{R}\frac{\partial \boldsymbol{y}}{\partial x} + \frac{\partial \boldsymbol{y}}{\partial t} = \boldsymbol{Q}\boldsymbol{y}. \tag{20}$$

This linear hyperbolic system may be solved by explicit finite-difference methods. These methods also depend on qt. The flux-limiting approach used in [31] preserves the non-negativity of $\mathscr{Y}(x, t)$ and is second order accurate. Since $O(t/\Delta t)$ integration steps are performed, the computation time depends on t. Because an upper bound on the largest eigenvalue of $\boldsymbol{Q}$ is directly related to q by the Gerschgorin circle theorem, and the size of Δt must be inversely proportional to the largest eigenvalue of $\boldsymbol{Q}$ in order for the solution method to converge to the correct result, $O(t/\Delta t)$ time steps must be taken. Since the number of points

Table 2
Polynomial-time computation of the distribution of accumulated reward

	Method	Full storage	Sparse storage	Reference
Acyclic CTMC	Time domain recursion	$O(N^3)$	$O(N^3)$	[6,10]
Acyclic or cyclic CTMC	Double transform inversion	$O(mN^4)$	–	[15]
	Double transform inversion	$O(mN^3)$	–	[26]
	Partial differential equation	$O(N^2(qt)^2)$	$O(\eta(qt)^2)$	[25]

evaluated at each time step is directly proportional to t, the number of mesh points to be determined is $O((t/\Delta t)^2)$. Thus, for sparse implementations of finite-difference methods, we have a worst case computation time of $O(\eta(t/\Delta t)^2) = O(\eta(qt)^2)$. Note that, in contrast with (19), using (20) computes the densities $y_i(x, t)$ rather than the distribution function $\mathcal{Y}_i(x, t)$. However, by summing $y_i(x, t)$ as it is computed, we can obtain the distribution function $\mathcal{Y}_i(x, t)$ at little additional cost.

For small problems, especially over long time intervals, the double transform inversion method is probably best. For large problems (≥ 400 states), the ability to take advantage of sparsity becomes more important. Because the double transform inversion method is a full-matrix computation, partial differential equation solution is especially attractive for large problems of moderate stiffness.

Table 2 summarizes the results of this section. The "—" denotes the full-matrix nature of the double-transform inversion method. Since, for this problem, the uniformization [4] requires exponential time in the number of distinct reward rates, it is not included in the table.

5. Computation of the distribution of interval availability

The market for computer systems with guaranteed levels of availability is growing. Many vendors have announced computer systems with guaranteed levels of availability in the 95 to 100 percent range. Substantial penalties are incurred if the guaranteed level of availability is not met. Thus it is important for vendors to be able to assess a system designs' ability to provide high interval availability. Consumers must be able to assess the value of various guarantees since the penalties, though substantial, do not fully compensate for inadequate service. The distribution of interval availability, $\Pr[\mathrm{IAV}(t) \geq x]$, will be a measure of interest to both producers and consumers of highly-available computer systems.

An important characteristic of interval availability is that the required values of $\mathrm{IAV}(t)$ will be nearly 1.0. Even for simple cases there are no closed-form solutions. For example, for a two-state Markov model, the distribution of interval availability is an infinite sum of Bessel functions. Consequently, we will describe numerical methods for determining the distribution of $\mathrm{IAV}(t)$. We include a section on this measure because of its wide use, and because the specialized reward structure of the interval availability problem has interesting computational implications.

As in the previous section, the three methods we mention are discrete approximation to a system of partial differential equations, transform inversion, and uniformization. The interval availability problem can be described by a linear hyperbolic system of partial differential equations [9]:

$$\frac{\partial y_i}{\partial x} + \frac{\partial y_i}{\partial t} = \boldsymbol{Q} y_i, \quad i \in \Omega_0,$$

and

$$\frac{\partial y_i}{\partial t} = \boldsymbol{Q} y_i, \quad i \in \Omega_f.$$

This system can be solved by explicit finite-difference methods. Since the guaranteed level, γ, of $\mathrm{IAV}(t)$ is nearly 1.0 [9], only approximately $(1-\gamma)((t/\Delta t)^2)$ points must be evaluated. Hence, for most problems of interest, the total computational effort required is only $O(N^2(1-\gamma)((t/\Delta t)^2))$. Since the stepsize Δt is inversely related to the maximum exit rate from a state, we have $O(N^2(1-\gamma)(qt)^2)$ as a bound for the computational complexity of the method.

The computational complexity of the transform inversion method, $O(mN^3)$, is unchanged by the

Table 3
Polynomial-time computation of the interval availability distribution

Method	Full storage	Sparse storage	Reference
Double transform inversion	$O(mN^3)$	–	[26]
Partial differential equation	$O(N^2(1-\gamma)(qt)^2)$	$O(\eta(1-\gamma)(qt)^2)$	[9]
Uniformization	$O(N^2(qt)^2)$	$O(\eta(qt)^2)$	[9,5]

specialized reward structure of the interval availability problem. Because the inversion method determines all the eigenvalues of $sI + uR - Q$, a non-uniform R is actually preferable, since it tends to separate the eigenvalues and make their determination by the QR algorithm more accurate.

Uniformization takes the original CTMC and transforms it into an equivalent stochastic matrix. First choose q greater than or equal to the maximum exit rate from all states. We then apply the transformation $Q \to Q^* = I + Q/q$. This has favorable numerical properties [11]. The computation is then

$$\begin{aligned}\Pr[\mathrm{IAV}(t) \geqslant \gamma t] &= \sum_{m=0}^{\infty} \Pr[M(t)=m] \\ &\quad \times \sum_{k=0}^{m+1} \Pr[K=k \mid M(t)=m] \\ &\quad \cdot \Pr[\mathrm{IAV}(t) \geqslant \gamma t \mid M(t)=m,\ K=k], \quad (21)\end{aligned}$$

where $M(t)$ is a Poisson process with rate q, and K is the number of sub-intervals spent in an operational state. It can be shown that the m points in $(0, t)$ are distributed as the order statistics of m uniform random variables. Consequently the subintervals themselves are interchangeable [23]. We know that

$$\Pr[M(t)=m] = \mathrm{e}^{-qt}\frac{(qt)^m}{m!}, \quad m=0,1,\ldots,$$

and putting the K operational subintervals first we have

$$\begin{aligned}&\Pr[\mathrm{IAV}(t) \geqslant \gamma t \mid M(t)=m,\ K=k] \\ &\quad = \sum_{i=k}^{m} \binom{m}{i} \gamma^i (1-\gamma)^{m-i}.\end{aligned}$$

The only difficult quantity is $\Pr[K=k \mid M(t)=m]$, which for convenience we denote as $\Pr[k \mid m]$. We can recursively evaluate this quantity:

$$\Pr_i[k \mid m] = \sum_{j \in \Omega} \Pr_j[k-1 \mid m-1] q^*_{j,i}, \quad i \in \Omega_0, \quad (22a)$$

and

$$\Pr_i[k \mid m] = \sum_{j \in \Omega} \Pr_j[k \mid m-1] q^*_{j,i}, \quad i \in \Omega_f. \quad (22b)$$

Since equation (21) is an infinite sum we truncate after m_t events, we can estimate the resulting global error

$$\epsilon(m_t) \leqslant 1.0 - \sum_{i=0}^{m_t} \mathrm{e}^{-qt}\frac{(qt)^i}{i!}. \quad (23)$$

Using this error bound we can guarantee the desired level of accuracy a priori. The choice of m_t is equivalent to the choice of Δt in the finite difference solution of the partial differential equation. Since m_t is $O(qt)$, each evaluation of $\Pr[k \mid m]$ in (22) is $O(m)$, and the recurrence in (22) computes a matrix multiply when carried out for all i. Consequently the computational complexity of the uniformization method is $O(N^2(qt)^2)$, and when we take advantage of sparsity, $O(\eta(qt)^2)$. In Table 3 we tabulate the results of this subsection. The '—' denotes the full-matrix nature of double-transform inversion.

6. Parametric sensitivity analysis

In addition to computing the measures we have already derived, it is often interesting to determine the performance or reliability 'bottleneck' of a system [7]. Towards this end, it is desirable to evaluate system sensitivities, the derivaties of system measures with respect to various model

parameters. Parameters with large sensitivities usually deserve close attention in the quest to improve system characteristics. With proper scaling, these sensitivities can be used to guide system optimization procedures [2].

Sensitivity analysis can also be used to identify parts of a model prone to error. When models are used for system evaluation, the analyst may decide to ignore certain features of the system such as structure, workload, fault-occurrence behavior, or fault/error-handling behavior. The system analyst needs to be aware of both errors in the model and errors in the model solution process. Because the parameters of a model are approximate, a model solution that is highly sensitive to changes in a parameter may indicate potentially serious inaccuracies in the model. It is often possible to analyze the effect of such parametric errors in reliability prediction models [18, 30] and in availability models [8]. A practical application of sensitivity analysis to the analysis of a multiprocessor system is discussed in [2].

We assume that some transition rates q_{ij} are functions of some parameter λ. We want to compute the derivative of various measures with respect to λ (e.g., $\partial P_i(t)/\partial\lambda$) for a given value of λ.

If we let $\boldsymbol{S}(t)$ be the row vector of the sensitivities $\partial P_i(t)/\partial\lambda$, then from (1) we obtain [27],

$$\dot{\boldsymbol{S}}(t) = \boldsymbol{S}(t)\boldsymbol{Q} + \boldsymbol{P}(t)\boldsymbol{V}, \tag{24}$$

where $\boldsymbol{V}$ is the derivative of $\boldsymbol{Q}$ with respect to λ. Assuming the initial conditions do not depend on λ, we have

$$\boldsymbol{S}(0) = \frac{\partial \boldsymbol{P}(0)}{\partial\lambda} = \lim_{t\to 0}\frac{\partial \boldsymbol{P}(t)}{\partial\lambda} = 0.$$

We can then solve (1) and (24) simultaneously,

$$(\dot{\boldsymbol{P}}(t),\ \dot{\boldsymbol{S}}(t)) = (\boldsymbol{P}(t),\ \boldsymbol{S}(t))\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{V}\\ \boldsymbol{0} & \boldsymbol{Q}\end{bmatrix}, \tag{25a}$$

$$(\boldsymbol{P}(0),\ \boldsymbol{S}(0)) = (\boldsymbol{P}_0,\ \boldsymbol{0}). \tag{25b}$$

This expanded system of differential equations can be solved using an explicit [27] or an implicit method of integration. For acyclic CTMC, we can use the the ACE algorithm [17]. Finally, uniformization can be used for sensitivity analysis [12]. If we let $\boldsymbol{Q}^* = \boldsymbol{Q}/q + \boldsymbol{I}$, then

$$\boldsymbol{S}(t) = \frac{\partial}{\partial\lambda}\sum_{i=0}^{\infty}\boldsymbol{P}(0)(\boldsymbol{Q}^*)^i\,\mathrm{e}^{-qt}\frac{(qt)^i}{i!}$$

$$= \sum_{i=0}^{\infty}\Pi(i)'\,\mathrm{e}^{-qt}\frac{(qt)^i}{i!}, \tag{26}$$

where

$$\Pi(i)' = \frac{\partial}{\partial\lambda}\Pi(i) = \frac{\partial}{\partial\lambda}(\Pi(i-1)\boldsymbol{Q}^*)$$

$$= \Pi(i+1)'\boldsymbol{Q}^* + \Pi(i-1)'\frac{\partial}{\partial\lambda}\boldsymbol{Q}^*. \tag{27}$$

Mathematical details for this approach are given in [12].

Similarly, we can derive the sensitivity of $E[Y(t)]$,

$$\frac{\mathrm{d}E[Y(t)]}{\mathrm{d}\lambda} = \sum_{i\in\Omega} r_i\int_0^t\frac{\mathrm{d}P_i(x)}{\mathrm{d}\lambda}\,\mathrm{d}x$$

$$= \sum_{i\in\Omega} r_i\int_0^t S_i(x)\,\mathrm{d}x. \tag{28}$$

As in the instantaneous measure case, methods for computing the sensitivity vector, $\int_0^t \boldsymbol{S}(x)\,\mathrm{d}x$, include numerical differential equation solution, the ACE algorithm [17] for acyclic CTMC, and uniformization.

Parametric sensitivities increase the utility of Markov models at little additional cost. Other techniques for increasing the utility of Markov models are under investigation.

7. Conclusion

We have surveyed several techniques for the transient analysis of Markov and Markov reward models. For transient state probability computation, size and stiffness determine the execution time. Efficient algorithms for non-stiff models include uniformization and explicit methods of numerical integration such as the Runge–Kutta–Fehlberg method. For stiff models, either a stable implicit method such as TR-BDF2 or an aggregation technique can be employed. The algorithms for computing instantaneous state probabilities can be extended to compute cumulative measures of CTMC behavior.

When moments of cumulative measures are inadequate, the distribution of accumulated reward for a Markov reward process needs to be computed, a more difficult problem. Uniformization is unsuitable for the general solution of problems with a large number of distinct reward rates, but it is an attractive approch to the more restricted interval availability problem, since error bounds are easily obtained a priori. Two polynomial-time methods that determine the distribution of accumulated reward for general Markov reward models are double-transform inversion and numerical partial differential equation solution. The transform inversion approach is independent of the degree of stiffness and the length of the solution interval, but requires $O(N^3)$ time and $O(N^2)$ space. Solving the system of partial differential equations is quadratically sensitive to the stiffness of the problem, but requires only $O(\eta)$ space.

Extensions to conventional analysis techniques can make Markov and Markov reward models more useful. We discussed one such technique, parametric sensitivity analysis. The results of parametric sensitivity analysis provide guidance for both refining the model and improving the system being modeled.

References

[1] Beaudry, M.D., "Performance related reliability for computer systems", *IEEE Transactions on Computers* C-27 (6) (1978) 540–547.

[2] Blake, J., Reibman, A., and Trivedi, K., "Sensitivity analysis of reliability and performability measures for a multiprocessor system", in: *ACM SIGMETRICS*, Conference on Measurement and Modeling of Computer Systems, 1988.

[3] Bobbio, A., and Trivedi, K.S., "An aggregation technique for the transient analysis of stiff Markov chains", *IEEE Transactions on Computers* C-35(9) (1986) 803–814.

[4] de Souza e Silva, E., and Gail, H.R., "Calculating availability and performability measures of repairable computer systems using randomization", Research Report, IBM T.J. Watson Research Center, 1986.

[5] de Souza e Silva, E., and Gail., H.R., "Calculating cumulative operational time distributions of repairable computer systems", *IEEE Transactions on Computers* 35 (4) (1986) 322–332.

[6] Donatiello, L., and Iyer, R.B., "Analysis of a composite performance reliability measure for fault-tolerant systems", *J. ACM* 34 (1) (1987) 179–199.

[7] Frank, P.M., *Introduction to System Sensitivity Theory*, Academic Press, New York, 1982.

[8] Goyal, A., Lavenberg, S., and Trivedi, K., "Probabilistic modeling of computer system availability", *Annals of Operations Research* 8 (1987) 285–306.

[9] Goyal, A., Tantawi, A., and Trivedi, K., "A Measure of Guaranteed Availability", IBM Research Report RC 11341, IBM T.J. Watson Research Center, 1985.

[10] Goyal, A., and Tantawi, A.N., "Evaluation of performability for degradable computer systems", *IEEE Transactions on Computers* C-36 (6) (1987) 738–744.

[11] Grassmann, W.K., "Transient solution in Markovian queueing systems", *Computers and Operations Research* 4 (1977) 47–56.

[12] Heidelberger, P., and Goyal, A., "Sensitivity analysis of continuous time Markov chains using uniformization", in: *Proceedings of the second international workshop on applied mathematics and performance/reliability models of computer/communication systems*, 1987.

[13] Howard, R.A., *Dynamic Probabilistic Systems, Vol. II: Semi-Markov and Decision Processes*, Wiley, New York, 1971.

[14] Iyer, B.R., Donatiello, L., and Heidelberger, P., "Analysis of performability for stochastic models of fault-tolerant systems", *IEEE Transactions on Computers* C-35 (10) (1986) 902–907.

[15] Kulkarni, V., Nicola, V.F., Smith, R.M., and Trivedi, K.S., "Numerical evaluation of performability measures and job completion time in repairable fault-tolerant systems", in: *Proc. 16th Int. Symp. on Fault-Tolerant Computing, IEEE, Vienna, Austria, 1986.*

[16] Kulkarni, V.G., Nicola, V.F., and Trivedi, K.S., "On modeling the performance and reliability of multi-mode computer systems", *The Journal of Systems and Software* 6(1/2) (1986) 175–183.

[17] Marie, R.A., Reibman, A.L., and Trivedi, K.S., "Transient solution of acyclic Markov chains", *Performance Evaluation* 7 (3) (1987) 175–194.

[18] McGough, J., Smotherman, M., and Trivedi, K.S., "The conservativeness of reliability estimates based on instantaneous coverage", *IEEE Transactions on Computers* 34 (7) (1985) 602–609.

[19] Meyer, J., "Closed-form solutions of performability", *IEEE Transactions on Computers* C-31 (7) (1982) 648–657.

[20] Puri, P.S., "A method for studying the integral functionals of stochastic processes with applications: I. The Markov chain case", *Journal of Applied probability* 8 (1971) 331–343.

[21] Reibman, A.L., and Trivedi, K.S., "Numerical transient analysis of Markov models", *Computers and Operations Research* 15 (1) (1988) 19–36.

[22] Reibman, A.L., and Trivedi, K.S., "Transient analysis of cumulative measures of Markov chain behavior", *Stochastic Models*, 1989, to appear.

[23] Ross, S.M., *Stochastic Processes*, Wiley, New York, 1983.

[24] Sahner, R.A., and Trivedi, K.S., "Reliability modeling using SHARPE", *IEEE Transactions on Reliability* R-36 (2) (1987) 186–193.

[25] Smith, R.M., *Markov Reward Models: Application Domains and Solution Methods*, PhD thesis, Computer Science Dept., Duke University, Durham, NC, 1987.

[26] Smith, R.M., Trivedi, K.S., and Ramesh, A.V., "Performability analysis: measures, an algorithm and a case

study", *IEEE Transactions on Computers* C-37 (4) (1988) 406–417.

[27] Smotherman, M.K., *Parametric Error Analysis and Coverage Approximation in Reliability Modeling,* PhD thesis, Computer Science Dept., Univ. of North Carolina, Chapel Hill, NC, 1984.

[28] Tardif, H., Ramesh, A.V., and Trivedi, K.S., "Closed-form transient analysis of Markov chains", submitted for publication.

[29] Trivedi, K.S., *Probability and Statistics with Reliability, Queueing, and Computer Science Applications,* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[30] Trivedi, K.S., Geist, R., Smotherman, M., and Dugan, J.B., "Hybrid modeling of fault-tolerant systems", *Computers and Electrical Engineering* 11(2/3) (1984) 87–109.

[31] Van Leer, B., "Toward the ultimate difference scheme II: monotonicity and conservation combined in a second order scheme", *Journal of Computational Physics* 14 (1974) 361–370.

[32] Wilkinson, J.H., and Reinsch, C., *Handbook for Automatic Computation, Vol. II: Linear Algebra*, Springer-Verlag, 1971.