# PROBABILISTIC MODELING OF COMPUTER SYSTEM AVAILABILITY

A. GOYAL and S.S. LAVENBERG

*I.B.M. Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA*

and

K.S. TRIVEDI

*Duke University, Durham, NC 27706, USA*

## Abstract

System availability is becoming an increasingly important factor in evaluating the behavior of commercial computer systems. This is due to the increased dependence of enterprises on continuously operating computer systems and to the emphasis on fault-tolerant designs. Thus, we expect availability modeling to be of increasing interest to computer system analysts and for performance models and availability models to be used to evaluate combined performance/availability (performability) measures. Since commercial computer systems are repairable, availability measures are of greater interest than reliability measures. Reliability measures are typically used to evaluate nonrepairable systems such as occur in military and aerospace applications. We will discuss system aspects which should be represented in an availability model; however, our main focus is a state of the art summary of analytical and numerical methods used to solve computer system availability models. We will consider both transient and steady-state availability measures and for transient measures, both expected values and distributions. We are developing a program package for system availability modeling and intend to incorporate the best solution methods.

## Keywords and phrases

Availability, probabilistic modeling, numerical methods, fault tolerance.

## 1. Introduction

Most of the modeling work in fault-tolerant computing has focused on models for ultrareliable systems with long mission time requirements such as aviation and space computers, wind-tunnel systems, ballistic missile defense computers, etc. The modeling of maintained or repairable systems with high availability requirements

© J.C. Baltzer A.G., Scientific Publishing Company

such as telephone switching systems, general-purpose data processing computers, database computers (e.g. banking, airline reservation, and insurance claims) and communication network computers has received relatively little attention.

In the former class of systems, no down time can be tolerated during the mission time. Therefore, redundancy is used to replace failed components *almost instantaneously* for a fixed level of service. Although repair can be performed on the failed components, usually repairmen are not handy during the mission. Even if repairmen were handy, a mission failure occurs either if redundancy is exhausted or if imperfect switching takes place. Reliability, i.e. the probability that the system remains operational over a given time period, is an appropriate measure for evaluating the effectiveness of this class of systems.

Systems in the second class are usually operated continuously and short down times during their operation can be tolerated. Therefore, redundancy is used to improve the performance under normal operation and to reduce the down time in the case of a failure. In some cases, hot standby systems are used just to reduce the down times. In this class of systems, both *preventive* and *corrective* maintenance can be performed to obtain the desired level of service. Availability, i.e. the fraction of time the system is operational, is a more appropriate measure for evaluating the effectiveness of this class of systems.

The choice of a dependability measure requires consideration about where the main cost or penalty of system failure is. If it is associated with the frequency of failures, a measure based on the time between successive failures is needed, and if it is associated with the fraction of time the system is down, an availability measure is needed. Although most of the work in performance modeling of computer systems has concentrated on the steady-state behavior of the system, in availability modeling the transient behavior of computer systems is also important because failures and repair occur very infrequently, and systems may not reach a steady state during the period of observation.

Recently, many vendors (e.g. DEC, HP, WANG, etc.) have announced computer systems with guaranteed levels of availability in the 0.95 to 1.0 range. They pay a penalty if this level of availability is not met over a finite time interval, for example, over a quarter, half a year, or a year. When vendors offer this guaranteed availability, they assess their risk of not meeting this guanranteed level and charge a premium in their maintenance agreement. The risk assessment involves evaluating the probability of not meeting the minimum requirements on the availability or, in turn, the distribution of availability over a finite interval.

In this paper we consider all three types of availability measures, namely steady-state availability, transient or interval availability, and distribution of availability over a finite interval. In sect. 2 we present system aspects which should be considered when constructing an availability model and define the various measures of availability. The next three sections consider analytical and numerical techniques

used in evaluating steady-state availability, transient or interval availability, and the distribution of availability, respectively. The focus is primarily on Markovian models. In sect. 6 we briefly describe combined measures of performance and availability, and refer the reader to relevant work done for evaluating these measures.

## 2. System aspects to be modeled

When evaluating the availability of a system, one has to be careful about the level of detail included in the model. The reason we are studying the availability of a system is to compare various redundancy techniques, to make subsequent design trade-offs, and to pinpoint the subsystems which are availability bottlenecks. We should also consider the accessibility of failure and repair data. Typically in a computing system, Field Replaceable Units (FRUs) are identified which are replaced if any component within the FRU fails. Therefore, failure/repair data on FRUs is easier to obtain. This suggests that we should model a system at a level where the redundancy occurs or at FRU level or at a subsystem level.

The smallest entity represented in the model is termed a *component.* We assume that components can be in one of two states: operational or failed. A system is considered operational, or available, if at least a minimum set of these components is operational. The failure and repair process of these components can be represented by a state graph with state space $\Psi$, where each state is a distinct combination of operational and failed states of each individual component. In the simplest case, the whole system can be considered a single component yielding a state graph with two states representing the system operational state and the system failure state, respectively. Components are not restricted to be hardware components. Software *in operation* can also be considered a component in the model. This is in contrast to software *in testing mode,* whose failure rate decreases as bugs are removed.

In availability models, many simplifying assumptions are made about the real system behavior which can reduce the model's accuracy. In order to improve this situation, we need to formulate and solve models that make less restrictive assumptions and to attempt to validate those that are made. There are three aspects of availability models which must be considered in creating believable models of real systems.

### 2.1. PHYSICAL INTERCONNECTION OF COMPONENTS

A system is considered to be an interconnected collection of components, each of which can be either operational or failed. Some availability models assume full connectivity (every component is connected to every other), which is not the case in most real computer systems. For example, consider a hypothetical fault-tolerant database system shown in fig. 1. Processors have their own local disks (databases) and share another disk where data logs are kept. Transactions are processed on both

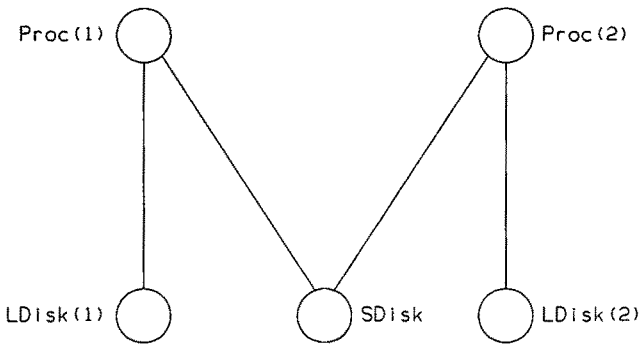


Fig. 1. Database example.

processor local disk pairs. The system is considered operational if at least one processor local disk pair is operational and the shared disk is operational. Such conditions for the system to be operational are called *assertions.*

In any given state of the system, we determine whether the assertion is satisfied or not as follows. We remove all failed components and their respective interconnections from the physical interconnection graph. This may create one or more interconnected sets of components called *partitions.* If any of these partitions satisfy the assertion, the system is considered operational. For the example of fig. 1, an assertion could be (1 of Proc) AND (1 of LDisk) AND (1 of SDisk). If Proc(1) and LDisk(2) have failed and the rest of the system is operational, the system still has 1 processor, 1 local disk, and the shared disk, and it may appear that the assertion is satisfied. However, the system is not operational because the assertion is not satisfied in any one of the partitions.

### 2.2. FAILURE BEHAVIOR

In most availability models, the components are assumed to fail independently. Common cause failures (e.g. power failure brings many components down, CPU failure brings operating system down), dormant failures (e.g. a standby component fails at a different rate than the operational component), and blocked failures (e.g. during system repair, certain components can not fail) are usually ignored. Moreover, the failure process is not modeled in detail. For example, a hardware fault could be transient, intermittent, or permanent. This fault is not detected until the hardware component is exercised and it causes an error which may then lead to component or system failure. An intermittent fault may exist for a long time before it causes an error. However, it may be difficult or impossible to estimate parameters at the fault level. We recommend modeling at the error level if the system implements some error recovery techniques (see subsect. 2.3) and otherwise at the failure level. Another important aspect of failures is their time-dependent behavior. For example, as design

faults or software bugs are detected, the frequency of failure decreases. One has to make a judicious choice of the failure behavior representation in the model based on the available data, the questions being addressed with the model and the desired model accuracy.

### 2.3. REPAIR BEHAVIOR

There are many aspects of repair which should be considered for inclusion in an availability model. Some systems have automatic recovery techniques. A very common technique to recover from transient error is to do checkpointing and retry. Since a large percentage of errors are known to be due to transient faults [31], many levels of retries (e.g. instruction retry, process retry, task retry, etc.) are implemented in most computer systems. If the automatic retry is not successful for any reason (e.g. permanent fault, error propagates beyond the checkpoint, number of retries attempted reaches a prespecified threshold, etc.), a manual repair takes place. There are various levels of manual repair. The operator may simply try to restart the system, which is often successful. The operator may isolate the failed component and try to restart the system without it, that is, the system is made available quickly with off-line repairs on the failed component. The last resort is to call repairmen (if not in-house) to perform reconfiguration and repairs. Other aspects of repair which could be included in the model are preventive maintenance and deferred and non-deferred repair. Repair dependencies could be included (e.g. a service processor must be repaired before the main processor can be repaired). Sometimes there is more than one repairman performing the job. There may be queueing for repairmen, in which case the queueing discipline should be represented.

Once again, we must make a judicious choice of including certain aspects of the repair behavior in the system model and approximating or ignoring the others. This choice must depend upon the available data, the question being addressed, and the accuracy desired.

In most availability models it is assumed that the times to failure and times to repair are exponentially distributed so that the model is Markovian. From a practitioner's point of view, these assumptions put severe limitations on the believability of the model results. Some attempts have been made to include non-exponential failure and repair times in availability models. These approaches include using phase-type distributions, using Markov renewal processes, and using Monte Carlo simulation. Although simulation is the most flexible approach, it has problems with ultra-high availability systems because of the desired accuracy of the solution. Some events are so rare that to obtain reasonable confidence intervals, an unreasonably large number of replications may need to be performed. Variance reduction techniques for ultra-high availability systems have been investigated to a limited extent [19].

### 2.4. AVAILABILITY MEASURES

In the remainder of this paper (with the brief exception of subsect. 3.2.1), we assume that the failure and repair time distributions are either exponential or phase-type. Therefore, the state of the system evolves in time as a finite state time-homogeneous continuous time Markov chain $\{\underset{\sim}{X}(t): t \geqslant 0\}$. We partition the total set of states ($\Psi$) into an operational set ($\Psi_o$) and a failed set ($\Psi_f$) based on interconnections and assertions. Define a random variable $\theta(t)$ such that

$$\theta(t) = \begin{cases} 1 & \underset{\sim}{X}(t) \in \Psi_o \\ 0 & \underset{\sim}{X}(t) \in \Psi_f \end{cases}, \tag{1}$$

and define $A(t)$ such that

$$A(t) = \frac{1}{t} \int_0^t \theta(s)\, ds . \tag{2}$$

The availability measures of significance are steady-state availability,

$$A = \lim_{t \to \infty} A(t) = \lim_{t \to \infty} E[A(t)] , \tag{3}$$

interval availability,

$$I(t) = E[A(t)] , \tag{4}$$

and the distribution of availability,

$$F(t, x) = \Pr[A(t) \leqslant x] . \tag{5}$$

In the following sections we survey the analytical and numerical techniques used to evaluate the above three measures of availability, and comment on our experience with these techniques in the context of developing a system availability modeling package called SAVE (System AVailability Estimator).

## 3. Steady-state availability

Steady-state availability is probably the most commonly used availability measure. If there is no interaction among components with respect to both their failure and repair behavior, i.e. if the sequences of times to failure and repair for

different components are mutually independent, then the steady-state system availability can be obtained directly from the steady-state component availabilities using combinatorial methods. We briefly discuss this case in subsect. 3.2.1. If on the other hand there is interaction among components, the combinatorial methods no longer suffice, as discussed next.

### 3.1. ARBITRARY INTERACTION AMONG COMPONENTS

In general, components interact due to contention for limited repair facilities, repair dependencies, common-cause and blocked failues, as discussed in sect. 2. As mentioned in subsect. 2.4, we consider only finite state time-homogeneous Markov chain models. The steady-state probabilities $\underset{\sim}{\pi}$ are obtained using

$$\underset{\sim}{\pi}\underset{\approx}{Q} = 0, \quad \sum_{\underset{\sim}{x} \in \Psi} \pi_{\underset{\sim}{x}} = 1, \tag{6}$$

where $\underset{\approx}{Q}$ is the transition rate matrix. The steady-state availability is evaluated based on interconnection and assertions as

$$A = \sum_{\underset{\sim}{x} \in \Psi_0} \pi_{\underset{\sim}{x}} . \tag{7}$$

Two characteristics of availability models that must be considered in computing $A$ using eqs. (6) and (7) are (i) the exponential increase in the number of states (size of matrix $\underset{\approx}{Q}$) with the number of components in the system, as well as with details incorporated about the failure and the repair behavior of the system, and (ii) the orders of magnitude differences in the transition rates (repair rates $\gg$ failure rates).

Since $\underset{\approx}{Q}$ is sparse, the state space size problem can be alleviated to some extent by using sparse matrix storage techniques. Iterative methods are particularly suitable for solving eq. (6). Sparse storage techniques for such methods are easy to implement, since the iterative methods do not alter the matrix. Direct methods, on the other hand, need much more sophisticated sparse storage schemes, for allowance must be made for fill-in (zero elements which become non-zero as a result of operations upon the matrix) as well as for the elimination of non-zero elements. In some cases, the fill-in can become very excessive, which is difficult to predict a priori. In addition, with iterative methods advantage can be taken of good initial approximations, especially when a series of related models are being solved. Also, the iterative procedure can be halted once a prespecified tolerance criterion has been satisfied (e.g. a user may need 3 decimal place accuracy), whereas direct methods, by definition, perform a fixed amount of computations and yield the best accuracy they can. Finally, iterative methods do not suffer from problems of stability, for successive iterates always refer

to the coefficient matrix *which is not altered.* For these reasons, we prefer iterative methods for solving large availability models. In developing the SAVE availability modeling package, we have experimented with a few numerical methods, including the Gauss–Seidel (with successive over relaxation), Power, Lanczos, and Lopsided iteration methods [34,35]. Selection of a particular method will be based on accuracy, convergence, and execution time. Since we have not yet experimented extensively with these methods, we report only our preliminary experience.

The Gauss–Seidel method can be programmed in two ways. Assuming that there is a total of $N$ states, the first method involves using the $N \times N$ singular $\underset{\sim}{Q}$ matrix in Gauss–Seidel iterations and renormalizing $\underset{\sim}{\pi}$ so that the probabilities sum to 1. Renormalization may be done either after the method converges, after each iteration, or at iterations where there is a possibility of an overflow or underflow. In the second method, we assume $\pi_N = 1$ and rewrite eq. (6) as $\underset{\sim}{\pi}^{\star}\underset{\sim}{Q}^{\star} = \underset{\sim}{b}^{\star}$, where the $^{\star}$ implies that the dimensions are $N-1$. Notice that $\underset{\sim}{Q}^{\star}$ is no longer singular and $\underset{\sim}{b}^{\star}$ is not a 0 vector. After the Gauss–Seidel iteration converges, we renormalize $\underset{\sim}{\pi}$ so that the probabilities sum to 1. The first method converges much faster than the second method; however, the first method is not guaranteed to converge, while the second is [17,12,37]. The convergence rate of the Gauss–Seidel method can be improved by using the successive over relaxation method and selecting the optimum relaxation parameter dynamically [35]. The Power method consists of first multiplying eq. (6) by a suitably chosen positive $\Delta t$ and then obtaining a fixed point iteration $\underset{\sim}{\pi} = \underset{\sim}{\pi}\underset{\sim}{P}$, where $\underset{\sim}{P} = \underset{\sim}{I} + \Delta t \underset{\sim}{Q}$. Now if $\Delta t \leqslant 1/q$, where $q = \max(-q_{i,i})$, then $\underset{\sim}{P}$ will be a stochastic matrix. If the original continuous time Markov chain is irreducible, then the resulting discrete time Markov chain (described by $\underset{\sim}{P}$) is also irreducible. Moreover, if $\Delta t < 1/q$, the discrete time Markov chain is also aperiodic. Therefore, the Power method using matrix $\underset{\sim}{P}$ will always converge. However, the number of iterations could be extremely large, as the second largest eigenvalue of $\underset{\sim}{P}$ can be very close to 1 for many availability modeling problems. The comparisons for the number of iterations taken by the Gauss–Seidel and the Power methods appear in [12]. The Lopsided method can increase the convergence rate of the Power method by using $m$ trial vectors. The convergence rate of the Lopsided method depends upon the $m+1$'th largest eigenvalue of $\underset{\sim}{P}$. The problem with the Lopsided method is that a good value of $m$ is not known a priori. The value of $m$ can become large, creating storage problems. Moreover, complete eigensolution of an $(m+1) \times (m+1)$ "interaction matrix" is needed, which requires complex arithmetic. There are other problems related to "defective" interaction matrices and "rotating eigenvectors" which complicate the use of the Lopsided method. However, solutions to these problems do exist [33]. The Lanczos method appears to be promising in some cases, details of which appear in [35]. We have not investigated other iterative methods which involve the use of exact aggregation and disaggregation [4].

We have been most successful by taking advantage of the special structure of availability models. Typically in availability models, states with a total of $i$ failed

components have higher probabilities of occupancies than states with $i + 1$ failed components, $i \in \{0, 1, \ldots\}$. We solve for the steady-state probabilities in a system with a maximum of $i + 1$ failed components by using the starting vector obtained from solving for the steady-state probabilities in a system with a maximum of $i$ failed components. The probabilities for states with exactly $i + 1$ failed components are assumed to be 0 in the starting vector. For $i = 1$, the steady-state probabilities can be evaluated in a closed form very easily [10]. The $\underset{\sim}{Q}$ matrix needs to be evaluated and stored once, except that the diagonal elements need to be re-evaluated when we go from $i$ to $i + 1$. For each $i \in \{2, 3, \ldots\}$, we begin with the first Gauss–Seidel method and if it starts diverging, we revert to one of the methods which is guaranteed to converge. Our stopping rule uses Cauchy's criterion which compares the maximum of the absolute differences in the elements of the solution vectors obtained on two successive iterations. This criterion may stop the iteration prematurely, and therefore we plan to investigate other stopping rules, discussed in [17,12,35]. Another advantage of breaking the problem as above is that we may decide not to consider states with more than $c$ failed components, based on the convergence in availability estimates obtained for $i \in \{1, 2, \ldots, c\}$, thus reducing the overall computation time and storage requirements.

### 3.2. REDUCING THE COMPUTATIONAL EFFORT IN SPECIAL CASES

The above approach for availability evaluation is very general, though restrictive in practice because of the exponential increase in the number of states with the number of components in the system as well as with the details incorporated about the failure and the repair behavior of the system. For restricted interaction among components, some specialized techniques exist to reduce storage and computation time requirements.

#### *3.2.1. No interaction among components*

In this case, we do not allow any failure or repair dependencies among components. However, the time to failure and time to repair distributions for each individual component can be arbitrary as long as either the repair instants or the failure instants form a renewal process (the time to failure and the time to repair of a component can be dependent). Then the steady-state availability of a component $i$ is given by [30]

$$A_i = \frac{\text{Mean time to failure}}{\text{Mean time to failure} + \text{Mean time to repair}} \tag{8}$$

and the availability of the system is given by a combination of the component availabilities given above. Physical interconnection of the components can be arbitrary with

arbitrary assertions placed on the operational conditions of the system. For the example of fig. 1, the steady-state availability is given by

$$A = A_{SD}\,[1-(1-A_{Proc(1)}A_{LD(1)})\,(1-A_{Proc(2)}A_{LD(2)})]\,. \tag{9}$$

A computer package called ADVISER developed at CMU [20] takes arbitrary physical interconnection among components and arbitrary assertions to yield a symbolic expression for the steady-state availability of the system. Notice that no failure and/or repair dependencies have been taken into account.

### 3.2.2. *Product form queueing networks*

An availability model can be represented by a queueing network with two service centers, one (typically an infinite server) corresponding to the failure process, and the other (typically a multiserver) corresponding to the repair process. A separate closed chain with population one is created for each component. However, if a group of components has probabilistically identical failure/repair behavior, then that group can be represented by a single closed chain with population equal to the number of components in the group. In the example of fig. 1, if the processors behave identically and the local disks behave identically, we can create a chain for processors, a chain for local disks, and a chain for the shared disk with populations of 2, 2 and 1, respectively

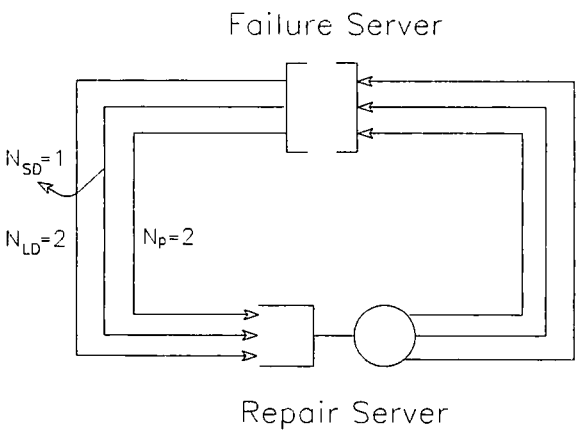


Fig. 2. Queueing network for the database example.

(fig. 2). Notice that the queueing disciplines should neither distinguish between Proc(1) and Proc(2) nor between LDisk(1) and LDisk(2). The steady-state probabilities for the Markov chain corresponding to this queueing network can be obtained using one of the methods given in subsect. 3.1. However, the state (population vector)

of the network does not distinguish among the components in a chain. Therefore, if the components are not fully connected, it may not be possible to determine from the state of the network whether the system is operational or failed. For example, if the components are not fully connected as in fig. 1, the state with 1 processor and 1 local disk failed will be an ambiguous state from the system availability point of view. However, as discussed in subsect. 3.2.3, it is possible to compute the steady-state availability from the steady-state probabilities even if the components are not fully connected.

If the queueing disciplines and the service time distributions at the service centers in the availability model belong to the product form class [1], the steady-state probabilities can be obtained using one of many algorithms for product form queueing networks [25]. A few useful examples of such service centers in availability models are:

(1) Infinite server,
(2) Last-Come First-Served Preemptive Resume (LCFSPR),
(3) Random Order Service Preemptive Resume (ROSPR) [10],
(4) First-Come First-Served (FCFS).

The service time distributions at the first three service centers can be phase-type, which can be different for different chains. However, at an FCFS service center they must be exponential with identical mean for different chains. The product form is preserved for more general availability models incorporating common-cause failures, blocked failures, dormant failures, automatic recovery mechanisms, etc., details of which appear in [10].

The advantage of preserving the product form is that the availability can be evaluated for much larger models with tremendous savings in storage and execution times (the $Q$ matrix does not need to be stored). We are investigating the stability of various product form algorithms [25] for orders of magnitude differences in service rate parameters and for state-dependent arrival and service rates.

If the availability model does not satisfy the product form conditions or the conditions of subsect. 3.2.1, then the following approach is useful [9].

### 3.2.3. *Lumping and unlumping states*

One technique to reduce the computational effort is to *lump* states into disjoint groups $\Phi_1, \Phi_2, \ldots, \Phi_n$. A necessary and sufficient condition for the lumped process to form a Markov chain is that for any two groups $\Phi_I$ and $\Phi_J$, the transition rate from a state $i$ in $\Phi_I$ to $\Phi_J$ (i.e. $\Sigma_{j \in J} q_{ij}$) is the same for each $i$ in $\Phi_I$ [18]. These common values form the transition rate matrix for the lumped chain. In order to compute $A$ from the lumped chain steady-state probabilities, a lumped group of states must contain either only operational states or only failed states. Therefore, when $A$ is to be computed, the physical interconnection of components and the

assertions also play a role in lumping. In the example of fig. 1, suppose that the failure and the repair behaviors for both processors are identical. If only one processor and LDisk(1) have failed, the system is operational if the failed processor is Proc(1), while it is failed if the failed processor is Proc(2). Therefore, it is not possible to lump states corresponding to the same number of failed processors and compute $A$ as discussed above. However, if stronger conditions than those required for lumping hold, it is possible to lump states into groups regardless of the physical interconnection and assertions while solving for the steady-state probabilties, and unlump groups into states (if needed) while evaluating the availability of the system based on the physical interconnection and assertions. To make the task of unlumping easier, we restrict the lumping as follows.

In addition to the condition for lumping given earlier in this section, we require that for every pair of groups $\Phi_I$ and $\Phi_J$, the transition rate from $\Phi_I$ to a state $j$ in $\Phi_J$ (i.e. $\Sigma_{i \in I} q_{ij}$) is the same for every state $j$ in $\Phi_J$, where the condition must also hold for $I = J$ [9]. Now the unlumping is easy because the above conditions are sufficient for the set of states belonging to a group to have equal probabilities. These stronger conditions for lumping are satisfied when there are groups of components such that the components in a group have identical failure behavior and identical repair behavior. Such lumping was done implicitly using the queueing network representation in subsect. 3.2.2 by representing a group of identical components by a single closed chain.

As an example, again consider the system in fig. 1 and assume that the processors are identical (same failure behavior and same repair behavior) and that the local disks are identical. Moreover, assume any finite number of repairmen and a queueing discipline which gives preemptive priority to the shared disk over the processors and to the processors over the local disks. Due to the priority queueing discipline, the solution is not the product form. Then the states having the same number of operational processors and the same number of operational local disks can be lumped. There are 18 groups and the lumped chain can be solved using one of the methods discussed in subsect. 3.1. The group with one processor, one local disk, and one shared disk operational is the only group which has some states belonging to $\Psi_o$ and the remaining to $\Psi_f$. The rest of the 17 groups have all states belonging to either $\Psi_o$ or $\Psi_f$. Therefore, only one group needs to be unlumped into its constituent states, Since

Pr[1 Proc .AND. 1 LD .AND. SD]

= Pr[Proc(1) .AND. LD(1) .AND. SD] system up

+ Pr[Proc(1) .AND. LD(2) .AND. SD] system down

+ Pr[Proc(2) .AND. LD(1) .AND. SD] system down

+ Pr[Proc(2) .AND. LD(2) .AND. SD] system up

and all states in the group have equal probabilities, the availability of the system is easily evaluated. The lumping-unlumping method, regardless of whether the solution is obtained via product form algorithms or by solving the Markov chain, reduces the number of states by orders of magnitude. Therefore, it is currently being used in the SAVE package.

### 3.3. SENSITIVITY EVALUATION

Sensitivity evaluation with respect to failure and repair parameters of the system is also very important. In eq. (6), we wish to evaluate the sensitivity of the vector $\underset{\sim}{\pi}$ with respect to one of the failure or repair rate parameters, say $\lambda$, which could be a part of many elements in the matrix $\underset{\approx}{Q}$. By differentiating both sides of eq. (6) with respect to $\lambda$, we get

$$\underset{\sim}{\pi} \frac{\mathrm{d}\underset{\approx}{Q}}{\mathrm{d}\lambda} + \frac{\mathrm{d}\underset{\sim}{\pi}}{\mathrm{d}\lambda} \underset{\approx}{Q} = 0, \sum \frac{\mathrm{d}\underset{\sim}{\pi}}{\mathrm{d}\lambda} = 0. \tag{10}$$

The above equation can be written as

$$\frac{\mathrm{d}\underset{\sim}{\pi}}{\mathrm{d}\lambda} \underset{\approx}{Q} = b,$$

where

$$b = -\underset{\sim}{\pi} \frac{\mathrm{d}\underset{\approx}{Q}}{\mathrm{d}\lambda}$$

can be evaluated after the steady-state probabilities have been computed. Therefore, we need to solve the same matrix equation as in eq. (6) except that the right-hand side is non-zero. The Gauss–Seidel methods can be used to solve eq. (10); however, the normalization is different. A Power-like iteration can also be created to solve eq. (10). In both cases, the convergence properties remain the same as discussed in subsect. 3.1 [35]. In general, the Lanczos or the Lopsided methods may not be used to solve a general set of simultaneous linear equations. Now, $\mathrm{d}A/\mathrm{d}\lambda$ can be obtained using eq. (7).

## 4. Interval availability

In this section we consider interval availability for finite state time-homogeneous Markov chain models. Following the standard approach, the dynamic behavior of the model can be characterized by a linear system of ordinary differential equations with constant coefficients [30]

$$\underset{\sim}{\pi}'(t) = \underset{\sim}{\pi}(t)\,\underset{\approx}{Q}, \tag{11}$$

where $\underset{\sim}{\pi}(t)$ is the state probability vector and $\underset{\approx}{Q}$ is the $N \times N$ transition rate matrix. Interval availability is then obtained from

$$I(t) = E[A(t)] = \sum_{\underset{\sim}{x} \in \Psi_0} \frac{1}{t} \int_0^t \pi_{\underset{\sim}{x}}(u)\,\mathrm{d}u. \tag{12}$$

It is convenient in what follows to let

$$\underset{\sim}{I}(t) = \frac{1}{t} \int_0^t \underset{\sim}{\pi}(u)\,\mathrm{d}u \tag{13}$$

so that

$$I(t) = \sum_{\underset{\sim}{x} \in \Psi_0} I_{\underset{\sim}{x}}(t). \tag{14}$$

There are many ways to solve eq. (11), some of which have been used in existing availability and reliability modeling packages. One form of the solution to eq. (11) is

$$\underset{\sim}{\pi}(t) = \underset{\sim}{\pi}(0)\,\mathrm{e}^{\underset{\approx}{Q}t}, \tag{15}$$

which can be evaluated using the power series to yield

$$\underset{\sim}{\pi}(t) = \sum_{k=0}^{\infty} (\underset{\sim}{\pi}(0)\,\underset{\approx}{Q}^k)\,\frac{t^k}{k!} \tag{16}$$

and hence

$$\underset{\sim}{I}(t) = \sum_{k=0}^{\infty} (\underset{\sim}{\pi}(0)\,\underset{\approx}{Q}^k)\,\frac{t^k}{(k+1)!}. \tag{17}$$

However, since $\underset{\approx}{Q}$ has both negative and positive elements, the above sum may not be well behaved in the numerical sense.

Another form of the solution to eq. (11) is expressed in terms of the eigenvalues and the corresponding eigenvectors of $\underset{\sim}{Q}$. Assuming distinct eigenvalues, the solution can be written as

$$\underset{\sim}{\pi}(t) = \sum_{i=1}^{N} e^{\lambda_i t} a_i \underset{\sim}{v}_i , \tag{18}$$

where $\lambda_i$ is an eigenvalue, $\underset{\sim}{v}_i$ is the corresponding eigenvector of $\underset{\sim}{Q}$, and $a_i$ are determined from the initial conditions

$$\underset{\sim}{\pi}(0) = \sum_{i=1}^{N} a_i \underset{\sim}{v}_i .$$

Now, $\underset{\sim}{\pi}(t)$ can be symbolically integrated term by term to obtain an expression for $\underset{\sim}{I}(t)$. A slightly more complicated form results if eigenvalues are repeated [3]. An alternative formulation used in the ARIES modeling package [28] avoids the computation of eigenvectors. However, the classical approach is $O(N^4)$, while the ARIES approach is $O(N^5)$ [6]. More importantly, however, there are several other problems with the eigenvalue approach [3]. First, even though $\underset{\sim}{Q}$ has only real entries, some of its eigenvalues can be complex. Furthermore, realistic problems can result in repeated eigenvalues and numerical problems arise in the case of numerically close eigenvalues. Added to these problems is the difficulty of finding eigenvalues of a large matrix.

The solution to eq. (11) can also be obtained using Laplace transforms, as follows:

$$\underset{\sim}{\pi}(t) = \underset{\sim}{\pi}(0) L^{-1} [(s\underset{\sim}{I} - \underset{\sim}{Q})^{-1}] . \tag{19}$$

This is the approach used in the SURF modeling package [24] and in general it requires numerical inversion of the Laplace transform, although for some models inversion via partial fractions may be possible. Since we are interested in $\underset{\sim}{I}(t)$ instead of $\underset{\sim}{\pi}(t)$, we should invert $(s\underset{\sim}{I} - \underset{\sim}{Q})^{-1}/s$, i.e.

$$\underset{\sim}{I}(t) = \frac{1}{t} \underset{\sim}{\pi}(0) L^{-1} [(s\underset{\sim}{I} - \underset{\sim}{Q})^{-1}/s] . \tag{20}$$

An approach that appears to be more promising than any of the above methods for computing $\underset{\sim}{\pi}(t)$ and $\underset{\sim}{I}(t)$ is the randomization (or uniformization) method [11,13,26]. The method is based on an equation which has a nice probabilistic interpretation [16,30] which we omit here. The resulting equation can also be derived

quite simply as follows. First make the change of variables $\underset{\sim}{\alpha}(t) = e^{qt}\underset{\sim}{\pi}(t)$, where $q \geqslant \max(-q_{ii})$, in eq. (11), yielding

$$\underset{\sim}{\alpha}'(t) = \underset{\sim}{\alpha}(t)\,\underset{\approx}{Q}^{\star}\,, \tag{21}$$

where $\underset{\approx}{Q}^{\star} = \underset{\approx}{I} + \underset{\approx}{Q}/q$. Next, apply the power series expansion method to the above equation [as it was applied to eq. (11) to yield eq. (16)] to solve for $\underset{\sim}{\alpha}(t)$, yielding

$$\underset{\sim}{\pi}(t) = e^{-qt}\,\underset{\sim}{\alpha}(t) = \sum_{k=0}^{\infty} \underset{\sim}{\pi}(0)\,(\underset{\approx}{Q}^{\star})^{k}\,\frac{e^{-qt}(qt)^{k}}{k!}\,. \tag{22}$$

(The above equation can also be dervived probabilistically as in [13].) Numerical problems are minimized, since the matrix $\underset{\approx}{Q}^{\star}$ has only nonnegative entries. Finally, by symbolically integrating eq. (22) we get

$$\underset{\sim}{I}(t) = \frac{1}{qt}\sum_{k=0}^{\infty} \underset{\sim}{\pi}(0)\,(\underset{\approx}{Q}^{\star})^{k} \sum_{n=k+1}^{\infty} \frac{e^{-qt}(qt)^{n}}{n!}\,. \tag{23}$$

All terms are nonnegative and the equation can be used to efficiently compute $\underset{\sim}{I}(t)$. It is easy to show that the $L_1$ norm of the error obtained by approximating $\underset{\sim}{I}(t)$ by the sum of the first $K$ terms in eq. (23) is upper bounded by

$$\sum_{n=K+1}^{\infty} \frac{e^{-qt}(qt)^{n}}{n!}\,. \tag{24}$$

It follows from eq. (14) that this value also upper bounds the resulting error in the interval availability.

Yet another approach to solving eq. (11) is to use numerical integration methods. In a test of numerical methods for initial value problems, Hull et al. [14] compared variable-order predictor-corrector methods, Runge–Kutta methods, and extrapolation methods on a wide selection of test problems, and concluded that when function evaluations are relatively expensive (as is the case for large and sparse transition matrices), variable-order methods are best, such as that of Krogh and Gear. Also, Lambert [23] suggests that Gear's method represents the most highly developed method for the treatment of stiff systems such as the ones we encounter due to the orders of magnitude differences in transition rates.

Once again, we are interested in $\underset{\sim}{I}(t)$ rather than $\underset{\sim}{\pi}(t)$. Symbolically integrating eq. (11) yields

$$\underset{\sim}{\pi}(t) = \left( \int_0^t \underset{\sim}{\pi}(u)\, du \right) \underset{\approx}{Q} + \underset{\sim}{\pi}(0), \tag{25}$$

or in terms of $\underset{\sim}{\beta}(t) = t\underset{\sim}{I}(t)$, we have

$$\underset{\sim}{\beta}'(t) = \underset{\sim}{\beta}(t)\underset{\approx}{Q} + \underset{\sim}{\pi}(0), \tag{26}$$

which can be integrated numerically to obtain $\underset{\sim}{\beta}(t)$ and hence $\underset{\sim}{I}(t)$.

We believe that either randomization, i.e. use of eq. (23), or numerical integration of eq. (26) using Gear-like methods are the preferred approaches for solving large availability models for interval availability. We are currently experimenting with these two methods for the SAVE package.

In order to derive the sensitivity of the vector $\underset{\sim}{I}(t)$ with respect to one of the failure or repair rate parameters, say $\lambda$, we proceed by differentiating both sides of eq. (26) with respect to $\lambda$, which yields

$$\frac{\partial \underset{\sim}{\beta}'(t)}{\partial \lambda} = \frac{\partial \underset{\sim}{\beta}(t)}{\partial \lambda} \underset{\approx}{Q} + \underset{\sim}{\beta}(t) \frac{\partial \underset{\approx}{Q}}{\partial \lambda}, \tag{27}$$

where $\underset{\sim}{\pi}(0)$ does not depend on $\lambda$ so that $\partial \underset{\sim}{\pi}(0)/\partial\lambda = \underset{\sim}{0}$. Reversing the order of differentiation with respect to $t$ and $\lambda$, we get

$$\underset{\sim}{S}'(t) = \underset{\sim}{S}(t)\underset{\approx}{Q} + \underset{\sim}{\beta}(t) \frac{\partial \underset{\approx}{Q}}{\partial \lambda}, \tag{28}$$

where $\underset{\sim}{S}(t) = \partial \underset{\sim}{\beta}(t)/\partial\lambda$ is the sensitvity vector for parameter $\lambda$. We can solve for $\underset{\sim}{\beta}(t)$ and $\underset{\sim}{S}(t)$ simultaneously (and hence $\underset{\sim}{I}(t)$ and $\partial \underset{\sim}{I}(t)/\partial\lambda$) with one large set of simultaneous differential equations, namely

$$(\underset{\sim}{\beta}'(t), \underset{\sim}{S}'(t)) = (\underset{\sim}{\beta}(t), \underset{\sim}{S}(t)) \begin{bmatrix} \underset{\approx}{Q} & \dfrac{\partial \underset{\approx}{Q}}{\partial \lambda} \\ \underset{\sim}{0} & \underset{\approx}{Q} \end{bmatrix} + (\underset{\sim}{\pi}(0), \underset{\sim}{0}). \tag{29}$$

This approach can easily be extended to solve simultaneously for $\underset{\sim}{\beta}(t)$ and multiple sensitivity vectors for different model parameters. The most promising method to solve eq. (29) appears to be numerical integration.

## 5. Distribution of availability

In this section we briefly consider the distribution of availability over a finite interval for finite state time-homogeneous Markov process models. In general, it is quite difficult to evaluate the above distribution in a closed form. While an expression for the Laplace transform of $F(t, x)$ with respect to variable $x$ has been obtained for multiple component models, we are not aware of any technique which can be used to invert this transform to yield closed form results for repairable systems. The Laplace transform could be inverted numerically to get $F(t, x)$. This approach is under investigation [22]. For very large $t$, $\sqrt{t}A(t)$ converges to a normal distribution (e.g. see [32]) whose mean and variance can be computed. However, there are two problems with using the normal approximation. First, most intervals of interest are short intervals where the normal distribution can give severe errors, and second, for high availability systems we are interested in the tail of the distribution where the normal approximation tends to fail [8]. For a two-state model with failure and repair rates $\lambda$ and $\mu$, respectively, the following equation was obtained in [29]

$$F(t,x) = 1 - \exp(-\lambda x)\,[1 + \sqrt{\lambda\mu x}\int_0^{t-x} \exp(-\mu y)\, y^{-1/2} I_1(2\sqrt{\lambda\mu y x})\,\mathrm{d}y], \quad (30)$$

where $I_1(a)$ is the Bessel function of order 1 defined by

$$I_1(a) = \sum_{j=0}^{\infty} \frac{(a/2)^{2j+1}}{j!(j+1)!}, \quad (31)$$

and the initial state is the operational state. Although $F(t, x)$ could be evaluated by numerically integrating the above equation, it has not been investigated. Moreover, it is not clear how to extend the above analysis to multiple-component availability models. Uniformization can also be used to evaluate $F(t, x)$ for two-state models, as shown in [30].

The most promising approach for multicomponent models that has been described in the literature is the numerical approach presented in [8]. The approach is based on evaluating the joint probability that the cumulative operational time during an interval of length $t$ is $x$ and that the system is in a particular operational or failure state at time $t$. Equations are obtained relating these joint probabilities for arguments $t$ and $x$ to those for arguments $t - \Delta$ and $x$, and those for arguments $t - \Delta$ and $x - \Delta$, where $\Delta$ is chosen small enough so that the probability of more than one event in the Markov process in time $\Delta$ is negligible. These equations allow recursive computation of probability density functions of system availability. Details are given in [8].

As an example, consider a system which consists of three components in series. The failure rates of the three components are $\lambda_1 = 1/196$, $\lambda_2 = \lambda_3 = 1/392$, respectively, and the repair rates of the three components are $\mu_1 = 1, \mu_2 = 0.5, \mu_3 = 0.25$, respectively. We do not allow another component to fail if one component has already failed. The mean time between system failures is 98 hours and the mean time to repair the system is 2 hours. Hence, the steady-state availability of this system is 0.98. Using a $\Delta$ of 0.1, we show in fig. 3 the probability density function of system availability for

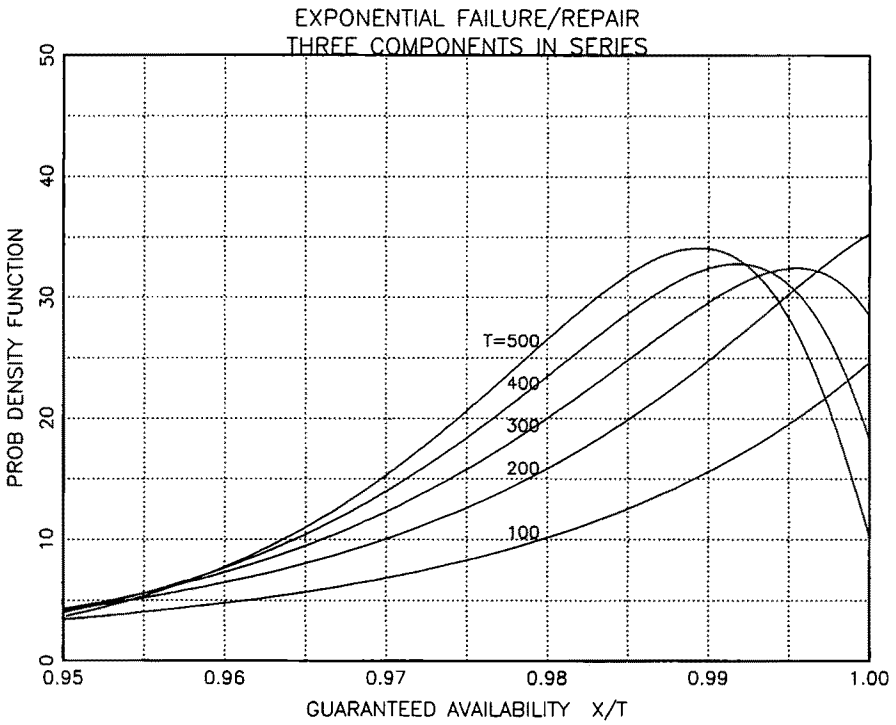


Fig. 3. Probability density function.

$t$ = 100, 200, 300, 400, and 500 hours. From fig. 3 we find that as the interval length $t$ increases, the distribution of system availability approaches very slowly to that of the steady-state availability, which is deterministic at $x/t = 0.98$. Also, we observe that for small values of $t$, the distribution of system availability is far from a normal distribution. There exists a finite probability of achieving a guaranteed availability of one, which is the probability of not failing during the entire interval $t$.

## 6. Combined performance and availability

Recently there has been a considerable interest in evaluating fault-tolerant computer systems that can operate in a degraded mode. Combined measures of performance and availability, called performability, have been proposed to evaluate such systems [2,27]. Performability measures are generalizations of availability measures and they also fall into three basic classes, namely steady-state performability, interval performability, and the distribution of performability over a finite interval.

We associate a performance level or a reward rate $r_i$ with being in state $i$ rather than simply a zero or one value, as has been the case in pure availability models. Define a random variable $\eta(t)$ such that

$$\eta(t) = r_i \quad \text{if} \quad \underset{\sim}{X}(t) = i \tag{32}$$

and define $Y(t)$ such that

$$Y(t) = \frac{1}{t}\int_0^t \eta(s)\,ds. \tag{33}$$

The steady-state performability is defined as

$$Y = \lim_{t \to \infty} Y(t) = \lim_{t \to \infty} E[Y(t)], \tag{34}$$

the interval performability as

$$C(t) = E[Y(t)], \tag{35}$$

and the distribution of performability as

$$G(t, x) = \Pr[Y(t) \leqslant x]. \tag{36}$$

The steady-state performability and the interval performability can be evaluated from $\underset{\sim}{\pi}$ and $\underset{\sim}{I}(t)$ directly. However, the distribution of performability poses considerable problems and has been evaluated only in special cases. For acyclic Markov chains (no recovery or repairs), $G(t, x)$ can be obtained analytically [7,5]. For cyclic Markov chains (models with repair), the Laplace transform of $G(t, x)$ with respect to $x$ has been obtained [15,21] and the moments of $Y(t)$ have been obtained [15]. The numerical inversion of the Laplace transform is under investigation [22].

## Acknowledgements

We would like to thank W.K. Grassman, V.G. Kulkarni and W.J. Stewart for their valuable technical suggestions.

## References

[1] F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios, Open, closed and mixed networks of queues with different classes of customers, Journal of ACM 22, 2(1975)248.

[2] M.D. Beaudry, Performance-related reliability measures for computing systems, IEEE Trans. on Computers 27, 6(1978)540.

[3] R.C. Buck and E.F. Buck, *Introduction to Differential Equations* (Houghton Mifflin Co., Boston, 1976).

[4] W.-L. Cao and W.J. Stewart, Iterative aggregation/disaggregation techniques for nearly decomposable Markov chains, JACM (1985), to appear.

[5] L. Donatiello and B. Iyer, Analysis of a composite performance reliability measure for fault tolerant systems, Revised IBM Research Report RC-10325, Yorktown Heights, JACM (1985), in review.

[6] R.M. Geist and K.S. Trivedi, Ultrahigh reliability prediction for fault-tolerant computer systems, IEEE Trans. on Computers 32, 12(1983).

[7] A. Goyal and A.N. Tantawi, Evaluation of performability for degradable computer systems, IBM Research Report RC-10529, Yorktown Heights (1984).

[8] A. Goyal and A.N. Tantawi, Numerical evaluation of guaranteed availability, Proc. FTCS-15 (1985).

[9] A. Goyal, On steady state availability modeling, in preparation.

[10] A. Goyal, Product-form in steady state availability models, in preparation.

[11] W.K. Grassman, Transient solutions in Markovian queueing systems, Computers in Operations Research 4(1977)47.

[12] D. Gross, L.C. Kioussis, D.R. Miller and R.M. Soland, Computational aspects of determining steady-state availability for Markovian multi-echelon repairable item inventory models, Tech. Rep. GWU/IMSE/Serial T-483/84, The George Washington University (1984).

[13] D. Gross and D.R. Miller, The randomization technique as a modeling tool and solution procedure for transient Markov processes, Oper. Res. 32, 3(1984)343.

[14] T.E. Hull, W.H. Enright, B.M. Fellen and A.E. Sedgewick, Comparing numerical methods for ordinary differential equations, University of Toronto Department of Computer Science Tech. Rep. No. 29 (1971).

[15] B. Iyer, L. Donatiello and P. Heidelberger, Performability evaluation of computing systems using reward models, IBM Research Report RC-10719, Yorktown Heights (1984).

[16] A. Jensen, Markoff chains as an aid in the study of Markoff processes, Skand. Aktuarietidskr. 36(1953)87.

[17] L. Kaufman, B. Gopinath and E.F. Wunderlich, Analysis of packet network congestion control using sparse matrix algorithms, IEEE Trans. on Communications 29, 4(1981)453.

[18] J.G. Kemeny and J.L. Snell, *Finite Markov Chains* (D. Van Nostrand Co., Princeton, NJ, 1967).

[19] L.C. Kioussis and D.G. Miller, An importance sampling scheme for simulating the degradation and failure of complex systems during finite missions, Proc. Winter Simulation Conf., Washington (1983) p. 631.

[20] V. Kini, Automatic synthesis of symbolic reliability functions for processor-memory-switch structures, Ph.D. Dissertation, Carnegie-Mellon University (1981).
[21] V.G. Kulkarni, V.F. Nicola and K.S. Trivedi, On modeling the performance and reliability of multi-mode computer systems, Int. Workshop on Modeling and Performance Evaluation of Parallel Systems (North-Holland, Amsterdam, 1984).
[22] V.G. Kulkarni, V.F. Nicola, K.S. Trivedi and R.M. Smith, Numerical evaluation of performability and job completion time in repairable fault-tolerant systems, Proc. FTCS-16 (1986).
[23] J.D. Lambert, *Computational Methods in Ordinary Differential Equations* (Wiley, New York, 1973).
[24] C. Landrault and J.C. Laprie, SURF – a program for modelling and reliability prediction for fault-tolerant comput. syst., in: *Information Technology*, ed. J. Moneta (North-Holland, Amsterdam, 1978).
[25] S.S. Lavenberg, editor, *Computer Performance Modeling Handbook* (Academic Press, New York, 1983).
[26] B. Melamed and M. Yadin, Randomization procedure in computation of cumulative-time distributions over discrete state Markov Processes, Oper Res. 32, 4(1984)926.
[27] J.F. Meyer, On evaluating the performability of degradable computing systems, IEEE Trans. on Computers 29, 8(1980)720.
[28] S.V. Makam, A. Avizienis and G. Grusas, UCLA ARIES 82 user's guide, Tech. Rep. No. CSD-820830 (1982).
[29] T. Nakagawa and A.L. Goel, A note on availability for a finite interval, IEEE Trans. on Reliability 22, 3(1973)271.
[30] S.M. Ross, *Stochastic Processes* (Wiley, New York, 1983).
[31] D.P. Seiwiorek and R.S. Swarz, *The Theory and Practice of Reliable System Design* (Digital Press, 1982).
[32] W.L. Smith, Renewal theory and its ramifications, J. Roy. Statist. Soc. Ser. B20(1958)243.
[33] W.J. Stewart, Markov analysis of operating system techniques, Ph.D. Thesis, Queen's University of Belfast (1974).
[34] W.J. Stewart, A comparison of numerical techniques in Markov modeling, CACM 21, 2 (1978).
[35] W.J. Stewart and A. Goyal, Matrix methods in large dependability models, IBM Research Report RC 11485 (1985).
[36] K.S. Trivedi, *Probability & Statistics with Reliability, Queueing, and Computer Science Applications* (Prentice-Hall, 1982).
[37] R.S. Varga, *Matrix Iterative Analysis* (Prentice-Hall, 1962).