

THE UNIVERSITY OF MICHIGAN  
COMPUTING RESEARCH LABORATORY

---

PERFORMABILITY EVALUATION OF COMPUTING  
SYSTEMS USING REWARD MODELS

D.G. Furchtgott and J.F. Meyer

CRL-TR-27-83

August 1983

Room 1079, East Engineering Building  
Ann Arbor, Michigan 48109  
USA  
Tel: (313) 763-8000

Enqm  
UMK  
1614

**PERFORMABILITY EVALUATION OF COMPUTING  
SYSTEMS USING REWARD MODELS**

**D. G. Furchtgott and J. F. Meyer**

**Computing Research Laboratory**

**The University of Michigan**

**Ann Arbor, MI 48109**

*ABSTRACT*--An evaluation algorithm is developed for a broad class of performability models wherein system performance is identified with "reward." More precisely, for a system  $S$  and a utilization period  $T$ , the performance variable of the model is the reward derived from using  $S$  during  $T$ . The state behavior of  $S$  is represented by a finite-state stochastic process (the base model); reward is determined by reward rates associated with the states of the base model. It is assumed that the corresponding reward model is a nonrecoverable process in the sense that a future state (reward rate) of the model cannot be greater than the present state. For this model class, we obtain a general method for determining the probability distribution function of the performance (reward) variable and, hence, the performability of the corresponding system. Moreover, this is done for bounded utilization periods, an assumption which demands a relatively complex solution. The result is an integral expression which can be solved either analytically or numerically. A program written for numerical solutions is discussed and an example illustrating the method is presented.

## I. INTRODUCTION

Much of the recent work in analytical evaluation of degradable fault-tolerant computing systems has concerned the evaluation of the total "benefit" derived from the computing system during some specified interval of time [1]-[19]. In these studies, a variety of concepts and terminology have been used to formalize benefit, including "work"[1], "capacity"[2], "reward"[7], and "performance"[4]. Other studies have concerned the evaluation of "cost" in conjunction with the evaluation of some form of benefit [8], [9], [11], [12], [15], [17]. Much of this work [3], [5]-[9], [11], [12], [16]-[18], [20] has considered systems in steady-state which are utilized over an unbounded period of time: repair of components is allowed (indeed, required), and the evaluated quantity is the expected rate at which benefit is derived under steady-state conditions. However, many important applications of degradable systems have bounded utilization periods. In such cases, a transient solution of the system benefit is usually required. Moreover, in many such applications, the user is interested in how benefit is distributed probabilistically (i.e., the probability distribution function of the benefit variable). This further complicates the evaluation process as compared, say, with the evaluation of expected benefit.

Conforming with terminology used in some of our earlier studies [10], [14], [21]-[23], we view system "performance" as a relatively general concept which includes various types of benefit as possible specializations. Formally, we define the *performance* of a system  $S$  over a specified time period  $T$  to be a random variable  $Y$  taking values in a set  $A$ . Elements of  $A$  are *accomplishment levels* representing how well  $S$  performs during  $T$ . Relative to a designated performance variable  $Y$ , *performability* is the probability measure  $p$  induced by  $Y$  where, for any measurable set  $B$  of accomplishment levels ( $B \subseteq A$ ),  $p(B)$  is the probability that  $S$  performs at a level  $B$ . Evaluation of performability thus requires solution of the probability distribution function of  $Y$ . This solution is based on an underlying stochastic process  $X$ , called the *base model* (of  $S$ ), which represents the dynamics of the system's structure, internal state, and environment during utilization.

Specific interpretations of performability thus depend on how values of  $Y$  (the accomplishment levels  $A$ ) are interpreted. In particular, if elements of  $A$  represent levels of benefit then performability describes the system's ability to benefit its user. In the discussion that follows, we consider this type of performability where benefit is equated with *reward* (see Howard [24] for example) derived from using the system during a specified period  $T$ . We are interested in the case where the time period  $T$  is bounded, although unbounded periods are considered during an intermediate step of the evaluation process. Also, by the definition of performability (see above), we seek to determine (either analytically or numerically) the probability distribution function of the reward variable  $Y$ .

Prior work relating to this problem has dealt primarily with evaluations of expected reward, i.e., the expected value of the reward variable  $Y$ . This background is nevertheless relevant and two approaches are of particular interest. The first is based on the concept of the "potential" of a semi-Markov process; see Cinlar [25] for an introductory discussion of potentials. Gay [20] (see also Gay and Ketelsen [3]) has applied potentials of Markov processes to modeling the performance of degradable systems. Most of this work deals with steady-state solutions of systems having repairable components. The second approach considers transient solutions of the expected reward and is based on semi-Markov reward models such as those discussed by Howard [24]. The formulation is in terms of a set of integrations and variables which must be solved in order to determine the expected reward. Because of the many inherent convolutions, the equations can be conveniently written in terms of Laplace transforms. However, the equations are generally difficult to solve (and if done in transform space, to invert), and so practical applications are limited. De Souza [7] has applied Howard's work to fault-tolerant computing systems via a unified reliability/cost model; however, these methods presume that the system is in steady state. Though some of the underlying characteristics (especially the nature of transition graphs) of these potential and reward models are allowed to be more general than those considered in this paper, these models are restricted to be semi-Markov and, more importantly, the evaluation results are

expected values rather than probability distribution functions.

Relatively little work has been done on obtaining transient solutions of the probability distribution function (PDF) of the reward variable. Notably, Cherkesov [26] presents an elegant solution for the probability of accumulating a minimum reward during a finite interval when the underlying process is semi-Markov. The length of the interval can itself be random. However, the solution is in terms of a system of equations of two-dimensional Laplace-Carson transforms (see p. 39 of [27] for a description of this transform). Solving and inverting these equations is intractable except in the simplest of cases. Instead, Cherkesov employs the transforms to obtain expected values. This work is mainly of theoretical interest.

In the presentation that follows, we obtain a method for determining the PDF of the reward variable, subject to certain conditions imposed on the base model and on the corresponding reward model. Specifically, we assume that the base model is a finite-state process and is “acyclic” in the sense that states are not revisited by the process. However, the process need not be Markov or even semi-Markov. We assume further that the corresponding reward model is a nonrecoverable process in the sense that a future state (reward rate) of the model cannot be greater than the present state. These conditions reflect properties that are typically exhibited by degradable, nonrepairable computing systems. For this model class, we are able to obtain the probability distribution function of the performance (reward) variable and, hence, the performability of the corresponding system. Moreover, this is done for bounded utilization periods, an assumption which demands a relatively complex solution.

The approach is based on the strategy used in [10], [14] to solve performability, once the performance (reward) rates were obtained via a queueing model analysis. (There, in reward terms, the reward rate of a given structure state was taken to be the normalized throughput rate; performance was viewed as average reward, i.e., total reward divided by the duration of utilization.) This earlier work, however, does not suggest specific techniques for implementing the prescribed steps. The computational example presented in [10], [14] (namely a 3-state process) is solved using graphical arguments to determine appropriate regions of integration. Such an approach becomes more difficult when the number of states is four and becomes intractable when the number of states is five or more. This paper describes a solution technique wherein regions of integration are determined in a tractable, algorithmic fashion. Section II discusses the model; Section III presents the solution and describes a program we have written for implementing the solution procedure. Finally, Section IV presents an example which illustrates the use of this methodology.

## II. THE MODEL

### A. Reward Models

We consider reward models of the type discussed in [24], although the stochastic process need not be semi-Markovian. We restrict our attention, however, to the case where reward is determined solely by reward rates (referred to in [28]-[30] as “operational rates”) associated with states of the model. We assume further that these rates are constant, i.e., time-invariant. More general reward structures (time-varying rates and “bonuses” associated with state transitions) might also be accommodated by our approach, but they are not considered in the presentation that follows.

A state, in this context, typically represents a certain “operational status” of the system, including configurations wherein the system is not operating (system failure). In a given state, the associated reward rate reflects the pace at which the system rewards its user. Such rates can thus be identified with aspects of system performance such as productivity, responsiveness, and utilization, or, at a higher level, with broader measures such as economic benefit. We discuss reward rates in greater detail below—what such rates can represent and how to obtain them.

More formally, let  $X_\tau$  be the random variable denoting the state of the system at time  $\tau$ .

Accordingly, the base model of the system is the stochastic process

$$X = \{X_\tau \mid \tau \in [0, \infty)\} . \quad (1)$$

For reasons given later in the discussion, we restrict  $X$  to be finite-state, i.e.,  $X_\tau \in Q = \{q_N, q_{N-1}, \dots, q_0\}$ . Suppose further that each  $q_i \in Q$  has an associated reward rate  $r_i$  (a nonnegative real number). Then there is a natural function

$$r: Q \rightarrow \mathbb{R}^{0,+} \quad (2)$$

where  $r(q_i) = r_i$  is the rate associated with  $q_i$ .  $r$  is referred to as a **reward structure** of  $X$ . Let  $\bar{X}_\tau$ ,  $\tau \in [0, \infty)$ , be a random variable [taking values in  $r(Q)$ ] representing the reward rate of  $S$  at time  $\tau$ , that is

$$\bar{X}_\tau = r(X_\tau) . \quad (3)$$

Then the stochastic process

$$\bar{X} = \{\bar{X}_\tau \mid \tau \in [0, \infty)\} . \quad (4)$$

is a **reward model** of the system. Such reward models are a special class of the type of ‘‘operational models’’ investigated by Wu [30]. (Also, compare with the ‘‘capacity models’’ of Gay [20], Gay and Ketelsen [3].)

The performability models we consider consist of a reward model  $\bar{X}$  along with a performance (reward) variable  $Y$  representing the total reward accrued during utilization of the system. More precisely, given a reward model  $\bar{X}$  and a utilization period  $T = [0, t]$ , we take  $Y$  to be the random variable

$$Y = \int_0^t \bar{X}_\tau d\tau . \quad (5)$$

Equivalently, in terms of the base model  $X$  and the reward structure  $r$ ,

$$Y = \int_0^t r(X_\tau) d\tau . \quad (6)$$

### B. An Example of a Reward Based Performability Model

Consider a computing system  $S$  consisting of  $N$  processors, each processor having a computational capacity of  $\delta$  jobs/hour. In addition, there is an air conditioner which does not affect the computational capacity of the system (though it may affect the failure characteristics of the processors). The base model  $X$  is a  $(2 \cdot (N+1))$ -state (not necessarily Markov) stochastic process whose state-transition diagram appears in Fig. 1. In state  $(i, j)$ ,  $i \in \{0, 1, \dots, N\}$  denotes the number of operational processors and  $j$  is 1 if the air conditioner is operational and 0 otherwise. Assuming system capacity is proportional to the number of operational processors, the reward structure is the function  $r(i, j) = i \cdot \delta$ . Accordingly, the reward model  $\bar{X}$  has  $N+1$  states corresponding to the  $N+1$  different reward rates. Relative to a specified utilization period  $T$ , the performance (reward) variable  $Y$  is the

number of jobs processed during  $T$ .

### C. Determination of Reward Rates

An important consideration when applying the methodology described in this paper is the interpretation and derivation of the reward rates. In this section, we delineate several possible interpretations and quantifications of reward. This survey is by no means complete; for a slightly more involved discussion, see [31].

1) *Capacity and Workload*: One basic measure of a computing system's reward rate is the speed at which the system is able to perform computations. This is commonly referred to as *capacity*. Work of Beaudry [2], Gay [20], Gay and Ketelsen [3], Castillo and Siewiorek [6], Oda, Tohma, and Furuya [12], and Munarin [18] model capacity. A related concept is that of *workload*, which is the demand for computation placed on the system by the environment. Studies by Gay [20] and Gay and Ketelsen [3] examine workload models.

2) *Queues and Networks of Queues*: A detailed view of the system's behavior often can be obtained by studying queueing models of the system. Examples of such models abound in the performance evaluation literature; see Kobayashi [32], Ferrari [33], Chandy and Sauer [34], and Trivedi [35]. For instance,  $S$  could be a multiprocessor ( $k$  servers) and a buffer (queue) for storing arriving tasks;  $S$  is then a  $G/G/i$  queueing system. If arrival and service rates are exponential and the buffer length is  $L$ , then  $S$  is the  $M/M/k/L+k$  system considered by Meyer [14].

The reward rate and total reward associated with a queueing system depends on what is of concern to the analyst. One could associate reward with throughput rate, system (or service) time of customers, server utilization, number of customers in the system, etc. In [14], for instance, the reward was based on a "normalized" throughput rate.

3) *Profit and Expense*: An important measure of the reward derived from a system is the profit (say, in terms of dollars) obtained from the system, or, in a negative context, the expense of the system. Each system state has an associated rate of profit or expense. Such values may be obtained, for instance, by economic analysis or by utility techniques, e.g., Raiffa [36]. Koren and Berg [8] and Huslende [17] have based analyses on economic factors.

4) *Control Theoretic*: When the system is a real-time control processor, the reward rate associated with a given state could be specified as functions of such control theoretic concepts as response time. For instance, Krishna and Shin [15] have examined such descriptions, while Gai and Adams [19] have discussed tradeoffs between "optimal" and "robust" response times.

### D. Nonrecoverable Processes

Consider the "desired" behavior of a degradable computing system used over a bounded period of time. Given a set of available resources, a well-designed degradable computing system configures itself to *maximize* the reward rate. Thus, when a component fails, the system does not reconfigure itself in a manner that causes the reward rate to be *greater* than before the failure. We will assume there is no repair (i.e., component replacement via an external source), so an increase in the reward rate due to the acquisition of additional components cannot occur. Transient faults that could lower the reward rate temporarily, thereby raising the rate again when the fault is corrected, are not be considered. Under these conditions, the reward rate of the system is non-increasing in time, which has the interpretation that the system does not become a "better" system after a change in state (e.g., a component failure). These assumptions are formalized by requiring that the reward model  $\bar{X}$  be a *nonrecoverable process*[30] relative to the usual ordering of real numbers. In other words, for all states

$q_1, q_2 \in Q$  and all times  $\tau, \nu \in [0, \infty)$  such that  $\tau \leq \nu$ , we require that

$$\text{Prob}[\bar{X}_\tau = r(q_1) \text{ and } \bar{X}_\nu = r(q_2)] > 0 \Rightarrow r(q_2) \leq r(q_1). \quad (7)$$

Consider the process  $X$  of the multiprocessor/air conditioner example discussed above. Clearly, by the transition diagram of Fig. 1 and the definition of the reward structure, the reward rate of the process cannot increase (with positive probability). Hence, the associated reward model  $\bar{X}$  is an example of a nonrecoverable process.

#### E. Acyclic Processes

An essential step of our model solution procedure (Section III) is to first evaluate conditional performabilities, conditioned on the sequence of states entered by the process during the unbounded period  $[0, \infty)$ . For this reason, we restrict our attention to finite-state base models. To insure a finite number of state sequences and hence conditions, we also require that the base model be "acyclic" in the sense that states are not revisited by the process. More precisely, let  $N_i$  be the random variable denoting the total number of visits of the process  $X$  to state  $q_i \in Q$  during the interval  $[0, \infty)$ .  $X$  is an *acyclic process* if, for all  $q_i \in Q$ ,  $\text{Prob}[N_i > 1] = 0$ .

When evaluating fault-tolerant systems where repair is not allowed, the acyclic condition is not restrictive since once the system leaves a state due to a fault, the system can never return to that state. Often, however, one is interested in analyzing systems with transient faults or temporary failures. Models of such systems are generally cyclic because if the system recovers from the fault, the system returns to some previously entered state. Such recoveries can occur an unbounded number of times during  $[0, t]$ , resulting in an infinite number of state sequences.

One approach for approximating cyclic base models is to "unravel" the process into an acyclic base model. This is done by simply choosing a finite set of state sequences to consider. For example, the set chosen can be those sequences which have at least some threshold probability of occurring during the utilization period  $[0, t]$ . In other words, those sequences of sufficiently low probability are ignored. Alternatively, the set could include all sequences less than or equal to some specified length.

A second approach is to "lump" all the states in each cycle. Because each state in a cycle must have the same reward rate (or else  $\bar{X}$  would not be nonrecoverable), the reward rate of each lumped state is the rate of each of the component states. The difficulty with this approach is determining the equivalent stochastic behavior of the lumped model.



### III. MODEL SOLUTION

#### A. A Partition of the Trajectory Space

As noted in [10], one approach to determining performability is to partition the trajectory space and solve for each of the resulting classes of trajectories. During the unbounded period  $[0, \infty)$ , the base model process  $X$  will, with probability 1, pass through some finite sequence of distinct states, say  $\mathbf{u} = (u_n, u_{n-1}, \dots, u_0)$ , where state  $u_n$  is the initial state and state  $u_0$  is an absorbing state.  $X$  is nonrecoverable (see above) so there is a sequence of reward rates  $(r(u_n), r(u_{n-1}), \dots, r(u_0))$ , corresponding to  $\mathbf{u}$ , such that

$$r(u_n) \geq r(u_{n-1}) \geq \dots \geq r(u_0) \quad (8)$$

Let  $\mathbf{U}$  be a random variable denoting the sequence of states visited by the base model during  $[0, \infty)$ . Since the base model is acyclic, and since there are  $N < \infty$  base model states, there are only a finite number of possible state sequences, i.e., sequences  $\mathbf{u}$  such that  $\text{Prob}[\mathbf{U} = \mathbf{u}] > 0$ . Let  $Y$  be the performance (reward) variable defined in (5) and let  $F_Y$  be the PDF of  $Y$ . Further, if  $F_{Y|\mathbf{U}}$  is the conditional PDF of  $Y$  given  $\mathbf{U}$ , then  $F_Y$  can be expressed as the following summation over a finite index set:

$$F_Y(y) = \sum_{\mathbf{u}} F_{Y|\mathbf{U}}(y|\mathbf{u}) \text{Prob}[\mathbf{U} = \mathbf{u}] \quad (9)$$

#### B. Notation

At this point, it is convenient to introduce a body of notation for dealing with the time the process spends in various states during both  $T = [0, t]$  and  $[0, \infty)$ . Unless otherwise noted, the remarks that follow assume  $\mathbf{U}$  to be some arbitrary sequence  $\mathbf{u} = (u_n, u_{n-1}, \dots, u_0)$  such that  $\text{Prob}[\mathbf{U} = \mathbf{u}] > 0$ .

For each state sequence  $\mathbf{u} = (u_n, u_{n-1}, \dots, u_0)$  there is a vector-valued random variable  $\mathbf{V}_{\mathbf{u}} = (V_n, V_{n-1}, \dots, V_0)$  taking values in  $(\mathbb{R}^{0,+})^n$ , which describes the time the process resides in each state in  $\mathbf{u}$ . For  $0 \leq i \leq n$ ,  $V_i$  is a random variable representing the time of the base model process resides in state  $u_i$ , during the interval  $[0, \infty)$ . State  $u_0$  is an absorbing state, and so  $v_0$  is  $\infty$ . We will be interested in the PDF for  $\mathbf{V}_{\mathbf{u}}$  conditioned on  $\mathbf{U} = \mathbf{u}$ , i.e.,

$$F_{\mathbf{V}_{\mathbf{u}}|\mathbf{U}}(\mathbf{v}_{\mathbf{u}}|\mathbf{u}) = \text{Prob}[\mathbf{V}_{\mathbf{u}} \leq \mathbf{v}_{\mathbf{u}} | \mathbf{U} = \mathbf{u}] \quad (10)$$

$$= \text{Prob}[V_n \leq v_n \wedge V_{n-1} \leq v_{n-1} \wedge \dots \wedge V_0 \leq v_0 | \mathbf{U} = \mathbf{u}]$$

If the conditional probability density function for  $\mathbf{V}_{\mathbf{u}}$  exists, it will be written  $f_{\mathbf{V}_{\mathbf{u}}}$ . We suppose that  $f_{\mathbf{V}_{\mathbf{u}}}$  exists and we find it convenient to expand  $f_{\mathbf{V}_{\mathbf{u}}}$  as follows:

$$f_{\mathbf{V}_{\mathbf{u}}|\mathbf{U}}(\mathbf{v}_{\mathbf{u}}|\mathbf{u}) = f_{V_n|\mathbf{U}}(v_n|\mathbf{u}) f_{V_{n-1}|V_n, \mathbf{U}}(v_{n-1}|v_n, \mathbf{u}) \dots \quad (11)$$

$$f_{V_0|V_n, V_{n-1}, \dots, V_1, \mathbf{U}}(v_0|v_n, v_{n-1}, \dots, v_1, \mathbf{u})$$

For a given  $\mathbf{u}$ , define  $W_{t'}^{\mathbf{u}}$  to be a random variable denoting the amount of time the process is in state  $u_i$ , during the utilization period  $T$ . The basic relationship between  $W_{t'}^{\mathbf{u}}$  and  $V_i$

can be expressed as

$$W_{t,i} = \begin{cases} \min(t, V_n), & \text{if } i = n \\ \max(\min(t - \sum_{j=i+1}^n V_j, V_i), 0), & \text{otherwise,} \end{cases} \quad (12)$$

Note that if  $\mathbf{U} = \mathbf{u}$  and if  $W_{t,i} = 0$  and  $V_i \geq 0$ , then  $W_{t,i-1}, W_{t,i-2}, \dots, W_{t,0} = 0$ . Also, note that the value of  $W_{t,i}$  can always be expressed by one of the following alternatives:  $V_i, t - \sum_{j=i+1}^n V_j$ , or 0.

### C. The Approach

We seek to determine the PDF  $F_Y$  of the reward variable  $Y$  (Eq. 9). The algorithm delineated in [10] will be employed:

**Algorithm 1:** (Determining the probability distribution function,  $F_Y$ )

- 1) Determine all state sequences  $\mathbf{u}$ , i.e., the range of the random variable  $\mathbf{U}$ .
- 2) For each of these  $\mathbf{u}$ , determine  $\text{Prob}[\mathbf{U} = \mathbf{u}]$ .
- 3) For each  $\mathbf{u}$  such that  $\text{Prob}[\mathbf{U} = \mathbf{u}] > 0$ , determine  $F_{Y|\mathbf{U}}(y)$ .
- 4) Apply Eq. 9 to arrive at  $F_Y$ .

For the present, we assume that steps 1) and 2) have been completed. Consider step 3), the calculation of  $F_{Y|\mathbf{U}}$ .

Using the notation introduced in the previous section, the reward variable  $Y$  can be directly expressed as a linear combination of the random variables  $W_{t,i}$ . If  $\mathbf{U} = \mathbf{u}$ , then

$$Y = \sum_{j=0}^n r(u_j) W_{t,j} \quad (13)$$

If the  $W_{t,j}$  were independent random variables, we could obtain the PDF  $F_Y$  by mathematical convolution. However, as discussed in [10], [14], the  $W_{t,i}$  are statistically dependent and a probabilistic characterization is generally difficult to obtain. Such complications are avoided by formulating  $Y$  as a function  $\gamma_{\mathbf{u}}$  of  $\mathbf{V}_{\mathbf{u}}$  (if it is known that  $\mathbf{U} = \mathbf{u}$ ). Although the basic concept is simple to state, the details are somewhat complex.

Using the relationship between  $W_{t,i}$  and  $V_i$  of Eq. 12, (see Eq. 6)

$$Y = \gamma(\mathbf{V}_{\mathbf{u}}) = \begin{cases} r(u_j)t + \sum_{k=j+1}^n (r(u_k) - r(u_j)) V_k & \text{if } \sum_{k=j+1}^n V_k < t, \sum_{k=j}^n V_k \geq t, \text{ for } n > j \\ r(u_n)t & \text{if } V_n \geq t. \end{cases} \quad (14)$$

See Fig. 2 and compare with Eq. 24 of [14].

Let  $B_y$  be the set of accomplishment levels (reward outcomes) not greater than  $y$ , i.e.,  $B_y = \{b \geq 0 \mid b \leq y\}$  and let  $\mathbf{C}_y = \gamma_{\mathbf{u}}^{-1}(B_y)$  be the set of all base model trajectories which traverse the state sequence  $\mathbf{u}$  and provide reward no greater than  $y$ . Then

$$F_{Y|U}(y|\mathbf{u}) = \text{Prob}[\mathbf{V}_{\mathbf{u}} \in \mathbf{C}_y \mid \mathbf{U} = \mathbf{u}] = \int_{\mathbf{C}_y} f_{\mathbf{V}_{\mathbf{u}}|U}(\mathbf{v}_{\mathbf{u}}|\mathbf{u}) d\mathbf{v}_{\mathbf{u}}. \quad (15)$$

Since the probability density function  $f_{\mathbf{V}_{\mathbf{u}}|U}(\cdot)$  is assumed to exist and be known, to formulate  $F_{Y|U}$  we must determine  $\mathbf{C}_y$ .

#### D. Formulation of $F_{Y|U}$

In the next section, we show that, depending on the values of  $\mathbf{u}$  and  $y$ ,  $\mathbf{C}_y$  can be easily partitioned, i.e., expressed as a union of disjoint regions. Such a partition provides the ability to break up  $F_{Y|U}$  into a sum of smaller, less complex integrations. In particular, if the length of the sequence  $\mathbf{u}$  is  $n+1$ ,  $\mathbf{C}_y$  will be decomposed into  $n+2$  disjoint regions

$$\{\mathbf{C}_y^{n+1}, \mathbf{C}_y^n, \dots, \mathbf{C}_y^0\}. \quad (16)$$

Let  $\mathbf{v}_{\mathbf{u}}$  be a point in  $\mathbf{C}_y^i$  and let

$$\mathbf{C}_{y,n}^i = \left\{ v_n \mid \text{there exist } v_{n-1}, v_{n-2}, \dots, v_0 \text{ such that } (v_n, v_{n-1}, \dots, v_0) \in \mathbf{C}_y^i \right\}, \quad (17)$$

and for  $n > j \geq 0$ , let

$$\begin{aligned} \mathbf{C}_{y,j}^i(v_n, v_{n-1}, \dots, v_{j+1}) &= \left\{ v_j \mid \text{there exist } v_{j-1}, v_{j-2}, \dots, v_0 \right. \\ &\quad \left. \text{such that } (v_n, v_{n-1}, \dots, v_0) \in \mathbf{C}_y^i \right\} \end{aligned} \quad (18)$$

In other words,  $\mathbf{C}_{y,j}^i(v_n, v_{n-1}, \dots, v_{j+1})$  is the set of all values  $v_j$  that are ‘‘contained’’ in some  $\mathbf{v}_{\mathbf{u}} \in \mathbf{C}_y^i$ . For conciseness, we shall write  $\mathbf{C}_{y,j}^i$  in place of  $\mathbf{C}_{y,j}^i(v_n, v_{n-1}, \dots, v_{j+1})$  when the  $(v_n, v_{n-1}, \dots, v_{j+1})$  is implicit.

Consider  $\gamma_{\mathbf{u}}$  (Eq. 14). The regions  $\mathbf{C}_y^i$  are *not* generally Cartesian, that is,  $\mathbf{C}_y^i$  is usually not expressible as the cross-product of the  $\mathbf{C}_{y,j}^i$ . Rather, a different relationship based on ‘‘i-resolvability’’ (see the next section) will be employed to define the  $\mathbf{C}_y^i$ .

By the definitions of the  $\mathbf{C}_y^i$  and  $\mathbf{C}_{y,j}^i$ , Eq. 15 can now be expressed as

$$\begin{aligned} F_{Y|U}(y|\mathbf{u}) &= \sum_{i=0}^{n+1} \text{Prob}[\mathbf{V}_{\mathbf{u}} \in \mathbf{C}_y^i \mid \mathbf{U} = \mathbf{u}] = \sum_{i=0}^{n+1} \int_{\mathbf{C}_y^i} f_{\mathbf{V}_{\mathbf{u}}|U}(\mathbf{v}_{\mathbf{u}}|\mathbf{u}) d\mathbf{v}_{\mathbf{u}} \\ &= \sum_{i=0}^{n+1} \int_{\mathbf{C}_{y,n}^i} \int_{\mathbf{C}_{y,n-1}^i} \cdots \int_{\mathbf{C}_{y,0}^i} f_{\mathbf{V}_{\mathbf{u}}|U}(\mathbf{v}_{\mathbf{u}}|\mathbf{u}) d\mathbf{v}_{\mathbf{u}} \end{aligned} \quad (19)$$

The expansion of  $f_{\mathbf{V}_{\mathbf{u}}|U}(\mathbf{v}_{\mathbf{u}}|\mathbf{u})$  in Eq. 11 is especially appropriate in this representation. Finally, combining Eqs. 9, 11, and 19 with the characterization of the  $\mathbf{C}_{y,j}^i$  developed below,

we arrive at an integral solution for the PDF  $F_Y$ .

#### D. *i-resolvability*

In this section, we describe a method of partitioning the set  $C_y = \gamma_u^{-1}(B_y)$  into the  $C_y^i$  (Eq. 16) and characterizing  $C_y^i$  (Eqs. 17 and 18). The immediate difficulty with characterizing the  $C_y^i$  arises from the nature of  $\gamma_u$  (Eq. 14), viz.,  $\gamma_u$  is a sum of a random number of random variables  $V_n, V_{n-1}, \dots, V_{j+1}$ . We introduce the notion of “*i-resolvability*” which allows the decomposition of the single large problem of describing the sum of a varying number of random variables into a set of smaller problems, each consisting of describing the sum of a fixed number of random variables. The number of random variables to be considered will be determined by *i-resolvability*, a notion which takes full advantage of both the monotonicity of  $\gamma_u$  and the finite utilization period. The concept can be loosely stated: “For a given sub-trajectory, based on the past and regardless of the future, what can we say about the entire trajectory?”

We are interested in determining the probability that the total accumulated reward  $\gamma_u(\mathbf{V}_u) \leq y$ . Recall that the time the process resides in each state of the sequence  $\mathbf{u}$  is given by the random variable  $\mathbf{V}_u = (v_n, v_{n-1}, \dots, v_0)$ . Suppose that the sojourn times for the first  $n-i+1$  of these values are known, i.e.,  $(V_n, V_{n-1}, \dots, V_i) = (v_n, v_{n-1}, \dots, v_i)$ . Knowledge of these times conveys significant information regarding the set of possible sojourn times  $(V_{i-1}, V_{i-2}, \dots, V_0)$  that can be associated with the remaining states and still satisfy  $\gamma_u(\mathbf{V}_u) \leq y$ . As an informal introduction, imagine that someone chooses at random some vector  $\mathbf{v}_u$  of sequence times and starts dealing, in order and one at a time, the values  $v_n, v_{n-1}, \dots, v_0$ . We look at the values as they are dealt. If *after* the value  $v_i$  has been dealt—but not before—we can determine with certainty that  $\gamma_u(\mathbf{v}_u) \leq y$ , then we shall say that  $\mathbf{v}_u$  is such that  $\gamma(\mathbf{v}_u) \leq y$  is *i-resolvable based on  $\mathbf{v}_u$* , or, more succinctly,  $\mathbf{v}_u$  is *i-resolvable* ( $y$  and  $\gamma_u$  are implicit). If at any time we can determine with certainty that  $\gamma_u(\mathbf{v}_u) > y$ , then  $\mathbf{v}_u \notin C_y$ ; we are not interested in that value  $\mathbf{v}_u$  (since it does not contribute to  $\text{Prob}[Y \leq y]$ ) and so we reject it.

We now formalize this property. Let  $\gamma_u: \mathbb{R}^n \rightarrow \mathbb{R}$  and  $\mathbf{v}_u = (v_n, v_{n-1}, \dots, v_0) \in \mathbb{R}^n$  be such that  $\gamma_u \leq y$ . In addition, let  $\gamma_u(v_n, v_{n-1}, \dots, v_i, \bullet, \bullet, \dots, \bullet)$  be the  $(n-i+1)$ -variable function derived from  $\gamma_u(\bullet, \bullet, \dots, \bullet)$  by setting  $V_n = v_n, V_{n-1} = v_{n-1}, \dots, V_i = v_i$ . When  $y$  is an upper bound<sup>1</sup> of the function  $\gamma_u(v_n, v_{n-1}, \dots, v_i, \bullet, \bullet, \dots, \bullet)$  but not of  $\gamma_u(v_n, v_{n-1}, \dots, v_{i+1}, \bullet, \bullet, \dots, \bullet)$ , we shall say that  $\mathbf{v}_u$  is *i-resolvable*. [As a special case, when  $y$  bounds  $\gamma_u(\bullet, \bullet, \dots, \bullet)$ , then  $\mathbf{v}_u$  is  $(n+1)$ -resolvable.] In other words, when a determination as to whether  $\gamma_u(\mathbf{v}_u) \leq y$  can not be made based solely on the values  $(v_n, v_{n-1}, \dots, v_{i+1})$ , but such a determination can be made with the additional knowledge of the value of  $v_i$ , we shall say that  $\mathbf{v}_u$  is *i-resolvable*. [When  $\mathbf{v}_u$  is  $(n+1)$ -resolvable, then the determination can be made with no knowledge of  $\mathbf{v}_u$ .]

One can define *i-resolvability* recursively, as follows:

For a given  $\mathbf{v}_u = (v_n, v_{n-1}, \dots, v_0)$ :

- i)  $\mathbf{v}_u$  is  $(n+1)$ -resolvable if and only if  $\gamma_u(\mathbf{v}_u) \leq y$  regardless of the value  $\mathbf{v}_u$ .
- ii)  $\mathbf{v}_u$  is *i-resolvable* if  $\mathbf{v}_u$  is not *j-resolvable* for all  $j > i$  but  $\gamma_u(v_n, v_{n-1}, \dots, v_i, \bullet, \bullet, \dots, \bullet) \leq y$  regardless of the values of  $(v_{i-1}, v_{i-2}, \dots, v_0)$ .

To motivate our interest in *i-resolvability*, consider the following example. Suppose  $\mathbf{U}$  is such that  $(r(u_2), r(u_1), r(u_0)) = (2, 1, 0)$  and that  $y = 5$  and  $t = 4$ . If  $V_2 = 1$ , then

---

<sup>1</sup>A constant  $y$  is a **upper bound** of the function  $f: D \rightarrow R$  if for all  $x \in D$ ,  $f(x) \leq y$ .

for  $\gamma_{\mathbf{u}}(\mathbf{V}_{\mathbf{u}}) \leq y$  to be true, we must have  $W_{t^1} \leq t - V_2 = 4 - 1 = 3$  (see Eq. 12), and

$$\sum_{i=0}^2 r(u_i) W_{t^i} = 1 \cdot W_{t^1} + 1 \cdot 2 = W_{t^1} + 2 \leq 3 + 2 = 5 = y. \quad (20)$$

Now  $W_{t^1} \leq 3$  irrespective of  $V_1$ , so  $V_1$  can therefore take on *any* value in the interval  $[0, \infty)$  and still satisfy  $\gamma_{\mathbf{u}}(\mathbf{V}_{\mathbf{u}}) \leq y$ . In other words, since  $V_2 = 1$  then, without further examination of  $V_1$  or  $V_0$ , we know  $\gamma_{\mathbf{u}}(\mathbf{V}_{\mathbf{u}}) \leq y$ . Thus, every  $(1, V_1, V_0)$  is 2-resolvable.

### E. Solution

The partition  $\{\mathbf{C}_y^i\}$  of  $\mathbf{C}_y$  that we desire is the one induced by  $i$ -resolvability. Specifically, let a trajectory  $\mathbf{v}_{\mathbf{u}} \in \mathbf{C}_y$  be in  $\mathbf{C}_y^i$  if  $\mathbf{v}_{\mathbf{u}}$  is  $i$ -resolvable. One can easily see that every  $\mathbf{v}_{\mathbf{u}} \in \mathbf{C}_y$  is  $i$ -resolvable for exactly one  $i$ . The recursive definition of “ $i$ -resolvability” yields the following algorithm for determining the  $\mathbf{C}_y^i$ :

**Algorithm 2:** (Determining the  $\mathbf{C}_y^i$ )

For a given  $B_y$ ,

- i) Determine  $\mathbf{C}_y^{n+1}$ , i.e., all  $\mathbf{v}_{\mathbf{u}} \in \gamma_{\mathbf{u}}^{-1}(B_y)$  such that  $\mathbf{v}_{\mathbf{u}} \in \gamma_{\mathbf{u}}^{-1}(B_y)$  regardless of the value of  $\mathbf{v}_{\mathbf{u}}$ . These are the  $\mathbf{v}_{\mathbf{u}}$  which are  $(n+1)$ -resolvable. ( $\mathbf{C}_y^{n+1}$  will either be empty or all  $\mathbf{C}_y$ , depending on the value of  $y$ .)
- ii) For each  $i = n, n-1, \dots, 0$ , determine all  $\mathbf{v}_{\mathbf{u}} \in \gamma_{\mathbf{u}}^{-1}(B_y)$  such that  $\mathbf{v}_{\mathbf{u}} \notin \mathbf{C}_y^{n+1} \cup \mathbf{C}_y^n \cup \mathbf{C}_y^{n-1} \cup \dots \cup \mathbf{C}_y^{i+1}$  and  $\mathbf{v}_{\mathbf{u}} \in \gamma_{\mathbf{u}}^{-1}(B_y)$  regardless of the values of  $(v_{i-1}, v_{i-2}, \dots, v_0)$ . These are the  $\mathbf{v}_{\mathbf{u}}$  which are  $i$ -resolvable.

To show that a given vector  $\mathbf{v}_{\mathbf{u}}$  is  $i$ -resolvable, the definition of  $i$ -resolvability may be used directly. Unfortunately, space does not allow the full derivation of the characterization of the  $\mathbf{C}_y^i$  (see [31]). An important intermediate result is the following (compare with Eq. 14 and see Eq. 12 for the relation between  $V_i$  and  $W_{t^i}$ ):

$\mathbf{v}_{\mathbf{u}}$  is  $i$ -resolvable ( $n \geq i > 0$ ) if and only if

$$y \geq r(u_{i-1})t + \sum_{j=i}^n (r(u_j) - r(u_{i-1}))w_{t^j} \quad (21)$$

and

$$y < \begin{cases} r(u_i)t + \sum_{j=i+1}^n (r(u_j) - r(u_i))w_{t^j} & \text{if } n > i > 0 \\ r(u_n)t & \text{if } i = n \end{cases} \quad (22)$$

As mentioned above, the  $w_{t^i}$  are statistically dependent and thus inconvenient to use. Hence, Eqs. 21 and 22 must be rewritten in terms of the  $v_i$  (see Eq. 12). We use the properties of  $i$ -resolvability and the monotonicity of  $Y$  to obtain the main result of this section:

Characterization of  $C_{y,j}^i$ :

i) If  $r(u_i) = r(u_{i-1})$ ,  $n \geq i > 0$ , then no  $\mathbf{v}_u$  is  $i$ -resolvable.  $\mathbf{C}_y^i = \phi$  and for each  $j$  such that  $n \geq k \geq 0$ :

$$C_{y,j}^i = \phi \quad (23)$$

ii) Let  $y \in [r(u_n)t, \infty)$ . Then every  $\mathbf{v}_u$  is  $n+1$ -resolvable.  $\mathbf{C}_y^{n+1} = \mathbf{C}_y$ ,  $\mathbf{C}_y^i = \phi$  for  $n \geq i \geq 0$ , and for each  $j$  such that  $n \geq j \geq 0$  and for each  $k$  such that  $n \geq k \geq 0$ :

$$C_{y,j}^{n+1} = [0, \infty) \quad (24)$$

$$C_{y,j}^k = \phi. \quad (25)$$

iii) Let  $y \in [r(u_{l-1})t, r(u_l)t)$ ,  $n \geq l > 0$ . Then for every  $i$  such that  $l \geq i > 0$  and  $r(u_i) \neq r(u_{i-1})$ , there exist vectors  $\mathbf{v}_u$  such that  $\mathbf{v}_u$  is  $i$ -resolvable.

a) If  $i = n$ , then

$$i) C_{y,n}^n = \left[ 0, \frac{y - r(u_{n-1})t}{r(u_n) - r(u_{n-1})} \right] \quad (26)$$

and for each  $j$  such that  $n > j \geq 0$

$$ii) C_{y,j}^n = [0, \infty), \quad n > j > 0 \quad \text{and} \quad (27)$$

$$iii) C_{y,0}^n = \infty. \quad (28)$$

b) If  $i \neq n$ , then

$$i) C_{y,n}^i = \left( \frac{y - r(u_{n-1})t}{r(u_n) - r(u_{n-1})}, \frac{y - r(u_{i-1})t}{r(u_n) - r(u_{i-1})} \right) \quad (29)$$

and for each  $j$  such that  $n > j > i$

$$\text{ii) } C_{y,j}^i = \left( \begin{array}{c} y - r(u_{j-1})t - \sum_{k=j+1}^n (r(u_k) - r(u_{j-1}))v_k \\ \hline r(u_j) - r(u_{j-1}) \end{array} \right), \quad (30)$$

$$\left. \begin{array}{c} y - r(u_{i-1})t - \sum_{k=j+1}^n (r(u_k) - r(u_{i-1}))v_k \\ \hline r(u_j) - r(u_{i-1}) \end{array} \right]$$

and for  $j = i$

$$\text{iii) } C_{y,i}^i = \left[ 0, \frac{y - r(u_{i-1})t - \sum_{k=j+1}^n (r(u_k) - r(u_{i-1}))v_k}{r(u_i) - r(u_{i-1})} \right] \quad (31)$$

and for each  $j$  such that  $\bar{i} > j \geq 0$

$$\text{iv) } C_{y,j}^i = [0, \infty), \quad i > j > 0 \text{ and} \quad (32)$$

$$C_{y,0}^i = \infty.$$

c) For each  $k$  such that  $n + 1 \geq k > l$  (or  $k = 0$ ) and for each  $j$  such that  $n \geq j \geq 0$

$$C_{y,j}^k = \phi. \quad (33)$$

where the notation  $\left( \begin{array}{c} a, b \\ \geq 0 \end{array} \right)$  means

$$\left( \begin{array}{c} a, b \\ \geq 0 \end{array} \right) = \begin{cases} [0, b] & \text{if } a \leq 0 \\ (a, b] & \text{if } a > 0 \end{cases} \quad (34)$$

If  $r(u_j) = r(u_{j-1})$ , then in Eqs. 29 and 30, terms with the denominator  $r(u_j) - r(u_{j-1})$  [or  $r(u_n) - r(u_{n-1})$ ] are replaced by 0, and Eqs. 26 and 31 are replaced by  $\phi$  [see case i)]. Also, in Eq. 30, if  $r(u_j) - r(u_{i-1})$  for  $n > j > i$ , then  $r(u_i) = r(u_{i-1})$  and Eq. 30 is replaced by  $\phi$  [see case i)].

An integral solution for the PDF  $F_Y$  (and hence the system's performability) is thus obtained by employing Eqs. 9, 11, 19, and 24-33.

### F. METAPHOR

The remaining difficulty is calculating the integrations. For certain classes of probability densities  $f_{v_i | v_n, v_{n-1}, \dots, v_{i+1}, u}$ , it may be possible to perform symbolically the integrations to obtain a true closed-form solution. For example, a closed-form solution for the performability of a queueing system is derived in [14]. The symbolic integrations may be performed either (laboriously) by hand or by computer programs that manipulate symbolic quantities, e.g., MAXIMA [37] and REDUCE2 [38]. A recursive version of Eqs. 19 and 24-33 is derived in [31] and may help in reducing the amount of work necessary to perform symbolically the required integrations.

However, when performability is evaluated for complex systems with large state spaces, one is primarily interested in numerical results as opposed to closed-form solutions. For this purpose, we have written a numerical program based on Eqs. 9, 11, 19, and 24-33. This program is now part of a larger performability evaluation package, METAPHOR (see [22], [31] for descriptions of METAPHOR). To obtain a system's performability, METAPHOR is given information about each trajectory sequence  $u$ , including the sequence itself, the conditional densities  $f_{v_i | v_n, v_{n-1}, \dots, v_{i+1}, u}$ , and the utilization period  $T = [0, t]$ . METAPHOR then computes the regions of integrations  $C_{y,j}^i$  and using Gaussian quadrature, computes  $F_Y | u$ .

## IV. MULTIPROCESSOR/AIR CONDITIONER EXAMPLE

To illustrate the application of the solution procedure for  $F_Y$  and to exhibit its ability to deal with non-semi-Markov base models, we consider the multiprocessor/air conditioner example discussed in Section II. B. (See Fig. 1). Recall that the performance (reward) variable  $Y$  is the number of jobs processed during some utilization period  $T = [0, t]$ . State  $(i, j)$  reflects the number of processors operational and whether the air conditioner is operational; each processor is identical and has computation rate of  $\delta$  jobs/hour. The reward structure is  $r(i, j) = i \cdot \delta$ . For simplicity, the example constrains the system to a single air conditioner. (A more complete example would include multiple air conditioners.)

Suppose that, relative to a specified threshold  $y$  ( $y > 0$ ), the system user is interested in processing more than  $y$  jobs during the bounded utilization period. For instance, the system might be a computer in a business or university during a working day, or it might be a processor handling financial transactions overnight. Note that especially in the latter category of applications, availability over the entire utilization period is not required since most of the computation could be done early in the period, or, alternatively, spread out over the entire period. In performability terms, we are considering (for a given value of  $y$ ) the set of accomplishment levels  $B^y = \{b | b > y\}$ . The performability  $p(B^y)$ , i.e., the probability that the number of jobs processed (reward)  $Y$  is greater than  $y$ , can then be obtained from  $F_Y$  since  $p(B^y) = \text{Prob}[Y > y] = 1 - F_Y(y)$ .

The stochastic properties of the processors are affected by the failure time of the air conditioner in such a way that the base model is not semi-Markov. At the beginning of the utilization period, the air conditioner and all of the processors are functioning and the temperature of the room containing the equipment is  $20^\circ\text{C}$ . Since the amount of heat which the air conditioner must dissipate places stress on the compressor, the air conditioner's failure rate is influenced by the number of processors it must cool. Assume the air conditioner fails with an exponential failure rate  $\lambda_{AC}(N) = N \cdot 0.05$  failures/hour, where  $N$  is the number of processors in the room. If the air conditioner fails, the ambient temperature in the room  $R$  begins to increase to a higher steady-state temperature with an exponential rise time; the number of processors in the room affects the speed at which the temperature rises. More pre-



cisely,

$$R(\Delta \tau) = 55^\circ + 2N - (55^\circ + 2N - 20^\circ) \exp(-N\mu\Delta \tau), \quad (35)$$

where  $(55^\circ + 2N)^\circ\text{C}$  is the new steady-state temperature,  $\Delta \tau$  is the time (in hours) since the air conditioner failed, and  $\mu = 10$  degrees/hour/processor is a constant reflecting the rate of temperature increase. If a processor fails, it does not shut off and so continues contributing heat to the room. The failure behavior of the processors is influenced by the ambient temperature; each processor fails at an exponential rate which varies linearly with room temperature (over the range  $20^\circ\text{C}$  to  $55^\circ\text{C}$ ) from 0.001 failures/hour to 0.1 failure/hour. If the room temperature is  $R^\circ\text{C}$ ,

$$\lambda_P(R) = 0.001 + \left( \frac{0.1 - 0.001}{55^\circ - 20^\circ} \right) (R - 20^\circ). \quad (36)$$

There are  $N+1$  state sequences  $\mathbf{u}$  such that  $\text{Prob}[\mathbf{U} = \mathbf{u}] > 0$  (see Fig. 1):

$$\begin{aligned} \{\mathbf{u}\} = & \{((N,1),(N,0),(N-1,0), \dots, (0,0)), \\ & ((N,1),(N-1,1),(N-1,0); \dots, (0,0)), \dots, \\ & ((N,1),(N-1,1),(N-2,1), \dots, (0,1),(0,0))\}. \end{aligned} \quad (37)$$

Let  $\mathbf{u} = ((N,1),(N-1,1), \dots, (k,1),(k,0), \dots, (0,0))$ . The probability of the sequence  $\mathbf{u}$  is

$$\text{Prob}[U = \mathbf{u}] = \begin{cases} \frac{\lambda_{AC}}{k\lambda_P + \lambda_{AC}} \times \prod_{j=k+1}^N \frac{j\lambda_P}{j\lambda_P + \lambda_{AC}} & \text{if } k < N \\ \frac{\lambda_{AC}}{N\lambda_P + \lambda_{AC}} & \text{if } k = N. \end{cases} \quad (38)$$

Suppose  $u_i = (j,1)$ ; the conditional probability density function for the time the process spends in state  $u_i$  is

$$\begin{aligned} f_{V_i | V_n, V_{n-1}, \dots, V_{i+1}, \mathbf{U}}(v_i | v_n, v_{n-1}, \dots, v_{i+1}, \mathbf{u}) \\ = (j\lambda_P(20^\circ) + \lambda_{AC}) \exp(-(j\lambda_P(20^\circ) + \lambda_{AC})v_i). \end{aligned} \quad (39)$$

Suppose  $u_i = (j,0)$  and that the first state in  $\mathbf{u}$  representing a failed air conditioner is  $u_k$ ,  $k \geq i$ ; the conditional probability density function for the time the process spends in state  $u_i$  is

$$f_{V_i | V_n, V_{n-1}, \dots, V_{i+1}, \mathbf{U}}(v_i | v_n, v_{n-1}, \dots, v_{i+1}, \mathbf{u}) = \frac{1}{K} j\lambda_P(R(\Delta \tau)) \exp(-j\lambda_P(R(\Delta \tau))v_i) \quad (40)$$

where  $\Delta \tau = v_k + v_{k-1} + \dots + v_i$ , and  $K$  is a normalization constant equal to

$$K = \int_0^{\infty} j \lambda_P(R(\Delta \tau')) \exp(-j \lambda_P(R(\Delta \tau'))v) dv, \quad (41)$$

(where  $\Delta \tau' = v_k + v_{k-1} + \dots + v_{i+1} + v$ ). That the base model is not semi-Markov can be seen from the time-varying failure rates associated with states  $(i,0)$  [Eq. 40]; these rates are functions of the process' history and cannot be inferred from the present state, present time, and time of entry to the state.

It is clear how Eqs. 38, 39, and 40 can be applied to Eqs. 9 and 11. To see how Eqs. 19 and 24-33 are employed to calculate  $F_Y$ , consider  $N = 3$ ,  $\mathbf{u} = ((3,1),(3,0),(2,0),(1,0),(0,0))$  and  $y = 1500 \in [r(u_1)t, r(u_2)t]$ . From Eq. 19, we see that  $F_Y$  is the sum of six multiple integrals, three of which are 0 since they are integrated over empty  $\mathbf{C}_y^i$  (these are  $\mathbf{C}_y^5$ ,  $\mathbf{C}_y^4$ , and  $\mathbf{C}_y^0$ ). The other 3  $\mathbf{C}_y^i$  correspond to the sets of base model trajectories which are 3, 2, and 1-resolvable. Consider  $\mathbf{C}_y^2$ . From Eq. 29:

$$C_{y,4}^2 = \left[ 0, \frac{(1500 - 1000)}{(300 - 100)} \right] = [0, 2.5]; \quad (42)$$

from Eq. 30:

$$\begin{aligned} C_{y,3}^2 &= \left( \frac{1500 - 2000 - (300 - 200)v_4}{300 - 200}, \frac{1500 - 1000 - (300 - 100)v_4}{300 - 100} \right) \\ &= \left[ 0, \frac{500 - 200v_4}{200} \right]; \end{aligned} \quad (43)$$

from Eq. 31:

$$\begin{aligned} C_{y,2}^2 &= \left[ 0, \frac{1500 - 1000 - (300 - 100)v_3 - (300 - 100)v_4}{200 - 100} \right] \\ &= \left[ 0, \frac{500 - 200(v_4 + v_3)}{100} \right]; \end{aligned} \quad (44)$$

and from Eq. 32:

$$\begin{aligned} C_{y,1}^2 &= [0, \infty) \text{ and} \\ C_{y,0}^2 &= \infty. \end{aligned} \quad (45)$$

Once regions  $\mathbf{C}_y^3$  and  $\mathbf{C}_y^1$  are similarly obtained, Eq. 19 can then be evaluated. This process must be repeated for each state sequence  $\mathbf{u}$  and then Eq. 9 can be applied to obtain the performability  $p(B^y)$ .

Fig. 3 shows a plot of  $p(B^y) = 1 - F_Y(y)$  for  $N = 1, 2, 3$ , and 4 processors,  $t = 10$  hours and  $\delta = 100$  jobs/hour. Note that these evaluations provide a considerable amount of information regarding the system's ability to perform (provide reward) in the presence of faults. It is easy to show, for example, that the performability  $p(B^y)$  is 0 when  $y$  is greater than or equal to  $N \cdot \delta \cdot t$  (e.g., for  $N = 2$ ,  $p(B^{2000}) = 0$ ; see Fig. 3). Hence, to obtain a nonzero performability, the number of processors must be greater than  $\frac{y}{\delta \cdot t}$ . For instance, to

have a nonzero probability of accomplishing more than 1500 jobs, one must have at least 2 processors. Generally, for the values of  $N$  shown on the plot, there is a significant gain in  $p(B^y)$  for values of  $y$  above 1000 when additional processors are included in the system. For values of  $y$  below 1000, there is relatively little gain from having more than a single processor. Indeed, if the specified minimum reward is between about 500 and 1000 jobs, a single processor provides a greater probability of performing within  $B^y$  than do two processors.

In non-critical applications, a system designer may choose to settle for a lower performability to avoid the cost of additional processors. Information such as that provided by Fig. 3 can be quite useful in investigating such tradeoffs. For example, suppose the threshold is  $y = 1500$ . The difference between the performability with  $N = 3$  and with  $N = 4$  is about 0.03, while the difference between  $N = 2$  and  $N = 3$  is about 0.12. The probability of accomplishing more than 1500 jobs with 3 processors is about 0.96. If this probability is adequate for the application, and the additional .03 probability is not worth the cost of an extra processor, the designer may well choose a 3-processor system.

## V. CONCLUSION

In summary, the techniques described in this paper provide a new means of evaluating performability for a broad class of systems and for a general class of performance variables. The evaluation procedure is admittedly complex and requires a programmed implementation for practical applications. This is true, however, for most evaluation methods, particularly those applicable to degradable fault-tolerant systems. Moreover, although not the subject of this paper, the feasibility of such implementations has been demonstrated (see Section III.F.). Future research will seek to accommodate more general reward structures (time-varying rates and "bonuses" associated with state transitions) and more generally defined performance variables, e.g., variables defined with respect to random utilization periods.

- [1] R. Troy, "Dynamic reconfiguration: An algorithm and its efficiency evaluation," in *Proc. 1977 Int. Symp. Fault-Tolerant Computing*, Los Angeles, CA, pp. 44-49, June 1977.
- [2] M. D. Beaudry, "Performance-related reliability measures for computing systems," *IEEE Trans. Comput.*, vol. C-27, pp. 540-547, June 1978.
- [3] F. A. Gay and M. L. Ketelsen, "Performance evaluation for gracefully degrading systems," in *Proc. 1979 Int. Symp. on Fault-Tolerant Computing*, Madison, Wisconsin, pp. 51-58, June 1979.
- [4] H. Mine and K. Hatayama, "Performance related reliability measures for computing systems," in *Proc. 1979 Int. Symp. on Fault-Tolerant Computing*, Madison, WI, pp. 59-62, June 1979.

- [5] T. C. K. Chou and J. A. Abraham, "Performance/availability model of shared resource multiprocessors," *IEEE Trans. Reliability*, vol. R-29, pp. 70-76, April 1980.
- [6] X. Castillo and D. P. Siewiorek, "A performance-reliability model for computing systems," in *Proc. 1980 Int. Symp. Fault-Tolerant Computing*, Kyoto, Japan, pp. 187-192, Oct. 1980.
- [7] J. M. De Souza, "A unified method for the benefit analysis of fault-tolerance," in *Proc. 1980 Int. Symp. on Fault-Tolerant Computing*, Kyoto, Japan, pp. 201-203, Oct. 1980.
- [8] I. Koren and M. Berg, "A module replacement policy for dynamic redundancy fault-tolerant computing systems," in *Proc. 1981 Int. Symp. Fault-Tolerant Computing*, Portland, ME, pp. 90-95, June 1981.
- [9] S. V. Makam and A. Avizienis, "Modeling and analysis of periodically renewed closed fault-tolerant systems," in *Proc. 1981 Int. Symp. Fault-Tolerant Computing*, Portland, ME, pp. 134-141, June 1981.
- [10] J. F. Meyer, "Closed-form solutions of performability," in *Proc. 1981 Int. Symp. on Fault-Tolerant Computing*, Portland, ME, pp. 66-71, June 1981.
- [11] T. Nakagawa, K. Yasui, and S. Osaki, "Optimum maintenance policies for a computer system with restart," in *Proc. 1981 Int. Symp. Fault-Tolerant Computing*, Portland, ME, pp. 148-150, June 1981.
- [12] Y. Oda, Y. Tohma, and K. Furuya, "Reliability and performance evaluation of self-reconfigurable systems with periodic maintenance," in *Proc. 1981 Int. Symp. Fault-Tolerant Computing*, Portland, ME, pp. 142-147, June 1981.
- [13] R. Huslende, "A combined evaluation of performance and reliability for degradable systems," in *ACM/SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, Las Vegas, NV, pp. 157-164, Sept. 1981.
- [14] J. F. Meyer, "Closed-form solutions of performability," *IEEE Trans. Comput.*, vol. C-31, pp. 648-657, July 1982.
- [15] C. M. Krishna and K. G. Shin, "Performance measures for multiprocessor controllers," CRL-TR-1-82, Computing Research Laboratory, The University of Michigan, Ann Arbor, MI, Oct. 1982.
- [16] J. Arlat and J. C. Laprie, "Performance-related dependability evaluation of supercomputer systems," in *Proc. 1983 Int. Symp. on Fault-Tolerant Computing*, Milano, Italy, June, 1983.
- [17] R. Huslende, "Optimal cost/reliability allocation in communication networks," in *Proc. 1983 Int. Symp. on Fault-Tolerant Computing*, Milano, Italy, June, 1983.

- [18] J. A. Munarin , "Dynamic workload model for performance/reliability analysis of gracefully degrading systems ," in *Proc. 1983 Int. Symp. on Fault-Tolerant Computing* , Milano, Italy, pp. 290-295, June, 1983 .
- [19] E. Gai and M. Adams , "Measures of merit for fault-tolerant systems ," in *Proc. Guidance and Control Conf.* , Gatlinburg, TN , Aug. 1983 .
- [20] F. A. Gay, "Performance Modeling for Gracefully Degrading Systems", Ph. D. Dissertation, Northwestern University, Evanston, IL, June 1979.
- [21] J. F. Meyer, "On evaluating the performability of degradable computing systems," in *Proc. 1978 Int. Symp. on Fault-Tolerant Computing*, Toulouse, France, pp. 44-49, June 1978.
- [22] J. F. Meyer, D. G. Furchtgott, and L. T. Wu, "Performability evaluation of the SIFT computer," *IEEE Trans. Comput.*, vol. C-29, pp. 501-509, June 1980.
- [23] J. F. Meyer, "On evaluating the performability of degradable computing systems," *IEEE Trans. Comput.*, vol. C-29, pp. 720-731, Aug. 1980.
- [24] R. A. Howard, *Dynamic Probabilistic Systems, Vol. II: Semi-Markov and Decision Processes*. New York, NY: Wiley, 1971.
- [25] E. Cinlar, *Introduction to Stochastic Processes*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [26] G. N. Cherkosov, "Semi-markovian models of reliability of multichannel systems with unreplenishable reserve of time," *Engineering Cybernetics*, vol. 18, March 1981.
- [27] V. A. Ditkin and A. P. Prudnikov, *Operational Calculus in Two Variables and its Applications*. New York: Pergamon Press, 1962.
- [28] J. F. Meyer and L. T. Wu, "Evaluation of computing systems using functionals of a Markov process," in *Proc. 14th Annual Hawaii Int. Conf. on System Sciences*, Honolulu, HI, pp. 74-83, Jan. 1981.
- [29] L. T. Wu, "Models for evaluating the performability of degradable computing systems", Ph.D. Thesis, University of Michigan, Ann Arbor, MI, 1982.
- [30] ———, "Operational models for the evaluation of degradable computing systems," in *ACM/SIGMETRICS Conf. Measurement and Modeling of Computer Systems*, Seattle, WA, pp. 179-185, Aug. 1982.
- [31] D. G. Furchtgott, "Performability models and solutions", Ph.D. Thesis, University of Michigan, Ann Arbor, MI, 1983.

- [32] H. Kobayashi, *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*. Reading, MA: Addison-Wesely, 1978.
- [33] D. Ferrari, *Computer Systems Performance Evaluation*. Englewood Cliffs, NJ: Prentice-Hall, 1978.
- [34] C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*. Englewood Cliffs, NJ: Prentice-Hall, 1981.
- [35] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [36] H. Raiffa, *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. Reading, MA: Addison Wesley, 1968.
- [37] J. Moses, "Symbolic integration: The stormy decade," *Comm. ACM*, vol. 14, Aug. 1971.
- [38] A. C. Hearn, "REDUCE 2: A system and language for algebraic manipulation," in *Proc. Second Symp. Symbolic and Algebraic Manipulation*.

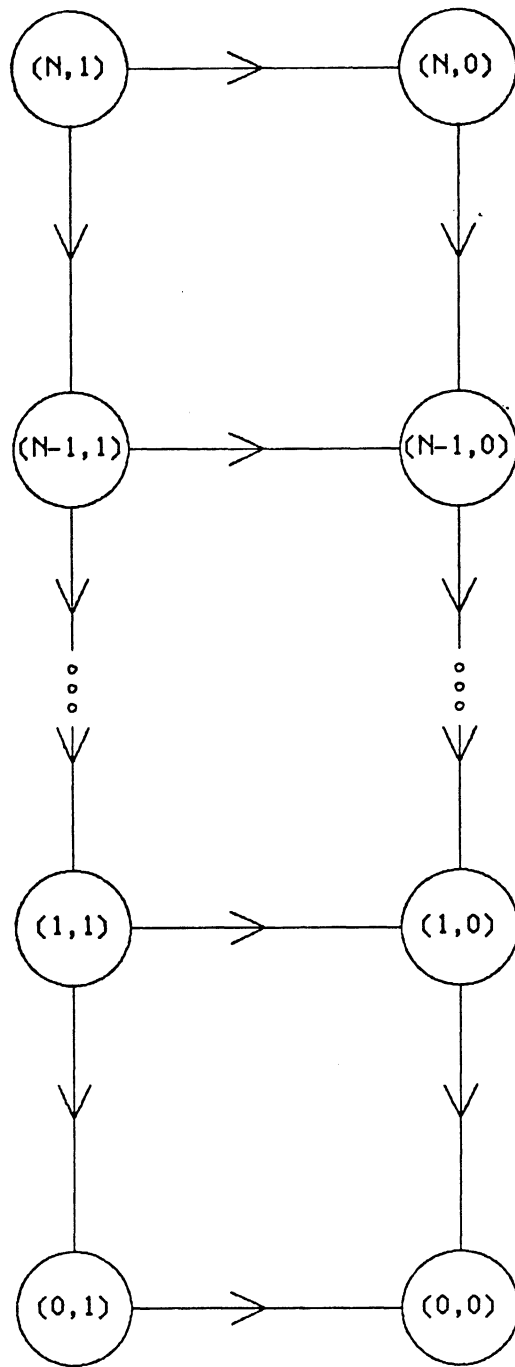


Figure 1.

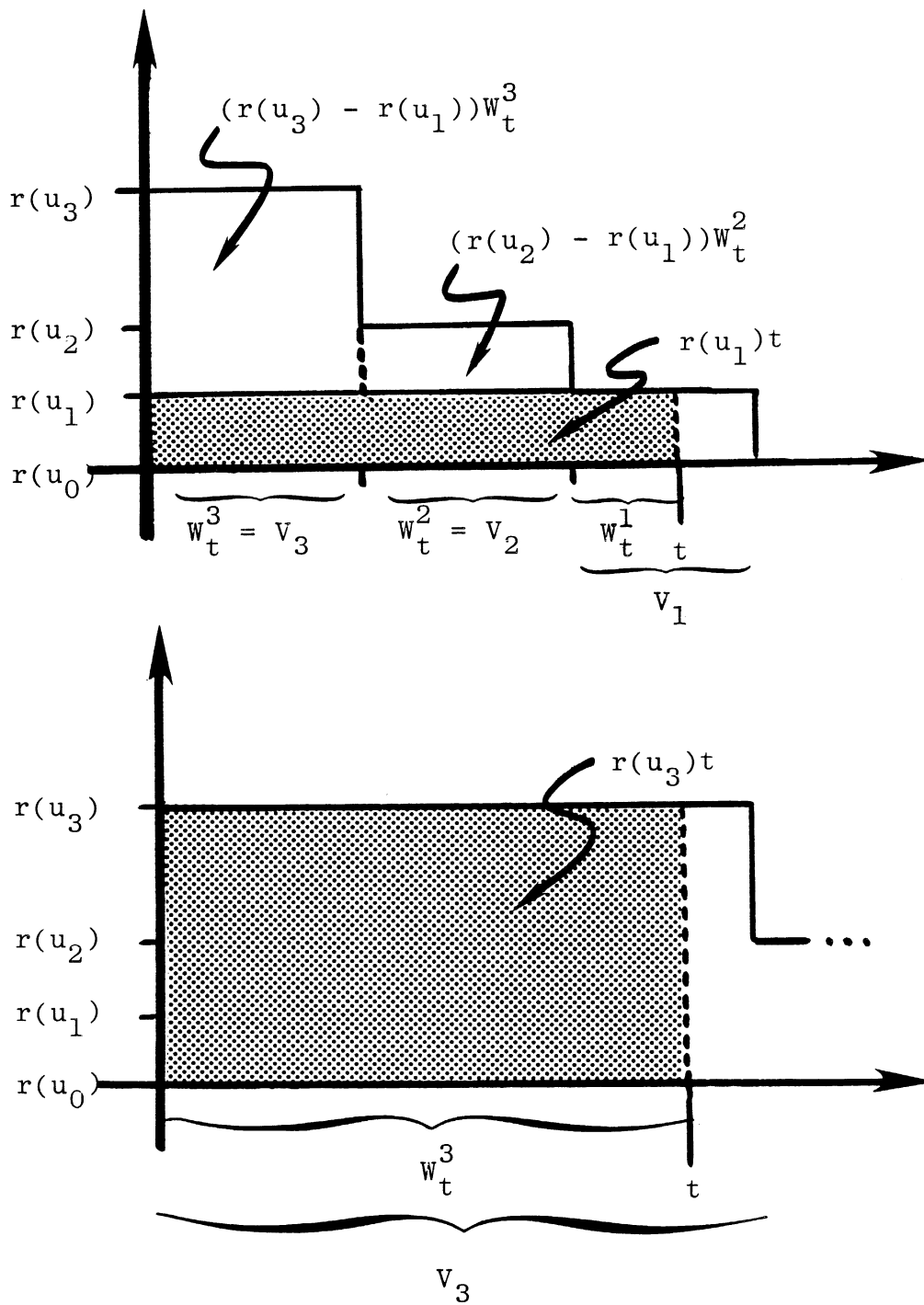


Figure 2.



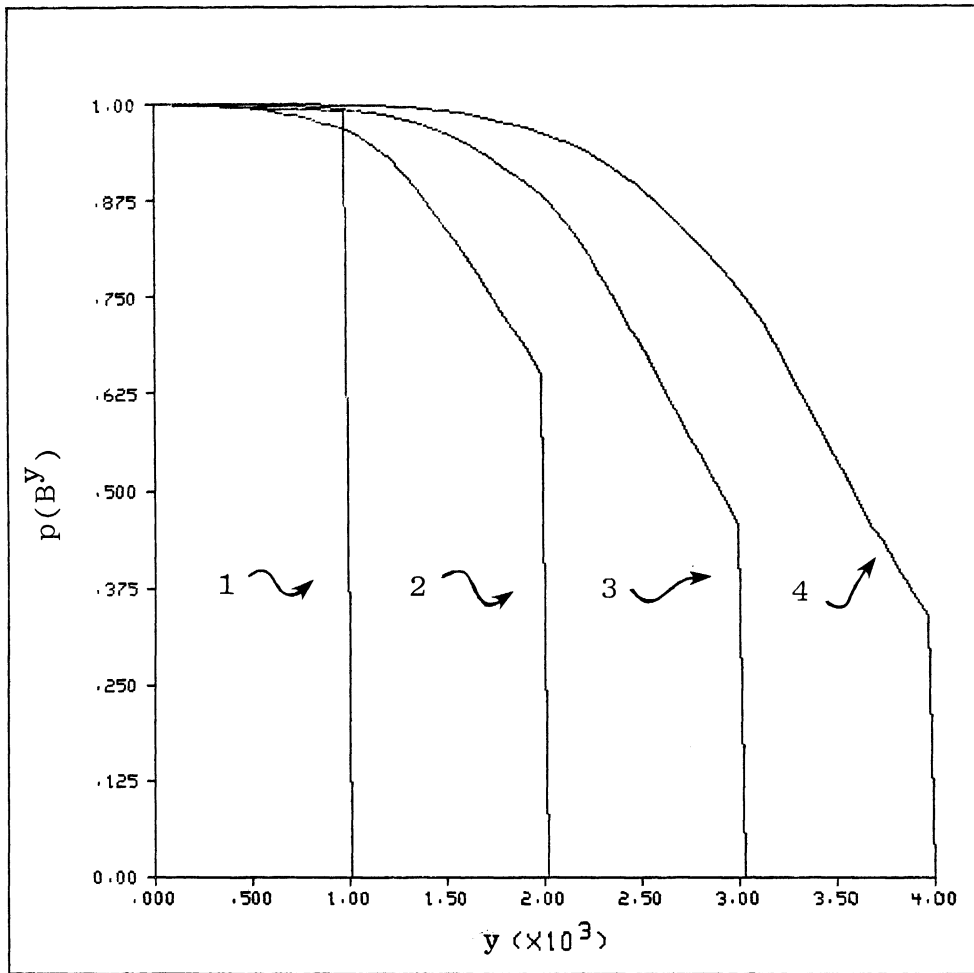


Figure 3.

