IV. Conclusion

We can associate a detection function with each stuck-at fault in a circuit. Functions that bound this detection function have been developed to make the construction practical. The concepts of the Cutting Algorithm can be extended to produce better bounds on signal probabilities and these extensions were applied to an example circuit very successfully (to evaluate the signal probability of the detection functions or their bounding functions). The signal probability range can then be easily converted into a fault detectability range.

REFERENCES


Computing Cumulative Measures of Stiff Markov Chains Using Aggregation

ANDREA BOBBIO AND KISHOR TRIVEDI

Abstract — We present an aggregation method for the computation of transient cumulative measures of large, stiff Markov models. The method is based on the classification of the states of the original problem into slow, fast transient, and fast recurrent states. We aggregate fast transient states and fast recurrent states so that an approximate value to the desired cumulative measure can be obtained by solving a nonsplit set of linear differential equations defined over a reduced subset of slow states only. Several examples are included to illustrate how stiffness arises naturally in actual queuing and reliability models, and to show that cumulative measures provide a better characterization of the time dependent system behavior.

Index Terms — Aggregation, decomposition, Markov chains, queuing networks, reliability models, stiffness, transient analysis.

I. INTRODUCTION

The increasing need for an accurate evaluation of the performance and reliability of highly reliable fault-tolerant systems has led to the definition of a new class of measures that reflect the system's operational characteristics over a finite interval rather than at a fixed time instant or in steady-state. We refer to this new class of measures as cumulative or integral measures [14]. Typical integral measures are the interval availability, which is the fractional system state in a given interval, or the accumulated reward, which is the total amount of work done (or lost) by the system in a given interval. The integral measures are of particular interest in real-time, highly reliable systems since these systems are likely to be evaluated over intervals that are short compared to the mean time to failure. In this paper, we address the problem of evaluating integral measures when the stochastic model assumes the form of a large, stiff Markov chain. Since integral measures are defined over finite time intervals, their evaluation requires the transient analysis of the underlying Markov chain [26].

Stiffness arises naturally in a large variety of stochastic models. A set of linear ordinary differential equations is said to be stiff [24] when its solution contains components that vary rapidly with respect to the length of the interval of integration. In Markov chain models, this condition corresponds to the presence of transition rates that are orders of magnitude larger than the reciprocal of the length of the interval of integration.

In reliability and performability modeling, stiffness is usually generated [3] by the need for analyzing the system behavior in a time interval comparable to the fault-occurrence model, while simultaneously taking into account phenomena that occur in a much shorter time scale, e.g., repair [21], fault-free-operation [23], [12], fault-handling-behavior [11], and so forth. In this paper, we will explore a wide area of possible applications, in which the simultaneous consideration of phenomena that occur in different time scales offers the opportunity of decomposing the system model.

Current research in the transient solution of large, stiff Markov models follows two main lines. The first one is aimed at improving numerical techniques for the transient solution of the Markov chain [24]-[26]. The second approach proceeds by decomposing the original model into smaller submodels. An approximate solution can be obtained by first solving the submodels in isolation (the aggregation step) and then combining the submodel solutions into the solution of the original model (the disaggregation step). Courtois has proposed a decomposition technique for the steady-state analysis of nearly completely-decomposable Markov chains [7]. An alternative method, for the steady-state analysis of general Markov chains, is the iterative aggregation-disaggregation algorithm proposed by Takahashi [32]. Schweitzer [28] discusses an extension of the iterative method to the transient case.

We have proposed a decomposition technique for the transient analysis of stiff Markov chains [3], [2]. A notable property of our decomposition technique is that besides reducing the size of the submodels on which the transient analysis is carried out, it also eliminates the stiffness, making the application of standard numerical integration methods more efficient [25]. Further elaboration of the above tech-
technique has been presented in [27], where the approximation algorithm is motivated in the framework of the perturbation theory of linear systems.

Our technique applies when stiffness arises from the presence of rates belonging to two well separated sets of values in the transition rate matrix of the Markov chain. These rates are accordingly classified into fast and slow rates. States of the Markov chain are also classified into fast and slow states; a state is fast if at least one outgoing transition has a fast rate, otherwise the state is a slow state. The decompositions of the matrix is based on the classification of the fast states, to which stiffness is due, into nearly completely decomposable subsets and into a nearly transient (or unilaterally coupled) subset of states. An appropriate aggregation algorithm is separately applied to each subset so that the final transient approximate solution is obtained by integrating a smaller, nonsiff set of linear differential equations. The paper is aimed to show how the previously proposed aggregation technique [3] can be applied to compute cumulative measures, and to investigate the reduction in the computational effort with respect to standard numerical techniques.

We introduce integral measures in the context of performance and reliability models in Section II. The computation of integral measures for stiff Markov chains, using the proposed decomposition technique, is addressed in Section III. Section IV is devoted to illustrate several examples that show the importance of the time dependent computation of the integral measures, and the capability of the aggregation method.

II. DEFINITIONS AND NOTATION

Matrices are denoted by bold face capital letters (e.g., $A$) and the matrix elements by the corresponding lower case letters (e.g., $a_{ij}$); $I$ is the identity matrix. Vectors are underlined (e.g., $P(t)$). Vector $I$ is the vector whose entries are all equal to one. All vectors are intended to be column vectors, and superscript $T$ denotes transposition. Matrices (vectors) with a subscript identify the partition of the matrix (vector) over the subset identified by the subscript.

Let $\{x(t), t \geq 0\}$ denote a continuous-time discrete-state homogeneous Markov chain defined over a state space $\Omega$ of cardinality $n$ and let $A$ denote its time independent transition rate matrix. The state probability vector $P(t)$ is obtained by solving the standard Markov equation

$$\dot{P}(t) = AP(t)$$  \hspace{1cm} (1)

under a given initial condition $Q = P(0)$. The normalization condition $\int P(t) = 1$ is satisfied for all time $t \geq 0$.

The system of differential equations (1) above can be solved using uniformization [16], [17] or numerical integration [20], [24], [25]. Furthermore, in the implementation of these methods, we may either use full matrix storage or take advantage of the sparsity. Let the number of nonzero entries in matrix $A$ be denoted by $e$ and let $a = \max_{i,j \in \Omega} \{a_{ij}\}$. In the case of full matrix implementation, the computation time of the above methods is known to be $O(n^2e)$ while with a sparse matrix implementation, the computation time reduces to $O(eat)$ [25].

Define the vector $B(t)$:

$$B(t) = \int_0^t P(z) \, dz$$  \hspace{1cm} (2)

whose $i$th entry $b_i(t)$ is the expected total time spent by $y(t)$ in state $i$ during the interval $(0, t)$. This can be seen by defining an indicator random variable $x_i(t)$ so that $x_i(t) = 1$ if $y(t) = i$ and $x_i(t) = 0$ otherwise [1]. Then $\int_0^t x_i(z) \, dz$ is the random variable denoting the total time spent by $y(t)$ in state $i$ during $(0, t)$. Therefore,

$$E \left[ \int_0^t x_i(z) \, dz \right] = \left\{ \int_0^t x_i(z) \, dz : x_i(z) = 1 \right\} = b_i(t).$$  \hspace{1cm} (3)

By direct integration of (1), (2) takes the form

$$\dot{B}(t) = AB(t) + Q$$  \hspace{1cm} (4)

with initial condition $B(0) = 0$.

Often we are interested in the time-averaged problem: let us define the vector $\bar{B}(t) = (1/t)B(t)$. This implies from (4),

$$\bar{B}(t) = (A - 1/t)\bar{B}(t) + Q/t.$$  \hspace{1cm} (5)

With $D(0) = 0$. From the normalization condition on $P(t)$ and from (2) we obtain the following normalization condition for $B(t)$ and $D(t)$, respectively:

$$\int_0^t B(t) = t \hspace{1cm} \text{and} \hspace{1cm} \int_0^t D(t) = 1,$$  \hspace{1cm} (6)

Equation (6) simply states that the sum of the expected times spent over all states adds up to the length of the interval of observation. Equation (5) has a singularity at $t = 0$; however, as $t$ increases, solving (5) is numerically more convenient than (4), because of the following asymptotic property [5]:

$$\lim_{t \to \infty} d_i(t) = \lim_{t \to \infty} \frac{1}{t} b_i(t) = \lim_{t \to \infty} p_i(t) = x_i$$  \hspace{1cm} (7)

$\forall i \in \Omega$ being the steady-state probability vector of $y(t)$.

The systems of linear differential equations (4) and (5) can be solved for $B(t)$ and $D(t)$ by the same methods used for solving (1) [26]. The complexity of computation is once again $O(n^2e)$ with the full matrix implementation, while the computation time with a sparse matrix implementation reduces to $O(eat)$.

We further introduce a reward model superimposed on the stocastic process. Let $R$ be the reward vector whose entry $r_i$ is the reward associated with state $i$. Following [18], we assume

$$r_i = c_i + \sum_{j=1 \neq i}^n h_{ij} a_{ij},$$  \hspace{1cm} (8)

The time-averaged expected accumulated reward is defined by

$$\chi(t) = B^T(t).$$  \hspace{1cm} (9)

Several significant measures can be derived from (9) by particularizing the reward structure.

1) Time-Averaged Expected Number of Transitions $\xi_j(t)$: Let $\xi_j(t)$ denote the time-averaged expected number [with $A$ and $B$] of transitions from state $i$ to state $j$ during the interval $(0, t)$. Its value can be obtained from (9), by assuming $c_i = 0$, $h_{ij} = 1$, and $h_{ki} = 0$, $\forall k \neq j$. In this case,

$$\xi_j(t) = -d_j(t).$$  \hspace{1cm} (10)

If the transition from $i$ to $j$ represents the occurrence of some event in the physical model of the system, $\xi_j(t)$ will give us the time-averaged number of times that event will occur in the interval $(0, t)$.

2) Interval Availability $\phi(t)$: We partition the state space into two mutually exclusive and exhaustive subsets $\Omega_u$ and $\Omega_f$ (with cardinalities $u$ and $f$, respectively, with $u + f = n$) such that $\Omega_u$ is the subset of the system up states and $\Omega_f$ is the subset of the system failed states. By setting in (9),

$$c_i = 1 \hspace{1cm} \forall i \in \Omega_u$$
$$c_i = 0 \hspace{1cm} \forall i \in \Omega_f$$
$$h_{ij} = 0 \hspace{1cm} \forall i, j \in \Omega_u$$

we obtain the interval availability $\phi(t)$, i.e., the expected total fractional up time in the interval $(0, t)$:

$$\phi(t) = I^T \bar{D}(t)$$  \hspace{1cm} (11)

where $u$ is a $u$-dimensional vector whose entries are all equal to one.
and $D_{e}(t)$ is the partition of $D(t)$ over the subset $\Omega_{e}$. Equation (6) implies that $D(t)$ is non-negative.

3) Mean First Passage Time $m_{a}$: For the case in which states in $\Omega$ are absorbing, or are made absorbing [4], the mean first passage time $m_{a}$ from subset $\Omega_{a}$ to subset $\Omega_{e}$ is given by

$$m_{a} = \lim_{t \to +\infty} \int_{\Omega_{a}} D(t) \, dt.$$  

Let $A_{aa}$ and $Q_{e}$ denote partitions of $A$ and $Q$ over $\Omega_{a}$, respectively. Noting that $\Omega_{a}$ is a set of absorbing states, the partitioned form of (4) is

$$B(t) = A_{aa}B(t) + Q_{e}.$$  

Taking the limit as $t$ approaches infinity, we get

$$m_{a} = \int_{\Omega_{a}} (A_{aa} - A_{aa}A_{aa}^{\top}A_{aa} - I_{a})^{-1}Q_{e}.$$  

When $\Omega_{a}$ is the subset of the system down states, $m_{a}$ coincides with the usual definition of MTTF (mean time to failure) [4].

III. Computing Integral Measures for Stiff Markov Chains

We will assume that matrix $A$ is separably stiff. By this we mean that, with respect to the time span $\tau$ for which the transient analysis is to be performed, the matrix contains rates of the order of magnitude of $1/\tau$ (the so-called slow rates), and rates of the order of magnitude of $a/\tau$ with $a > 1$ (the fast rates). The parameter $a$ is the degree of stiffness of the matrix. The above implies that $a = \max_{i,j \in \Omega_{a}} |a_{ij}|$ is of the order of $a/\tau$ and therefore $a \tau$ is very large. Since the run time of both uniformization and explicit numerical integration depends upon $a \tau$ [25], stiffness causes the above methods to be inefficient.

Following the approach presented in [1], we classify states of the Markov chain into fast states and slow states. A state is fast if at least one of the outgoing transitions is fast; otherwise, the state is slow. Next, we generate the subgraph of the fast connections, by deleting all the slow rates in the original Markov chain. States in the subgraph are further classified into a nearly-transient subset and into nearly-recurrent subsets. The nearly-transient subset groups the fast states which are transient in the subgraph of the fast connections; these are the fast states that in the original chain were connected to slow states by means of fast transitions. On the other hand, the nearly-recurrent subsets group fast states which form ergodic subchains in the reduced graph of the fast connections. In the original graph, each nearly recurrent subset is connected to the slow states or to the other nearly recurrent subsets by means of slow transitions only.

A. Fast Transient Subset

In this paragraph, we suppose that the fast states are only of the transient type. Consequently, the state space $\Omega_{a}$ of the original Markov chain is partitioned into a subset $\Omega_{a}$ (of cardinality $a$) containing only slow states and a subset $\Omega_{e}$ (of cardinality $\nu$, $\sigma + \nu = n$) containing the nearly-transient subset of fast states. The transition rate matrix is partitioned in the following form:

$$A = \begin{bmatrix} A_{aa} & A_{ae} \\ A_{ea} & A_{ee} \end{bmatrix}$$  

where by hypothesis $A_{ea}$ and $A_{ee}$ contain only slow entries; $A_{ea}$ contains at least one fast entry and all the diagonal entries of $A_{ee}$ have absolute values of the order of the fast rates. Subset $\Omega_{a}$ is called a fast transient subset since the presence of fast rates in matrix $A_{aa}$ assures that exit from subset $\Omega_{a}$ (once entered) occurs in a time $O(\tau/\alpha)$. Equation (5) becomes, in partitioned form,

$$\begin{align*}
D_{a} &= (A_{aa} - I_{a})^{-1}D_{e} + A_{aa}D_{a} + Q_{e} \left/ \int_{\Omega_{a}} \right. \\
D_{e} &= A_{ae}D_{a} + (A_{ee} - I_{e})^{-1}D_{e} \left/ \int_{\Omega_{e}} \right. .
\end{align*}$$  

An assumption common to our work [2], [3], and other authors [10], [27], is to capture the effect of the nearly transient fast states, by approximating (17) in the limit $\alpha \to \infty$, so that at the integration step $t = \tau$, the fast transient states have already reached their steady-state value. The above assumption implies that $D_{a}(t) = 0$ by (7), and that $I_{e} = \int_{\Omega_{e}}$ is negligible with respect to $A_{ee}$ in (17). Calling $D_{a}(t)$ and $D_{e}(t)$ the approximate values of $D_{a}(t)$ and $D_{e}(t)$, respectively, when $\alpha \to \infty$, we obtain from (16) and (17)

$$\begin{align*}
D_{a}(t) &= (A_{aa} - A_{aa}A_{aa}^{-1}A_{ee} - I_{a})^{-1}D_{e}(t) \\
&+ (Q_{e}D_{e} - A_{aa}A_{aa}^{-1}Q_{e})/t \\
&\quad + (Q_{e}D_{e} - A_{aa}^{-1}A_{ee}Q_{e})/t .
\end{align*}$$

Equation (19) is a set of algebraic equations, while (18) is the set of differential equations representing the averaged integral probabilities up to time $t$ of a Markov chain defined over the state space $\Omega_{a}$, with transition rate matrix $A_{a}$ and initial probability vector $Q_{e}$ given by, respectively,

$$\begin{align*}
A_{a} &= A_{aa} - A_{aa}A_{aa}^{\top}A_{aa} \\
Q_{e} &= Q_{e} - A_{aa}^{\top}Q_{e}.
\end{align*}$$

Matrix $A_{a}$ is sometimes referred to as the Schur complement of $A_{ee}$ in $A$ [6], [13] and has the important property that it coincides with the $a$-dimensional matrix obtained by applying to matrix $A_{ee}$, $a$ steps of a standard Gaussian-elimination algorithm [13]. Exploitation of this property is quite useful in the numerical implementation of the aggregation technique.

A probabilistic argument leading to (20) and developed in [3] and [15] shows that a generic entry $a_{ij}$ of the Schur complement $A_{a}$ (with $i, j \in \Omega_{a}$) is the sum of the direct transition rate from state $j$ to state $i$ in the original matrix $A$, plus the rate obtained by accounting for all the possible paths connecting state $j$ to state $i$ through states in $\Omega_{e}$.

Since matrix $A_{a}$ of (20) is a transition rate matrix of a Markov chain, it follows from (6) that

$$\int_{\Omega_{a}} D_{a}(t) = 1 \quad \text{and} \quad \int_{\Omega_{a}} D_{a}(t) + \int_{\Omega_{a}} D_{e}(t) = s \geq 1 .$$

Thus, we define a new approximate averaged integral vector $D^{\ast}(t)$ by normalizing the entries of $D(t)$ with respect to the normalization factor $s$. The final approximation is defined in the following equation:

$$\begin{align*}
D^{\ast}(t) &= \frac{1}{s} \left[ \begin{array}{c} D_{a}(t) \\ D_{e}(t) \end{array} \right] \quad \text{with} \quad \int D^{\ast}(t) = 1 .
\end{align*}$$

The computational complexity of the aggregation algorithm when a fast transient subset is present can be estimated by the following argument. The approximate transient analysis is accomplished by numerically integrating the reduced set of $a$ equations (18), uniformization and explicit numerical methods provide a run time of the order of $O(\epsilon^{d+1}/a)$. If full matrix storage is used, aggregation reduces the run time with respect to the original problem by a factor of the order of $O((n/\epsilon)^{2})$ which is <1. Solution of (18) requires the partition of the original set of states $\Omega_{a}$ into the subsets $\Omega_{a}$ and $\Omega_{e}$, and the calculation of the Schur complement $A_{a}$ (20). Calculating the Schur complement by means of Gaussian elimination implies a complexity $O(\epsilon^{d})$ [31].

B. Fast Recurrent Subsets

In this section, we introduce the decomposition algorithm in the case in which the state space $\Omega_{a}$ of the Markov chain is partitioned into a subset $\Omega_{a}$ of slow states and into a single nearly recurrent subset of fast states $\Omega_{e}$ ($\sigma + \rho = n$). The generalization to the case where several recurrent subsets are present together with a fast transient subset can be obtained in a straightforward manner, following the pattern presented in [3] for the computation of the instantaneous probabilities.
The transition rate matrix is partitioned as follows:

\[
A = \begin{bmatrix}
A_{ss} & A_{sp} \\
A_{ps} & A_{pp}
\end{bmatrix}
\]  

(24)

where the matrix \(A_{ss}\) contains at least one fast rate in each column, and all the other matrices contain only slow rates. The matrix \(A_{sp}\) is nearly-composite [7], since states inside subset \(p\) are, by definition, strongly connected through fast rates, and are only weakly connected to states outside \(p\). The fast recurrent subset thus forms a nearly ergodic Markov chain.

The transient analysis, in this case, is based on the classical theorems of Simon and Ando discussed in [7]. The theorems state that the system dynamics can be separated in two phases: a short-term phase and a long-term phase. During the short-term phase, the attainment of equilibrium inside each nearly completely decomposable subset is weakly affected by the interactions among aggregates. In the long-term phase, the relative values attained in the short-term equilibrium are approximately maintained, and the system dynamics is dominated by the interaction among the aggregates.

For the case of separably-stiff Markov chains, the equilibrium inside a fast recurrent aggregate occurs in a time which is \(\alpha\) orders of magnitude shorter than the time to reach equilibrium in the slow states. Since our main concern is the transient analysis, we can assume that in a time span \(\tau\) of the order of the reciprocal of the slow rates, the fast recurrent subsets have already reached their short-term equilibrium.

Approximate values to the short-term equilibrium probabilities can be calculated by considering the fast recurrent subset in isolation, and converting it into an ergodic Markov chain. To this end the slow rates directed toward states outside the fast recurrent subset must be redistributed into the states of the subset. The way in which this redistribution is done, is that in the way in which the fast recurrent subset is made ergodic, affects the final level of accuracy of the algorithm [8].

In the literature, the ergodicity inside the fast recurrent aggregate is usually obtained [3], [27] in the limit \(\alpha \to \infty\), i.e., by neglecting the slow rates outgoing from the fast recurrent states. However, different and better strategies can be envisaged [8]. A strategy based on the conversion of the fast recurrent subset into an ergodic chain whose transition rate matrix is given by the Schur complement of \(A_{ss}\) in \(A\) would provide asymptotically exact results [34]. In this latter case, the steady-state solution would require the same amount of computation [22] as would be required by the straightforward solution of the undecomposed problem. However, stiffness is avoided in the transient analysis, thus making easier the implementation of the integration routine from the point of view of the computation time and memory requirements.

Let \(A_{ss}\) denote the ergodic Markov transition rate matrix obtained by suitably modifying \(A_{ss}\) (for instance, neglecting the slow rates), and let \(X_s\) denote its equilibrium probability vector. The effect of the fast recurrent subset can be condensed into a single slow state, whose exit rates toward the other slow states are computed by weighting the original rates with the steady-state probabilities of the aggregate. By this construction, the original Markov process can be reduced to a new Markov process containing \((\alpha + 1)\) slow states and whose transition rate matrix is given by

\[
A^* = \begin{bmatrix}
A_{ss} & A_{sp}X_s \\
\bar{\Lambda}_p & A_{pp}
\end{bmatrix}
\]  

(25)

with \(\bar{\Lambda}_p = \sum_p A_{ps}X_s\). Equations (4) and (5) are now solved using the matrix (25) to give approximate values for the slow states integral probabilities \(D_s^*(t)\) and a value \(d(t)\) for the slow state representing the recurrent aggregate. An approximate value for the integral measures for the states inside the fast aggregate is obtained by the following disaggregation step:

\[
D_s^*(t) = X_s d(t).
\]  

(26)

Subsequently, we can obtain the needed integral measures as in Section II. By an argument similar to the one developed for the

fast transient subset, we can show that the run time of the transient solution of the Markov chain of (25) is \(O(\tau + 1)^2\alpha^{-1}\). The reduction of the computation complexity of the numerical integration for the aggregated chain with respect to the complete algorithm is \(O((\alpha + 1)^3/n^2)\). Equation (25) requires the preliminary partition of the original state space into the fast transient and fast recurrent subsets, and the evaluation of the steady-state probability vector \(X_s\), whose computation complexity is \(O(n^2)\) in the full matrix implementation and \(O(\alpha n)\) in the sparse matrix case (where \(n\) is the number of nonzero entries in matrix \(A_{ss}\)).

IV. EXAMPLES

The following examples are meant to illustrate interesting aspects of the aggregation algorithm. They should also suggest further possible areas of application.

A. M/M/1/k Queueing System with Delay and Transmission Errors

The main results in queueing system analysis [19] are obtained under the assumption that there is no delay between successive services, whenever the queue is not empty. A further assumption is that a customer in line reaches the service station with probability one provided the server is idle. Both assumptions characterize an instantaneous and failure-free communication link between the buffer and the service facility. However, in many real systems, the physical nature of the link introduces a nonzero delay which possibly affects all the standard performance measures of the queueing system.

A number of examples can be drawn from telecommunication systems or distributed computer systems, where the transmission of the tasks from one processing center to another may really affect the system throughput. However, an area in which this problem can be particularly severe is the one of VLSI circuits, where the circuit speeds are likely to be dominated by the delay due to the wiring capacitance [29].

A second level of detail that can be considered when modeling a physical communication link is the possibility of occurrence of a failure during the transmission, that is the possibility of losing a job in the passage from the buffer to the service facility. In fact, most of the faults in computer systems are due to faults in the transmission phase rather than in the processing phase [30].

As a typical performance measure, we will compute the time-averaged expected throughput in the interval \((0, t)\). In particular, we will examine the effect of the delay and of the failure of the communication link in reducing the performance level of the queueing system.

1) Exponentially Distributed Transfer Delay: Let \(\lambda\) denote the constant arrival rate and let \(\mu\) the constant service rate. We further assume that the system has a buffer with limited capacity equal to \(k\) and that the transfer delay from the line to the service station is exponentially distributed with rate \(\delta\). The resulting Markov chain is shown in Fig. 1. The first component in each state indicates the
number of customers in the line while the second component indicates
the server status (0 = idle; 1 = busy). We further assume as initial
condition that the queueing system is empty (q0 = 0 = i = 1).
If δ is a fast rate with respect to both λ and ρ (δ ≫ λ, ρ), states
{(i, 0), i ≥ 1} are transient fast states since the state (i, 0)
is connected by means of the fast rate δ to the corresponding slow state
(i − 1, 1). By rearranging slow and fast states as in (15), the transition
rate matrix for the case k = 2 takes the form
\[
A = \begin{bmatrix}
-λ & μ & 0 & 0 & 0 & 0 \\
0 & -(λ + μ) & 0 & 0 & δ & 0 \\
0 & λ & -(λ + μ) & 0 & 0 & δ \\
0 & 0 & λ & -μ & 0 & 0 \\
λ & 0 & μ & 0 & -(λ + δ) & 0 \\
0 & 0 & 0 & μ & λ & -δ
\end{bmatrix}.
(27)
\]
By assuming that fast states have reached equilibrium, we transform
matrix (27) into matrix (28) by means of (20):
\[
A_s = \begin{bmatrix}
-λ & μ & 0 & 0 & 0 & 0 \\
δβ & -(λ + μ) & μ(1 − β) & 0 & 0 & 0 \\
λβ & λ & -(λ + μ(1 − β)) & μ & 0 & 0 \\
0 & 0 & λ & -μ & 0 & 0 \\
0 & 0 & 0 & μ & λ & -δ
\end{bmatrix}.
(28)
\]
Here, β = λ/(λ + δ) is the steady-state probability that a new cus-
tomer arrives while the server is passing from the buffer to the service
facility. The aggregated chain, obtained by eliminating the fast transient states, is shown in Fig. 2. Note that the aggregated
chain is no longer stiff since it does not contain rates of the order of δ.

Since the system has a finite capacity, we assume that an arriving
job that finds the system full is rejected. The time-averaged expected
throughput in the interval (0, t), \(1/τ\mathbb{E}[W(t)]\), has the form
\[
\frac{1}{τ} \mathbb{E}[W(t)] = \sum_{i=0}^{k} x_i(0, i) μ = \sum_{i=0}^{k} d^*_s(0, 1, i). 
(29)
\]
The expected number of incoming jobs being rejected because the buffer is full in the interval (0, t) is given by the following expression:
\[
λ(δ τ_s^*_s(0, 1) + δ τ_s^*_s(1, 0)). 
(30)
\]
\[
A_s = \begin{bmatrix}
-λ & μ & 0 & 0 & 0 & 0 \\
\mu δ τ_s^* & -(λ + μ) & μ(1 − v) & 0 & 0 & 0 \\
λ δ τ_s^* & λ & -(λ + μ(1 − v)) & μ & 0 & 0 \\
0 & 0 & λ & -μ & 0 & 0 \\
0 & 0 & 0 & μ & λ & -δ
\end{bmatrix}
(32)
\]
where \(τ_s = 1/(δ + λv)\).

Fig. 2. The aggregated Markov chain obtained from the queueing system of Fig. 1 with k = 2, by eliminating the fast transient states.

2) Transmission Errors: A further step toward the modeling of physical communication links is to allow the possibility of a loss of a message (job) during transmission. Let \(v\) denote the conditional transmission probability, defined to be the probability that a message reaches the server given that the link is not empty and the server is idle. In this model, (1 − v) accounts for phenomena such as: hardware failures on the communication link, disturbances on the channel, and so forth. The Markov transition graph for this system is shown in Fig. 4. The transition rate matrix for the particular case k = 2 takes the following form:
\[
A = \begin{bmatrix}
-λ & μ & 0 & 0 & 0 & 0 \\
0 & -(λ + μ) & 0 & 0 & vδ & 0 \\
0 & λ & -(λ + μ) & 0 & 0 & vδ \\
0 & 0 & λ & -μ & 0 & 0 \\
λ & 0 & μ & 0 & -(λ + δ) & (1 − v)δ \\
0 & 0 & 0 & μ & λ & -δ
\end{bmatrix}
(31)
\]
By assuming, as in the previous example, that δ ≫ λ, ρ, the states
{(i, 0), i ≥ 1} are fast transient states. Their elimination, by means of
the aggregation step, produces the following transformed Markov
chain defined over the subset \(Ω_σ\):

In Fig. 3 we report \(1/τ\mathbb{E}[W(t)]\) as a function of t for a queueing system with \(λ = μ = 1\). The cumulative index is plotted for two values of the delay rate (δ = 10, δ = 1000) and for two different values of the buffer capacity (k = 2, k = 100). The corresponding steady-state values (t → ∞) are also reported for comparison. The graphs show that the time-averaged expected throughput increases as a function of time and that the maximum value is reached in steady state. The system performance is reduced by increasing the

delay or by decreasing the buffer size. However, we observe that for shorter utilization times, the effect of the delay and the effect of the limited buffer size is less significant, suggesting the application of appropriate strategies for the optimum system design.

In Fig. 5, \(1/τ\mathbb{E}[W(t)]\) is plotted as a function of t for a system with \(λ = μ = 1\), delay rate δ = 1000, and two different values of the buffer capacity (k = 2, k = 100). The effect of the conditional error probability on the message trans-
Fig. 3. Time-averaged expected throughput in (0, ε), as a function of time and of the delay rate. The steady-state values are reported on the right.

Fig. 4. Markov transition graph of an M/M/1/k queueing system with exponentially distributed delay and conditional transmission probability v.

Fig. 5. Time-averaged expected throughput in (0, ε), as a function of time and of the transmission probability v for a delay rate δ = 1000. The steady-state values are reported on the right.

mission is examined by comparing the performance index computed for different values of v (v = 0.8, v = 0.9, v = 0.99). The ideal case (v = 1) is inferred from the corresponding curves of Fig. 3. The steady-state values (t → ∞) are reported on the right. Decreasing the conditional failure probability v reduces the overall system performance, as expected.

The idea of modeling the physical process of transmitting a message from the line to the server can be further elaborated by inserting more details in the model. For example, complex self-checking and recovery strategies might be considered. When the rates of the recovery model are fast with respect to both the arrival and the service rates, the proposed aggregation technique allows us to separately perform computations on a group of fast states and on a group of slow states with a tolerable increase in the amount of overall computations.

B. A Queueing Network Model of an Interactive System

Consider a queueing network model of an interactive system [33] (Fig. 6), consisting of n_t terminal users, each submitting a request at a rate λ whenever in think state. The computer subsystem consists of a CPU and an I/O device with service rates µ_u and µ_t, respectively.

When a request is issued by a terminal user, it may have to wait in the memory queue. Admittance of the user requests to the CPU–I/O subsystem is scheduled on an FCFS basis with the total number of concurrent requests in the subsystem being limited to a value k. An admitted request joins the CPU queue. Both CPU and I/O queues are scheduled on an FCFS basis. Upon completion of a CPU burst, the terminal request is completed with probability u_0. With probability u_1 (1 − u_0), the terminal command needs to access the I/O device. Upon completion of an I/O request, the command rejoins the CPU queue.

Fig. 7(a) shows the Markov transition graph in the case in which the system has n_t = 5 terminals, one I/O device, and the upper bound on the CPU–I/O subsystem population is k = 2. States of the Markov chain are triples (i, j, l), where i is the number of command requests at the CPU, j is the number of command requests at the I/O server, and l is the number of command requests in the memory queue. We further define m = l + j + l.
We assume that the probability of completing a terminal request is low with respect to the probability of joining the I/O queue (\( v_0 \ll v_1 \)). Furthermore, the rate \( \lambda \) at which terminals submit requests to the CPU queue is slow compared to both the CPU and I/O service rates (\( \lambda \ll \mu_0, \mu_1 \)). Hence, from the point of view of the aggregation algorithm, we can assume \( \lambda \) and \( v_0 \mu_0 \) to be slow rates and \( \lambda \mu_0 \) and \( \mu_1 \) to be fast. With the above classification of fast and slow rates, the root subchains of Fig. 7(a), characterized by a fixed value of \( m \) (\( 0 \leq m \leq 5 \)), form fast recurrent subsets of states. Notice that, with \( m = 1 \), the fast recurrent subset has 2 states and, with \( m = 0 \), the fast recurrent subset reduces to the single slow state (0, 0, 0).

Each fast recurrent subset, considered in isolation from the rest of the system, forms an \( M/M/1/k \) queue, with \( k' = m \) if \( m < k \), and \( k' = k \) otherwise. It follows that the steady-state probability vector \( \tilde{\pi} \) of each fast recurrent subset has the following entries [33]:

\[
x_{m,j} = \frac{\gamma^j}{1 - \gamma} \frac{1 - \delta}{1 - \delta^k + 1}
\]  

where \( \gamma = \frac{\lambda \mu_0 v_1}{\mu_1} \) is the traffic intensity and \( j \) is the number of requests in the I/O queue.

By applying the aggregation technique described in Section III-C, each fast recurrent subset is aggregated into a single slow state. Thus, the final Markov chain on which the transient analysis is carried out assumes the form shown in Fig. 7(b). In Fig. 7(b), the label inside each state is the value of \( m \), and the transition rates have the following expressions:

\[
\mu_m = (X_{m,0} + X_{m,1}) \mu_0 v_0
\]

\[
\lambda_m = (5 - m) \lambda
\]

We assume that the system is in state (0, 0, 0) at time 0, with probability one. By choosing the reward rate for state \((i, j, l)\) equal to \( r_{(i,j,l)} = l \), the time-averaged expected accumulated reward \( x(t) \) defined in (9), represents the time-averaged expected number of terminal requests waiting to enter the CPU-I/O subsystem. This quantity is plotted in Fig. 8 as a function of time, assuming \( \lambda = 1/20, \mu_0 = \mu_1 = 100, \) and \( v_0 \) varying from 0.01 to 0.1.

The accuracy of the aggregation method is illustrated in Table I, as a function of the dimensionless quantity \( \mu_0 t \). Table I shows the relative error incurred by computing the performance index \( x(t) \) for the Markov chain of Fig. 7(a) and the aggregated chain of Fig. 7(b). The approximation error decreases with \( v_0 \), or as the degree of stiffness increases. Note that the asymptotic error shown in the table is a consequence of the approximate evaluation of the local equilibrium probability, calculated from (33), by considering each nearly recurrent fast subset in isolation.

V. CONCLUSION

We have extended our earlier work on aggregation technique for the transient analysis of stiff Markov chains to the computation of their integral measures. When the transition rate matrix is separably stiff, the state space of the Markov chain can be decomposed into a subset of slow states, a subset of fast transient states, and subsets of fast recurrent states; an approximate solution to the original problem is obtained by solving smaller stiff problems inside each subset.

In particular, the paper addresses the transient analysis, and shows that, when the integration time is of the order of the reciprocal of the slow rates, the influence of fast rates on the dynamics of the slow states can be captured by solving a steady-state problem inside each fast subset. The transient numerical integration is thus accomplished on a reduced set of equations defined over the slow subset only. The run time of the numerical integration for the aggregated chain is reduced, with respect to the original problem, by a factor which depends on the number of slow states and on the degree of stiffness. Furthermore, as the degree of stiffness increases, both the accuracy of the algorithm and the numerical efficiency also increase.

We have presented several fully elaborated numerical examples. The aim of the examples is twofold. First, we want to show how time dependent cumulative measures provide a better characterization of the system behavior, and how these measures can be computed from the vector of the time-averaged expected total time spent in each Markov state during the interval of observation. Second, the examples show how stiffness arises in practical applications when we want to include into a single model effects that occur in different time scales. The proposed aggregation procedure has proved to be an effective method for automatically isolating the components which have the largest influence on the system dynamics in each time interval. In the queueing network model of an interactive system, our approach leads to a separate analysis of a slow submodel, consisting of the submission and completion of terminal requests, and that of a fast submodel consisting of accessing and servicing of the CPU and the I/O device. Moreover, aggregation offers the analyst the opportunity of refining the model with a tolerable increase in the model complexity. A typical example of the model refinement is the consideration of a transfer delay time and the probability of a fault in the transfer process in a classical queueing system.

Aggregation coupled with efficient numerical integration techniques provides an effective procedure for tackling large and stiff Markov models.

ACKNOWLEDGMENT

The numerical computations presented in this paper were performed using the package ESP written by A. Cuman at IEN and presented in [9].

REFERENCES


A Characterization of t/s-Diagnosability and Sequential t-Diagnosability in Designs

JOO-KANG LEE and JON T. BUTLER

Abstract—A multiprocessing system is t/s-diagnosable if all faulty processors can be identified to within s processors provided there are no more than t faulty processors. A characterization theorem of Karanasthii and Friedman [4] for t/s-diagnosability in certain special cases of systems called designs is extended to the entire class of D_{t,s}(n) designs. We show that for large t, s is approximately t^2/4s. Furthermore, the minimum number of processors needed to attain a given diagnosability is derived.

A multiprocessor system is sequentially t-diagnosable if at least one faulty processor can be identified provided there are no more than t faulty processors. A theorem by Preparata, Metze, and Chien [7] giving a sufficient condition for sequential t-diagnosability in the single loop system, a special case of designs, is extended to the entire class of D_{t,s}(n) designs. We show that, for large t, approximately t^2/4s nodes are needed for a D_{t,s}(n) design to be sequentially t-diagnosable.

Index Terms—Multiprocessing systems, reliable computing, systems diagnosis, t-diagnosable, t/s-diagnosable, testing.

I. INTRODUCTION

In the systems diagnosis approach to reliable computing, fault location is achieved by tests among processors. We assume that fault-free processors produce test results that are true representation of the tested processor, fail if it is faulty and pass if it is fault-free. In the case of faulty processors, however, the test results by such processors may not be correct. The goal is to determine exactly which processors are faulty. However, if there are too many faulty processors, incorrect test information can cause ambiguity.

Our model is that of Preparata, Metze, and Chien [7]. A system S is a directed graph where nodes represent processors and arcs represent tests among processors. Node u_i tests node u_j iff there is a directed arc from u_i to u_j. Each node has one of two states, faulty or fault-free, and each arc has one of two weights, pass or fail. For example, Fig. 1 shows a system of 12 nodes and two arrangements of three faulty nodes, which are indicated by X’s. Fail test outcomes are indicated by 1’s, while unmarked arcs correspond to pass outcomes.

A system is (one-step) t-diagnosable if all faulty nodes can be uniquely identified provided there are no more than t of them. For example, the system in Fig. 1 is not 3-diagnosable because the set of test outcomes shown in Fig. 1(b), which is produced with u_1, u_4, and u_6 faulty, can also be produced with just u_1 and u_4 faulty. Thus, if we assume there are three or fewer faulty nodes in the system, u_5 cannot be uniquely identified as faulty. t-diagnosability represents worst case conditions. For example, the three faulty nodes in Fig. 1(a) are uniquely faulty.

S is a D_{t,s}(n) design if an arc exists from node u_i to u_j for