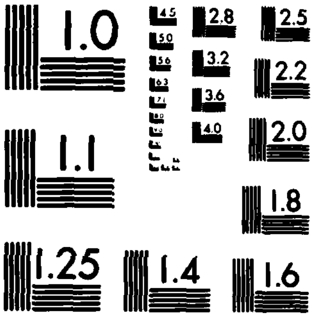
AD-A150 476   AUTOMATIC SYMBOLIC SOLUTION OF MARKOV CHAINS(U) DUKE   1/1
UNIV DURHAM NC DEPT OF COMPUTER SCIENCE R MARIE ET AL.
1984 CS-1984-23 AFOSR-TR-85-0012 AFOSR-84-0132

UNCLASSIFIED   F/G 12/1   NL


1.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

5

CS-1984-23

Automatic Symbolic Solution of Markov Chains

Raymond Marie
I.N.S.A
35031 Rennes Cedex, France

Andrew Reibman and Kishor Trivedi
Department of Computer Science
Duke University
Durham, NC 27706 USA

Approved for public release;
distribution unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
CS-1984-23

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 85-0012

6a. NAME OF PERFORMING ORGANIZATION
Duke University

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Air Force Office of Scientific Research

6c. ADDRESS *(City, State and ZIP Code)*
Dept of Computer Science
Durham NC 27706

7b. ADDRESS *(City, State and ZIP Code)*
Directorate of Mathematical & Information
Sciences, Bolling AFB DC 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
AFOSR

8b. OFFICE SYMBOL *(If applicable)*
NM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR-84-0132

8c. ADDRESS *(City, State and ZIP Code)*
Bolling AFB DC 20332-6448

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A5	

11. TITLE *(Include Security Classification)*
AUTOMATIC SYMBOLIC SOLUTION OF MARKOV CHAINS

12. PERSONAL AUTHOR(S)
Raymond Marie*, Andrew Reibman and Kishor Trivedi

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*
1984

15. PAGE COUNT
20

16. SUPPLEMENTARY NOTATION
*I.N.S.A., 35031 Rennes Cedex, France.

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Continuous time Markov chains are commonly used in system performance modeling. Increasing system complexity and non-Markovian behavior can drastically increase the size of a Markov model's state space. Accordingly, approximation techniques have been introduced to reduce the resources needed to solve Markov chain models. In this paper the authors discuss a method for automatically deriving symbolic solutions of Markov chains. Symbolic solutions should provide insight when attempting to evaluate the validity of both Markov models and approximation techniques for their solution.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
MAJ Brian W. Woodruff

22b. TELEPHONE NUMBER *(Include Area Code)*
(202) 767- 5027

22c. OFFICE SYMBOL
NM

Automatic Symbolic Solution of Markov Chains[*]

Raymond Marie[**]
IRISA
I.N.S.A
35031 Rennes Cedex, France

Andrew Reibman and Kishor Trivedi
Department of Computer Science
Duke University
Durham, NC 27706 USA

Abstract

Continuous time Markov chains are commonly used in system performance modeling. Increasing system complexity and non-Markovian behavior can drastically increase the size of a Markov model's state space. Accordingly, approximation techniques have been introduced to reduce the resources needed to solve Markov chain models. In this paper we discuss a method for automatically deriving symbolic solutions of Markov chains. Symbolic solutions should provide insight when attempting to evaluate the validity of both Markov models and approximation techniques for their solution.

1. Introduction

Continuous time Markov chains (CTMC) are commonly used tools in computer systems modeling. CTMC have been used to model program behavior,[1] system performance,[2,3] system reliability,[4,5] and system availability,[6] and also in the combined evaluation of performance and reliability.[7,8] Although the limitation of exponentially distributed state occupancy times, as implied by a homogeneous CTMC, appears to be restrictive, it is possible to use the Coxian method of stages to allow arbitrary phase type distributions.[3,9,10,11]

In general, once a Markov chain model of a system has been constructed, there are several solution methods available. Figure 1 summarizes these methods and typical modeling packages that employ them. The Markov model of a system can be

[*] This work was supported by the Air Force Office of Scientific Research under grant AFOSR-84-0132 and by the Army Research Office under grant DAAG-84-0045.

[**] This work was begun while Dr. Marie was a visiting scientist in the Department of Computer Science at Duke University.

solved using integral equations, formally taking the convolution of the probability of entering the state with the probability of remaining in it.[12, 13] Alternatively, the Markov chain can be converted to a coupled set of homogeneous differential equations.[3] This set of equations can be solved using either numerical techniques or Laplace transforms. A third solution method is simulation. As an introduction, we will briefly discuss some of the advantages and disadvantages of each of these methods.

Simulation can be more realistic than analytic models. Many types of complex systems, particularly those for which independence assumptions are invalid, can be directly modeled using simulation. However simulation models can have high development costs and may require large amounts of computer time to obtain statistically significant results. Therefore, the low cost alternative of analytic modeling can be attractive, even if approximating assumptions are necessary. If a range of systems must be compared, similar systems must often be simulated individually. Analytic modeling may permit the comparison of such systems without repeated simulation.

The most common analytic approach is to represent the Markov chain as a set of coupled differential equations. Each equation describes the flow "balance" conditions in a corresponding state of the chain, i.e. the instantaneous rate of change in the probability of being in a state is equal to the rate of arrival into the state less the rate of departure from the state at that instant. This set of equations can be solved using either Laplace transforms or numerical techniques. Using Laplace transforms for large systems (either numerically or symbolically) may require finding the roots of many large polynomials, a computationally expensive task. One advantage of numerical techniques is that they can be easily extended to evaluate non-homogeneous Markov chains. However, if the system is stiff, i.e. if two or more transition rates out of any single state differ greatly in magnitude, special care will be needed to get an accurate solution.

Sets of integral equations are similar to coupled differential equations.[12] Integral equations provide a basis for modeling both semi-Markov processes and non-homogeneous Markov chains. Integral equations for Markov chain state probabilities will also provide the basis for the closed form solution techniques discussed later in this paper.

One issue we do not address in detail is the solution of cyclic Markov chains. In the context of modeling fault-tolerant systems, we are restricting our attention to non-repairable systems. Such systems can generally be represented by Markov chains without cycles. If cycles are present in the model, all the solution methods discussed are of diminished utility. The simulation of a cyclic Markov chain may be more expensive (for the same degree of accuracy) than for an acyclic chain of the same size, as the number of possible paths through the chain are no longer finite. If an analytic solution of a cyclic Markov chain is desired, numerical solution of systems of either differential or integral equations is usually the recommended approach. Approximation techniques may still allow us to obtain a symbolic solution, albeit an inexact one.

To illustrate the use of Markov models, we consider an example from reliability modeling.[14] Figure 2 depicts the transition diagram of a Markov chain representing a 3-component parallel redundant system. The individual components have lifetimes that are independent and exponentially distributed with parameter λ. When an individual component fails, a reconfiguration process with rate parameter δ begins. This process is guaranteed to reconfigure the system as long as a second fault does not occur before the reconfiguration is completed. The reliability of the system at time t, denoted $R(t)$, is given by $1 - P\{$process in state F_c or F_e at time $t\}$.

A package employing traditional (i.e. numerical) solution techniques would input numeric values for the parameters of the model (here δ and λ) and would solve the system numerically for t less than some fixed value. To determine the actual behavior of the system as a function of a parameter other than t would require many runs of such a program.

In this paper, we discuss a method for the derivation of state probabilities of an acyclic Markov chain in a symbolic fashion. Closed form results (that previously could be obtained only by hand) give greater insight into actual system behavior by allowing us to easily study the relationship between input parameters and the resulting state probability distributions. Our approach is based on the use of integral equations. It is computationally similar to using Laplace transforms to solve systems of coupled ODE's. The program implementing the algorithm discussed in the paper is called ACE (*Acyclic* Markov *Chain Evaluator*). The solution of cyclic chains (which presents additional difficulties) is omitted from our discussion.

In section 2, we describe our method which is partially inspired by the program SPADE.[15] In section 3 we describe our program's implementation. Some examples of the use of ACE are given in section 4.

2. Basic Approach

Consider an acyclic continuous time Markov Chain. Let the states be numbered 1,2,...,N. Let $\lambda_1, \lambda_2, \ldots, \lambda_M$ be the transition rate variables. The transition rate from state i to state j is denoted by q_{ij}, where q_{ij} can be expressed as a linear sum of transition rate variables, i.e.

$$q_{ij} = \sum_{k=1}^{M} c_{(ij)k} \lambda_k \qquad c_{(ij)k} \in (-\infty, \infty). \tag{1}$$

Further let $q_i = \sum_j q_{ij}$ denote the total exit rate from state i.

For any state i of an acyclic Markov chain, let $P_i(t)$ be the probability that the system is in state i at time t. For any state i, $P_i(t)$ may be written as a polynomial of the form

$$P_i(x) = \sum_j e^{\gamma_{ij} x} [\sum_k a_{ijk} x^k]. \tag{2}$$

The fact that the state probability distributions are of this form is easily derived. First observe that the initial state(s) probability(ies) has (have) this form. The probability of being in (non-initial) state i at time t can be written as:

$$P_i(t) = \int_0^t \sum_{j \in J(i)} P_j(x) q_{ji} e^{-q_i(t-x)} dx \tag{3}$$

where $J(i)$ is the set of states with a transition leading to state i. By induction, it is easy to show that, if every $P_j(x)$ has the form (2), then so does any $P_i(x)$ derived using (3).

We now derive the equations needed to calculate the constants of equation (2) for any state of an actual acyclic Markov chain. Let $S(i)$ be the set of poles of the Laplace-Stieltjes Transform of $P_i(t)$, i.e. the set of the γ_{ij}'s of (2). Let us rename $\gamma^* \triangleq -q_i$ and define $S(J(i)) \triangleq \bigcup_{j \in J(i)} S(j)$. Setting $N(j) = |S(j)|$, we can write $S(j) = \{\gamma_{j1}, \gamma_{j2}, \ldots, \gamma_{jN(j)}\}$. If we number the poles then $P_j(t)$ may be written as:

$$P_j(t) = \sum_{l=1}^{N(j)} e^{\gamma_{jl} t} [\sum_{k=0}^{L(j,l)} a_{jlk} t^k] \tag{4}$$

$L(j,l)$ is the maximum power of t associated with pole γ_{jl} in $P_j(t)$ such that $a_{jlk} \neq 0$. If $N(j) > 1$, it is easy to show using an inductive proof that, for any pole γ_{jl},

$$(a_{jlL(j,l)} \neq 0) \quad => \quad \forall k < L(j,l) \; (a_{jlk} \neq 0) \tag{5}$$

If $N(j)=1$, then $P_j(t)$ is of the form $at^k e^{\gamma t}$. This corresponds to the case where there is only one directed path from the original state to state j and all the transition rates along this path are all equal, i.e. $q_j = -\gamma$. Only in this case is the implication in (5) not satisfied. Thus $P_i(t)$ may be written

$$P_i(t) = \sum_{j \in J(i)} \sum_{l=1}^{N(j)} \int_0^t e^{\gamma_{jl} x} [\sum_{k=0}^{L(j,l)} a_{jlk} x^k q_{ji}] e^{\gamma^*(t-x)} dx \; . \tag{6}$$

But, especially when the $P_j(t)$'s have common poles, i.e. when $|S(J(i))| < \sum_{j \in J(i)} |S(j)|$, this expression reduces to

$$P_i(t) = \sum_{\gamma\in S(J(i))} \int_0^t e^{\gamma x}[\sum_{j\in J(i)} \sum_{l=1}^{N(j)} 1_{(\gamma_{j,l}=\gamma)} (\sum_{k=0}^{L(j,l)} q_{ji} a_{jlk} x^k)] e^{\gamma^*(t-x)} dx$$

$$= \sum_{\gamma\in S(J(i))} \int_0^t \sum_{k=0}^{L_i(\gamma)} a_{\gamma,k}^{\sim} x^k e^{\gamma x} e^{\gamma^*(t-x)} dx \tag{7}$$

where

$$a_{\gamma,k}^{\sim} \triangleq \sum_{j\in J(i)} \sum_{l=1}^{N(j)} 1_{(\gamma_{j,l}=\gamma)} q_{ji} a_{jlk}, \qquad \gamma\in S(J(i)) \tag{8}$$

and

$$L_i(\gamma) = \max_{j\in J(i)} \{ L_{(j,l)} \mid \gamma_{j,l} = \gamma \} . \tag{9}$$

Moving the integral inside the summation we obtain

$$P_i(t) = \sum_{\gamma\in S(J(i))} \sum_{k=0}^{L_i(\gamma)} e^{\gamma^* t} a_{\gamma,k}^{\sim} \int_0^t x^k e^{(\gamma-\gamma^*)x} dx . \tag{10}$$

The resolution of this integral depends on whether γ differs from γ^*. If $\gamma=\gamma^*$, then.

$$\int_0^t x^k e^{(\gamma-\gamma^*)x} dx = \int_0^t x^k dx = \frac{t^{k+1}}{(k+1)} . \tag{11}$$

Otherwise, when $S=(\gamma-\gamma^*)$, integration by parts yields

$$\int_0^t x^k e^{Sx} dx = e^{St}[\sum_{l=0}^{k} (-1)^l \frac{k!}{(k-l)!} \frac{t^{k-l}}{S^{l+1}}] + (-1)^{k+1} \frac{k!}{S^{k+1}} . \tag{12}$$

We can now write the equation for $P_i(t)$

$$P_i(t) = \sum_{\gamma\in S(J(i))} \{ 1_{(\gamma\neq\gamma^*} \sum_{k=0}^{L_i(\gamma)} [e^{\gamma t} a_{\gamma,k}^{\sim} (\sum_{l=0}^{k} (-1)^l \frac{k!}{(k-l)!} \frac{t^{k-l}}{S^{l+1}})$$

$$+ a_{\gamma,k}^{\sim} (-1)^{k+1} \frac{k!}{S^{k+1}} e^{\gamma^* t}] + 1_{(\gamma=\gamma^*)} \sum_{k=0}^{L_i(\gamma^*)} e^{\gamma^* t} a_{\gamma^*,k}^{\sim} \frac{t^{k+1}}{(k+1)} \} . \tag{13}$$

We note that

$$S(i) = \{\gamma^*\} \cup S(J(i)) \tag{14}$$

and define

$$L(i,l) = \begin{cases} L_i(\gamma_l) & \text{if} \gamma_l \neq \gamma^* \\ L_i(\gamma_l)+1 & \text{if} \gamma_l = \gamma^* \text{ and } \gamma^* \in S(J(i)) \\ 0 & \text{if} \gamma_l = \gamma^* \text{ and } \gamma^* \notin S(J(i)) \end{cases} \tag{15}$$

From (13) note that, if $u = k-l$, we have

$$\sum_{k=0}^{L_i(\gamma)} a_{\gamma,k}^{\sim} [\sum_{l=0}^{k} (-1)^l \frac{k!}{(k-l)!} \times \frac{t^{k-l}}{S^{l+1}}]$$

$$= \sum_{k=0}^{L_i(\gamma)} a_{\gamma,k}^{\sim} [\sum_{u=0}^{k} (-1)^{k-u} \frac{k!}{u!} \times \frac{t^u}{S^{k+1-u}}]$$

$$= \sum_{u=0}^{L_i(\gamma)} t^u \sum_{k=u}^{L_i(\gamma)} a_{\gamma,k}^{\sim} (-1)^{k-u} \frac{k!}{u!} \times \frac{1}{S^{k+1-u}}$$

If we write $P_i(t)$ as

$$P_i(t) = \sum_{l=1}^{N(i)} e^{\gamma_l t} \sum_{k=0.}^{L(i,l)} a'_{\gamma_l,k} t^k \tag{16}$$

then

$$a'_{\gamma,u} = \sum_{r=u}^{L_i(\gamma)} \cdot (-1)^{r-u} \frac{r!}{u!} \times \frac{a_{\gamma,r}^{\sim}}{S^{r+1-u}}.$$

If $\gamma_l \neq \gamma^*$, we have

$$a'_{\gamma_l,k} = \sum_{r=k}^{L_i(\gamma_l)} (-1)^{r-k} a_{\gamma_l,r}^{\sim} \cdot \frac{r!}{k!} \frac{1}{(\gamma_l - \gamma^*)^{r-k+1}} \quad k=0,\ldots,L_i(\gamma_l) \tag{17}$$

If $\gamma^* \notin S(J(i))$ then

$$a'_{\gamma^*,0} = \sum_{l=1}^{N(i)} 1_{(\gamma_l \neq \gamma^*)} \sum_{k=0}^{L_i(\gamma_l)} (-1)^{k+1} \cdot k! \frac{a_{\gamma_l,k}^{\sim}}{(\gamma_l - \gamma^*)^{k+1}} \tag{18}$$

Otherwise, if $\gamma^* \in S(J(i))$,

$$a'_{\gamma^*,k+1} = \frac{a_{\gamma^*,k}^{\sim}}{(k+1)} \qquad k=0,\ldots,L_i(\gamma^*) \tag{19}$$

With these equations, we can easily compute the coefficients of the polynomials in t that multiply the exponentials in the state probability expressions.

3. Implementation

In this section we outline the procedure used by the ACE program to compute state probability expressions for an acyclic Markov Chain. Techniques for chains with cycles are more complex, requiring either the symbolic solution of a set of equations or some form of approximation. We also briefly discuss the operation and user interface of ACE.

The ACE procedure is detailed in the Appendix; we briefly outline it here. First the states are sorted according to the partial order induced by the transitions. For state i, the probability expression is computed by first determining all the poles of states that have transitions leading to state i. If the pole associated with state i's outgoing transition is not in the incoming set, coefficients for the polynomials multiplying the incoming exponential terms are computed using equation (17). The new pole's polynomial multiplier is a constant computed using equation (18). If the outgoing pole is also in the set of incoming poles, the degree of its polynomial will be incremented. The new coefficients for the incremented polynomial can be computed using equation (19).

ACE is being developed as the first stage of a testbed for aggregation techniques. Two versions are currently being implemented. The first version supports an unlimited number of symbolic variables but generates answers that are symbolic only in the poles (the powers of the exponential terms). The coefficients of the polynomials in t that multiply the exponentials are numeric. The second version of ACE is fully symbolic in one variable and numeric in other variables, i.e. the coefficients of the polynomials in t that multiply the exponentials have both numeric and symbolic parts. This allows us to conduct a parametric sensitivity analysis in a fully symbolic fashion. Eventually these two methods will be combined yielding completely symbolic poles and coefficients that are fully symbolic in at least one variable.

Several problems have arisen in constructing the ACE package. When computing symbolic coefficients, the size of the coefficients grows linearly with the number of symbolic variables used along all paths to the state. The coefficients rapidly reach an unmanageable size, even for a small chain. Restricting the lengths of the paths through the chain would greatly reduce the package's utility, particularly for chains that are "long" (e.g. simple death processes). Instead, we restrict the number of symbolic variables that are maintained in a given run of the program. All variables not treated symbolically are merged numerically. If poles are still maintained in a fully symbolic fashion, care must be taken to correctly merge the numeric values of symbolically different, numerically identical poles.

Further efforts, include the construction of a "user-friendly" interface and the addition of a block definition and solution facility. The user will be able to define blocks of states with fixed entry and exit points. The blocks could be evaluated by direct insertion of their states into the chain. Alternatively, the block could be solved in isolation using symbolic or numerical approximation methods. This capability should further facilitate the use of ACE in evaluating aggregation methods.

4. Examples and Conclusions

In this section we demonstrate the use of ACE-like symbolic computation. We begin by symbolically solving the example given in Figure 2 using the method described in sections 2 and 3. We then apply a simple aggregation technique to the chain and re-solve the system. We give examples that demonstrate the utility of a symbolic solution for bounding, sensitivity analysis, and comparison of aggregation techniques.

4.1 Exact Solution of 3-Component System

Given the chain shown in Figure 2, we follow the algorithm outline given in section 3. We first observe that the only γ for the probability distribution of state 3 is -3λ. As state 3 has no parent the constant $\alpha'_{1,0}=1$. Accordingly we can write:

$$P_3(t) = e^{-3\lambda t}. \tag{20}$$

We can continue following the algorithm sketch and derive the following equations for the functioning states:

$$P_{2R}(t) = -\frac{3\lambda}{\lambda-\delta}e^{-3\lambda t} + \frac{3\lambda}{\lambda-\delta}e^{-(2\lambda+\delta)t} \tag{21}$$

$$P_2(t) = \frac{3\delta}{\lambda-\delta}e^{-3\lambda t} - \frac{3\lambda}{\lambda-\delta}e^{-(2\lambda+\delta)t} + 3e^{-2\lambda t} \tag{22}$$

$$P_{1R}(t) = -\frac{6\lambda\delta}{(\lambda-\delta)(2\lambda-\delta)}e^{-3\lambda t} + \frac{6\lambda}{\lambda-\delta}e^{-(2\lambda+\delta)t} - \frac{6\lambda}{\lambda-\delta}e^{-2\lambda t} + \frac{6\lambda\delta}{(\lambda-\delta)(2\lambda-\delta)}e^{-(\lambda+\delta)t} \tag{23}$$

$$P_1(t) = \frac{3\delta^2}{(\lambda-\delta)(2\lambda-\delta)}e^{-3\lambda t} - \frac{6\lambda\delta}{(\lambda-\delta)(\lambda+\delta)}e^{-(2\lambda+\delta)t} + \frac{6\delta}{\lambda-\delta}e^{-2\lambda t} - \frac{6\lambda\delta}{(\lambda-\delta)(2\lambda-\delta)}e^{-(\lambda+\delta)t} + \frac{3\delta}{(\lambda+\delta)}e^{-\lambda t} \tag{24}$$

For the state that corresponds to failure due to exhaustion of components we can write:

$$P_{F_e}(t) = -\frac{\delta^2}{(\lambda-\delta)(2\lambda-\delta)}e^{-3\lambda t} + \frac{6\lambda^2\delta}{(\lambda-\delta)(\lambda+\delta)(2\lambda+\delta)}e^{-(2\lambda+\delta)t} - \frac{3\delta}{\lambda-\delta}e^{-2\lambda t} + \frac{6\lambda^2\delta}{(\lambda-\delta)(\lambda+\delta)(2\lambda-\delta)}e^{-(\lambda+\delta)t} - \frac{3\delta}{\lambda+\delta}e^{-\lambda t} + \frac{\delta^2}{(2\lambda+\delta)(\lambda+\delta)} \tag{25}$$

For the state that corresponds to a coverage failure we can write:

$$P_{F_c}(t) = (\frac{2\lambda}{\lambda-\delta} + \frac{2\lambda\delta}{(\lambda-\delta)(2\lambda-\delta)})e^{-3\lambda t} - \frac{12\lambda^2}{(\lambda-\delta)(2\lambda+\delta)}e^{-(2\lambda+\delta)t} + \frac{3\lambda}{\lambda-\delta}e^{-2\lambda t} - \frac{6\lambda^2\delta}{(\lambda-\delta)(\lambda+\delta)(2\lambda-\delta)}e^{-(\lambda+\delta)t} + \frac{2\lambda}{2\lambda+\delta} + \frac{\lambda\delta}{(2\lambda+\delta)(\lambda+\delta)} \tag{26}$$

We note that the reliability of the 3-component system is given by

$$R(t) = 1 - (P_{F_e}(t) + P_{F_c}(t)) \tag{27}$$

To derive the information contained in this symbolic reliability expression, even a highly flexible conventional reliability evaluation package would require several runs for different parameter values. For example, to see the effect of the fault-handling rate we consider the reliability expression as a function of delta. Fixing $\lambda=10^{-4}$

failure/hour and $t = 10$ hours, in Figure 3 we graph $-log_{10}$ of the system unreliability as a function of $-log_{10}$ of the mean time to handle a fault. When faults are handled quickly (in this case about 100 ms), we see that the reliability of the system approaches that of a system with perfect coverage. When fault-handling is slow (minutes or hours), imperfect coverage dramatically reduces the system's reliability.

4.2 An Approximate Solution of the 3-Component System

When Markov models are used for realistic systems, the state space often grows beyond practical limits. Accordingly, reliability evaluation packages often use various aggregation or lumping methods to reduce the size of the state space. For example, a system model can be decomposed into sub-models of fault-handling and fault-occurrence behavior.[16] Short of assuming all faults are successfully handled, one of the simplest approaches is to condense the second fault rate and fault-handling parameters into a single constant c denoting *coverage*, the probability that an arbitrary fault is successfully handled. When this approach is used with our example, we obtain the chain shown in Figure 4. Its state probability equations are

$$P_{3A}(t) = e^{-3\lambda t} \tag{28}$$

$$P_{2A}(t) = -3c_1 e^{-3\lambda t} + 3c_1 e^{-2\lambda t} \tag{29}$$

$$P_{1A}(t) = 3c_1c_2\, e^{-3\lambda t} - 6c_1c_2\, e^{-2\lambda t} + 3c_1c_2\, e^{-\lambda t} \tag{30}$$

$$P_{F_e}(t) = -c_1c_2\, e^{-3\lambda t} + 3c_1c_2\, e^{-2\lambda t} - 3c_1c_2\, e^{-\lambda t} + c_1c_2 \tag{31}$$

$$P_{F_c}(t) = (2(1-c_2)c_1 - (1-c_1))e^{-3\lambda t} - 3c_1(1-c_2)e^{-2\lambda t} + (1-c_1) + c_1(1-c_2) \tag{32}$$

As in the original chain,

$$R(t) = 1 - (P_{F_e}(t) + P_{F_c}(t))$$

One interesting problem is correctly choosing values for the c parameter. If two c parameters were employed, as in our example, the instantaneous coverage approximation would usually be

$$c_1 = \frac{\delta}{2\lambda+\delta} \qquad (33)$$

$$c_2 = \frac{\delta}{\lambda+\delta} \qquad (34)$$

However, if only a single coverage parameter were chosen, the choice might well depend on the period of time over which we were interested in evaluating the reliability of the system.

With the reliability expressions for both the aggregated and original Markov chains, we can evaluate the acceptability of the aggregation scheme by comparing the results they produce. Figure 4 shows the graphs of three estimates of $-log_{10}$ of system unreliability as a function of $\log_{10}$ of t. With λ fixed at 10^{-4} failures/hr. for all three curves, the lower curve is derived by solving the original Markov model in Figure 2 with $\delta = 1$. Using this δ value, the middle curve is derived using the aggregated chain in Figure 4 and instantaneous coverage estimates derived from equations (33) and (34). The upper curve is derived using a naive *perfect coverage* model, i.e. fault handling is assumed to always succeed instantaneously. Even for this contrived situation (λ and δ are probably much closer in value than they would be in practice), we see that a constant coverage assumption still can provide a good estimate of system reliability. For our particular example, if a more realistic δ value is chosen, the reliability estimates provided by the original and aggregated chains are essentially identical Extending this validation of a simple approximation scheme for a small model to more realistic models may require significant effort.

Symbolic solutions of CTMC should provide at least two benefits. First, it should be possible to compare the results obtained by exact and approximate solution methods for small to medium sized CTMC. By indicating the magnitude of error that approximate solutions introduce, this type of analysis should provide a good indication of an aggregation/approximation technique's utility for larger, more realistic problems. Second, symbolic solutions allow us to easily examine the influence of changing parameter values on the solutions of Markov models. This type of investigation could be very expensive using conventional simulation or numerical solution techniques. By providing easy access to symbolic solutions of CTMC, the ACE package should enhance our ability to study Markov reliability models. and approximation techniques for their solution.

References

1. D. F. Towsley, J. C. Browne, and K. M. Chandy, "Models for Parallel Processing Within Programs: Applications to CPU:I/O and I/O:I/O Overlap," *CACM*, pp. 821-31, October 1978.
2. Steven A. Lavenberg, ed., *Computer Performance Modeling Handbook*, Academic Press, 1983.
3. Kishor S. Trivedi, *Probability & Statistics with Reliability, Queuing & Computer Science Applications*, Prentice-Hall, 1982.
4. Ying-Wah Ng and Algirdas Avizienis, "A Unified Reliability Model for Fault-Tolerant Computers," *IEEE Transactions on Computers*, pp. 1002-1011, November 1980.
5. Robert Geist and Kishor Trivedi, "Ultra-High Reliability Prediction for Fault-Tolerant Computer Systems," *IEEE Transactions on Computers*, pp. 1118-1127, December, 1983.
6. Daniel P. Siewiorek and Robert S. Swarz, *The Theory and Practice of Reliable System Design*, Digital Press, 1982.
7. John F. Meyer, "On Evaluating the Performability of Degradable Computer Systems," *IEEE Transactions on Computers*, August 1980.
8. Ambuj Goyal and Asser Tantawi, "Evaluation of Performability in Acyclic Markov Chains," IBM Computer Science Research Report, May 1984.
9. D.R. Cox, "The Use of Complex Probability in the Theory of Stochastic Processes," in *Proceedings Cambridge Philosophical Society*, 1955.
10. Marcel F. Neuts, *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*, John Hopkins University Press, 1981.
11. A. Costes, J. E. Doucet, C. Landrault, and J.C. Laprie, "SURF: A Program for Dependability Evaluation of Complex Fault-Tolerant Computing Systems," in *Proceedings IEEE 11-th Fault Tolerant Computing Symposium*, pp. 72-78, June 1981.
12. Sheldon M. Ross, *Stochastic Processes*, J. Wiley & Sons, 1983.
13. J. J. Stiffler, L. A. Bryant, and L. Guccione, "CARE III Final Report Phase I," NASA Contractor Report 159122, November 1979.
14. Kishor Trivedi and Robert Geist, "A Tutorial on the CARE III Approach to Reliability Modeling," NASA Contractor Report 3488, 1981.
15. Robin Sahner and Kishor Trivedi, "SPADE: Series-Parallel Directed Acyclic Graph Evaluator," Technical Report CS-1984-15, Department of Computer Science, Duke University, 1984. Submitted for Publication
16. Kishor Trivedi, Robert Geist, Mark Smotherman, and Joanne Bechta Dugan, "Hybrid Modeling of Fault-Tolerant Computer Systems," *International Journal of Computers and Electrical Engineering*, 1984. To appear, special issue on "Reliability and Verification of Computing Systems."
17. R. B. Conn, P. M. Merryman, and K. L. Whitelaw, "CAST - A Complimentary Analytic-Simulative Technique for Modeling Fault-Tolerant Computing Systems," in *Proceedings AIAA Computers in Aerospace Conference*, Los Angeles, November, 1977.
18. S.V. Makam and A. Avizienis, "ARIES 81: A reliability and life-cycle evaluation tool for fault-tolerant systems," in *Proceedings IEEE 12-th Fault Tolerant Computing Symposium*, pp. 267-274, June 1982.

Appendix - The ACE Procedure

GIVEN: An acyclic Markov chain with state space $\{1,2,\ldots,N\}$ sorted so that the ancestors of a every state precede it in the list (for convenience).

```
BEGIN
    FOR each state $s \in \{1,2,\ldots,N\}$
        Determine $J(s)$ the set of states with transitions leading to $s$
        Determine $S(J(s))$ and $\{L_i(\gamma),\ \gamma \in S(J(s))\}$ using (9) and (15)
        Compute $a_{\gamma,k}$  $k=0,\ldots,L_i(\gamma)$,  $\gamma \in S(J(i))$
        Let $\gamma^* = q_i$ and  $a'_{\gamma^*,0}=0$
            Note: $\gamma^*$ for an absorbing state is 0.
        FOR $\gamma \in S(J(i))$, $\gamma \neq \gamma^*$
            FOR $k=0,1,\ldots,L_i(\gamma)$
                Compute $a'_{\gamma,k}$ using formula (17)
                Accumulate $a'_{\gamma^*,0} = a'_{\gamma^*,0} + (-1)^{k+1} \cdot k! \frac{a_{\gamma,k}}{(\gamma-\gamma^*)^{k+1}}$
            END FOR
        END FOR
        If $\gamma^* \in S(J(i))$, compute $a'_{\gamma,k+1}$ from (19) for $k=0,\ldots,L_i(\gamma^*)$
        $P_s(t) = \sum_{\gamma \in S(J(i))} e^{\gamma t} \sum_{k=0}^{L_i(\gamma)} a'_{\gamma,k} t^k$
    END FOR
END
```

Method: Simulation

- **Package:** CAST[17]
  - **Note:** Allows full simulation of a restricted class of systems

Method: Differential Equations (Numerical Solution)

- **Package:** HARP[5]
  - **Domain:** Homogeneous and Non-Homogeneous acyclic CTMC
- **Package:** SAVE*
  - **Domain:** Cyclic CTMC

Method: Differential Equations (Laplace Solution)

- **Package:** SURF[11]
  - **Domain:** Non-Markov Processes
  - **Note:** Approximate solution using Coxian method of stages

Method: Integral Equations (Numerical Solution)

- **Package:** Care III Coverage Model[13]
  - **Domain:** Semi-Markov Processes
- **Package:** Care III Reliability Model[13]
  - **Domain:** Non-Homogeneous CTMC

Method: Closed Form Solution

- **Package:** ARIES[18]
  - **Domain:** Cyclic Homogeneous CTMC
  - **Note:** Poles and their coefficients derived numerically
- **Package:** ACE
  - **Domain:** Acyclic Homogeneous CTMC
  - **Note:** Poles and their coefficients derived symbolically

Figure 1: Reliability Modeling Packages Employing Markov Chain Techniques

*"System Availability Estimator", in preparation under an IBM-Duke Joint Study.

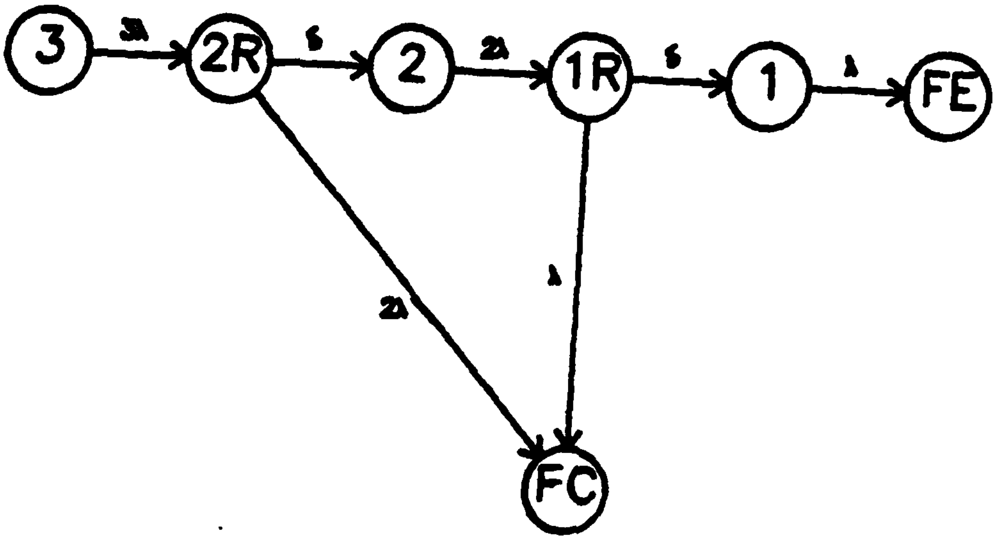


Figure 2: Markov Chain State Diagram for a 3-Component Parallel Redundant System
Coverage failure results only from a second fault occurrence during reconfiguration

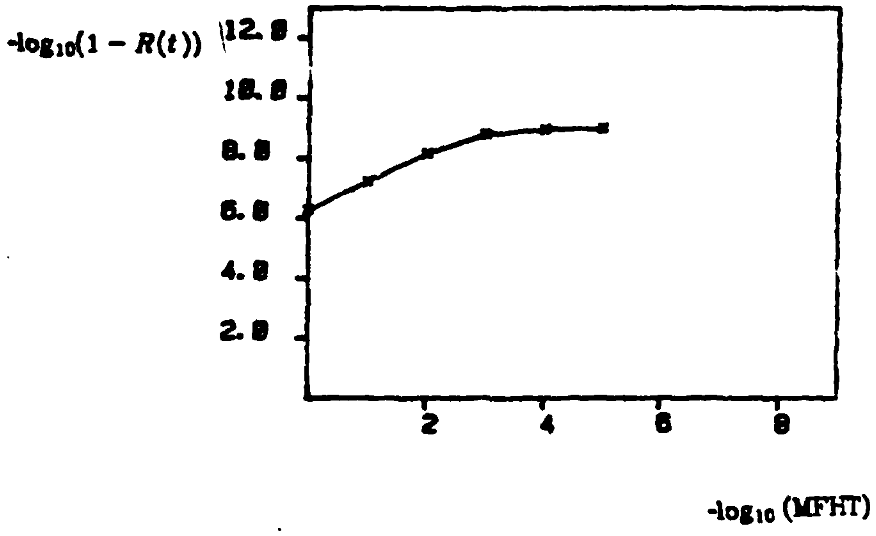


Figure 3: Effect of Fault-Handling Rate on Reliability

Mission Time = 10 hrs. $\lambda = 10^{-4}$

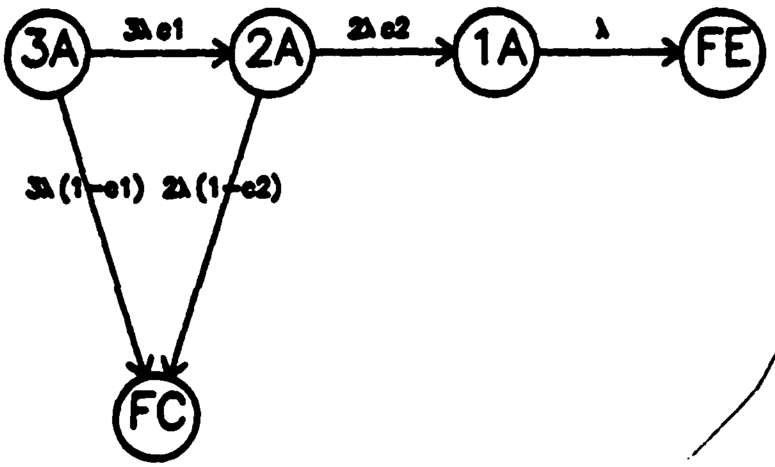


Figure 4: Aggregated Markov Chain State Diagram for 3-Component Parallel Redundant System
C_1 and C_2 are instanteous coverage estimates

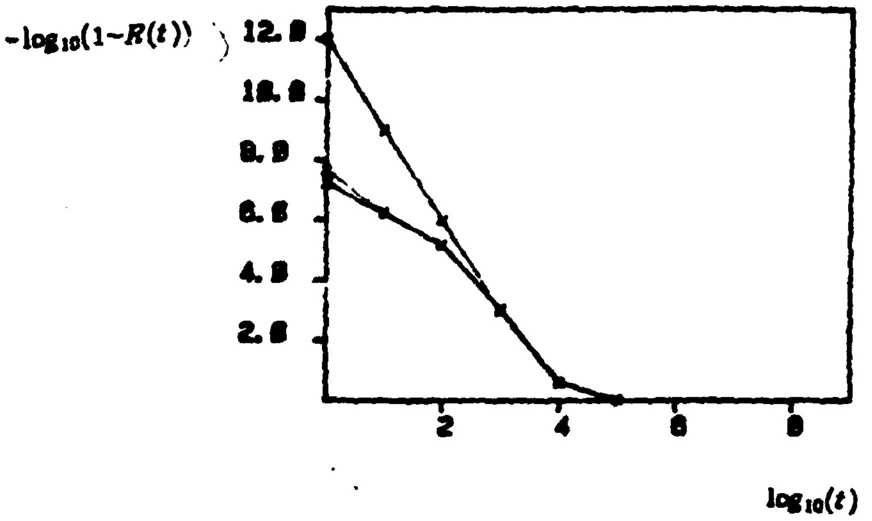
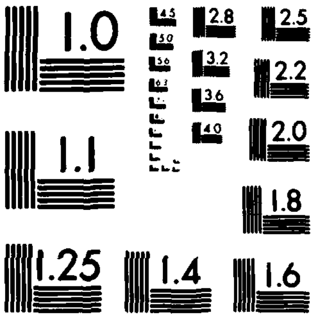


Figure 5: Aggregation's Effect on Reliability Estimates

The bottom curve is an estimate of reliability derived from the *original* Markov chain.

The middle curve is derived from the aggregated chain in Figure 4

The top curve is a *perfect coverage* estimate



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A