Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

COMPUTING BOUNDS ON THE STEADY STATE AVAILABILITY OF REPAIRABLE COMPUTER SYSTEMS

Richard R. Muntz
John C.S. Lui

December 1989
CSD-890066

Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

COMPUTING BOUNDS ON THE STEADY STATE AVAILABILITY OF REPAIRABLE COMPUTER SYSTEMS

Richard R. Muntz
John C.S. Lui

December 1989
CSD-890066

Computing Bounds on the Steady State Availability of Repairable Computer Systems*

Richard R. Muntz John C.S. Lui

UCLA Computer Science Department

Abstract

System availability is an important reliability measure for computer designers. Most often Markov models are used in representing systems for reliability/availability analysis. Yet due to the size and complex nature of these systems, the model often has an unmanageably large state space and it quickly becomes impractical to even generate all the states in the system model. In this paper, we present a methodology for bounding steady state availability and at the same time, drastically reduce the state space of the models that must be solved. The bounding method generates part of the transition matrix at each step and also at each step, computes tighter availability bounds. We also present an iterative method to refine the error accumulated at each step. This general bounding methodology provides an efficient way to evaluate reliability models with large state spaces without generating the entire transition matrix.

*This research was supported by a University of California MICRO grant from IBM.

1 Introduction

Computer system availability and reliability are crucial measures for system designers, particularly life situation or when large financial loss is possible. System dependability has long been an active area of research [5,7,9,10,19] and tools [2,3,8,12,20] have built to aide the specification and evaluation of dependability models. Increased complexity of modern systems and their sophisticated reliability requirements continue to challenge us to find effective ways to evaluate their performance and reliability. There are two major types of dependability measures that are of interest. One type consists of transient dependability measures, e.g, mean time availability over a mission period, while the other contains steady state dependability measures, e.g, steady state availability for system that are expected to operate for a long period of time. In this paper, we will concentrate on the computation of steady state availability, i.e, the fraction of time that a system is operational.

Various methods have been proposed for dependability analysis. This includes combinatoric analysis, Markov and semi-Markov analysis and simulation. Continuous time Markov models are still the most widely used. Unfortunately, due to the complex nature of real systems (e.g. complex interaction of components, complex scheduling and repair policy etc), closed form solutions are extremely difficult to obtain. Therefore numerical solution techniques are widely used. The most pervasive limitation of numerical techniques for analyzing Markov models is the inability to handle the large state spaces of realistic models. The Markov models of real systems generally have a very large state space and they can easily outstripped the memory and storage capacity of current (or foreseeable) systems [16]. Various methods have been proposed to calculate different performance and reliability measures from large Markov models. Among these are the aggregation-disaggregation method for chains with *nearly completely decomposable structure* [4], the iterative aggregation-disaggregation method [17,18], and matrix-geometric based methods [15].

A Markov chain is nearly completely decomposable if the interactions between groups of states are not comparable with the interactions within the groups. This is usually not the case for reliability models of computer systems and the aggregation-disaggregation method is unlikely to yield a sufficiently accurate approximation. The iteration aggregation-disaggregation method uses a method similar to the Gauss-Seidel iteration technique. One problem is that the convergence rate may be slow. Also, existence and uniqueness are difficult to show. The matrix-geometric techniques require certain regularity of structure that is often missing from reliability models. In this paper,we will propose a methodology that provides efficient calculation of bounds on steady state availability by explicitly taking advantage of the properties of reliability models.

Exact reliability measures are generally not required but rather we are satisfied with sufficiently accurate approximate results. If the approximate results can be given in term of upper and lower bounds then this is ideal. Recently, the results reported in [14,11] provided a method to compute bounds on steady availability for models of repairable computer systems. In [14], rather than generating all the states in the system model, the approach calls for generating those states that account for most of the probability mass while all other states are grouped into a small number of aggregate states. With this approach a decision is made a priori as how much of the transition rate matrix to expand in detail. In [11], this approach was extended to a "multi-step" procedure in which successively more of the transition rate matrix is generated while at the same time obtaining tighter availability bounds. In this paper, we will present a general bounding algorithm that will include the "one-step", the "multi-step" and an iterative error refinement procedure.

We will present the "one-step" and the "multi-step" bounding algorithms in Section 2 and Section 3. The error-refinement process in presented in Section 4. Section 5 will provide a heuristic algorithm to decide whether to generating more transition rate matrix or go backward and refine the accumulated errors. The final algorithm for bounding availability is presented and an illustrative example is given in Section 6.

2 One-step Bounding Algorithm

Many performance and reliability measures can be expressed in terms of a reward function. If $r(i)$ is the reward rate for state i, the expected reward rate, $\mathcal{M}$ can be expressed as :

$$\mathcal{M} = \sum_{i \in S} r(i)\pi(i) \tag{1}$$

with S being the state space of the Markov model and $\pi(i)$ being the stationary state probability for state i. Availability is a special case in which the "operational" states have reward of 1 and the "non-operational" states have reward of 0.

Systems are generally designed to have a high level of availability. It is reasonable therefore to expect that during the life time of the system, most of the components are operational. With this in mind, we partition the state space of the model into $\mathcal{F}_i$, $0 \leq i \leq n$, where n is the number of system components and where $\mathcal{F}_i$ contains all the states that have *exactly* i failed components. The idea is to represent the detailed behavior of the model for $\mathcal{F}_i$, $i \leq K$ for some small value of K and approximate the remainder of the model via aggregation.

The transition rate matrix can then be viewed as shown in Figure 1 in which submatrix Q_{ii} corresponds to $\mathcal{F}_i$. In this figure the submatrices denoted by 0 contain all zero elements. This is a consequence of an assumption that there is zero probability of two or more components becoming operational at exactly the same time. Note that this does not preclude multiple repair facilities or any other common feature of reliability models.

We now summarize one of the results from [14]. Consider three sets of states:

$$\mathcal{G}_0 = \mathcal{F}_0; \qquad \mathcal{G}_1 = \{\bigcup_{i=1}^{K} \mathcal{F}_i\}; \qquad \mathcal{G}_2 = \{\bigcup_{i=K+1}^{n} \mathcal{F}_i\}.$$

$$\begin{bmatrix} Q_{00} & Q_{01} & Q_{02} & & \cdots & & Q_{0n} \\ Q_{10} & Q_{11} & Q_{12} & & \cdots & & Q_{1n} \\ 0 & Q_{21} & Q_{22} & & \cdots & & \\ 0 & 0 & Q_{32} & Q_{33} & & & \cdot \\ \cdot & & 0 & Q_{43} & Q_{44} & & \cdot \\ \cdot & \cdots & & \ddots & & \ddots & \cdot \\ \cdot & & & & & & \\ 0 & \cdots & & & 0 & Q_{n-1,n} & Q_{nn} \end{bmatrix}$$

Figure 1: Transition matrix.

Figure 2 illustrates the transition matrix G in which G_{ii} is the principal submatrix corresponding to states in $\mathcal{G}_i$. (The submatrix shown as 0 is a consequence of the "nearest neighbor" property discussed previously.) Now construct a new transition matrix G' as shown in Figure 3. The relationship between the process defined by G and that defined by G' is illustrated in Figure 4. Basically, in the new process there are two sets of states corresponding to the states $\mathcal{G}_1$ of the original model. Let us call them $\mathcal{G}'_{1u}$ and $\mathcal{G}'_{1d}$ as shown in Figure 4. The idea behind this transformation can be explained as follows. Assume the system starts in the "all components up" state, i.e. $\mathcal{F}_0$. As components fail and are repaired the system will stay in states in $\mathcal{G}_0$ and $\mathcal{G}'_{1u}$ until the first time that there are $K+1$ or more failed components. At this point the system is in a state of $\mathcal{G}_2$. However when the number of failed components falls to K, the system now enters a state in $\mathcal{G}'_{1d}$. (Now the notation is explainable; "u" stands for "going Up" and "d" stands for "going Down". As the number of failed components goes up, the system visits states in $\mathcal{G}'_{1u}$ and after $K+1$ failures have been reached, it visits the states in $\mathcal{G}'_{1d}$ as the number of failed components goes down.)

From the construction it is easy to show that the two transition matrices are such that the steady state probabilities of the original process(G), can be calculated from the steady state probabilities of the second process (G'). Specifically, if

$[\pi'_0, \pi'_{1_1}, \pi'_{1_2}, \pi'_2]$ is the solution of $\pi' G' = 0$ and $\sum \pi'(i) = 1$, then

$[\pi'_0, \pi'_{1_1} + \pi'_{1_2}, \pi'_2]$ is the solution of $\pi G = 0$ and $\sum \pi(i) = 1$

There is a natural mapping of the states in G' to states in G. In terms of rewards, the reward function for each state in G' is the same reward as the associated state in G. It is clear then that the mean availabilities of the two systems are identical.

Now consider aggregation of the states $\mathcal{F}_i$, $i \geq K+1$ and $\mathcal{F}'_i$, $1 \leq i \leq K$ as illustrated in Figure 5. The corresponding transition rate matrix is shown in Figure 6. The results from [14] show that if the rates indicated by a '+' are replaced by upper bounds (on the actual values) and the rates indicated by '−' are replaced by lower bounds then bounds

$$\begin{bmatrix} G_{00} & G_{01} & G_{02} \\ G_{10} & G_{11} & G_{12} \\ 0 & G_{21} & G_{22} \end{bmatrix}$$

Figure 2: Matrix G.

$$\left[\begin{array}{cc|cc} G_{00} & G_{01} & 0 & G_{02} \\ G_{10} & G_{11} & 0 & G_{12} \\ \hline G_{10} & 0 & G_{11} & G_{12} \\ 0 & 0 & G_{21} & G_{22} \end{array}\right]$$

Figure 3: Duplication of states. Matrix G'.

on the steady state availability, $\mathcal{A}$, can be expressed as:

$$\begin{aligned} \sum_{i \in D} r(i)\pi'(i) \quad &\leq \quad \mathcal{A} \quad \leq \quad \sum_{i \in D} r(i)\pi'(i) + \left(1 - \sum_{i \in D} \pi'(i)\right) \qquad (2) \\ where \;\; r(i) \quad &= \quad 1 \;\; for\ operational\ states \\ &= \quad 0 \;\; for\ non-operational\ states \end{aligned}$$

The $\mathcal{D} = \mathcal{G}_0 \bigcup \mathcal{G}'_{1u}$ and $\pi'(i)$ is the steady state probability for state i in the model with the indicated replacement of transition rates by bounds.

Since the above result is valid regardless of the set of operational states it follows trivially that:

$$\pi'(i) \leq \pi(i) \quad \forall\ states\ i \in \mathcal{D}$$

thus we can view the results from [14] as providing a means of obtaining lower bounds on the stationary state probabilities of states in D .

3 Multi-step Bounding Algorithm.

In the "single step" procedure described in the previous section the decision as to the dimension of the matrix G_{11} is made a priori. This implies that once the bounds have been calculated there is no means provided for utilizing the work that has been already done in solving the first portion of the matrix to further tighten the bounds. In [11], we presented a

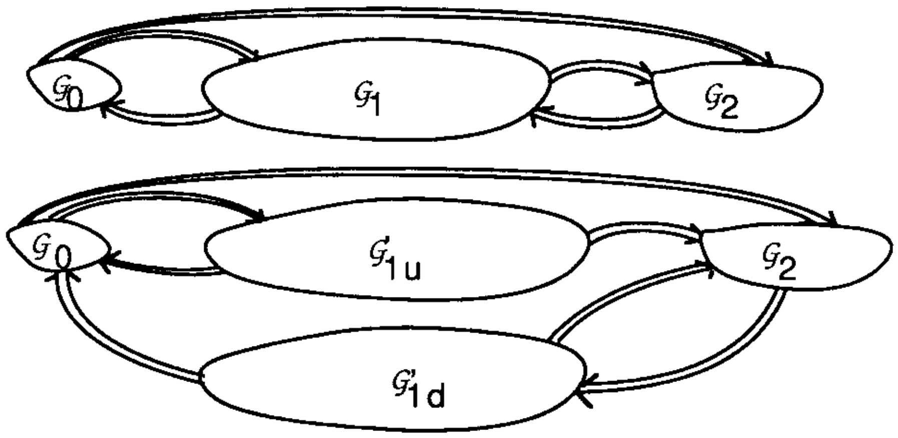

Figure 4: Relationship of G and G'.

method (*Multi-step Bounding algorithm*)that can alleviate these difficulties. The algorithm allows an incremental generation of the transition matrix. At each step a new portion of the matrix is generated. Further, at each step the results from the previous steps are used to form a transition rate matrix whose solution allows us to bound the stationary state probabilities for an additional set of states. This allows us to incrementally improve the bounds on availability.

At each step (after the first) there are three sets of states of the original model that we will need to distinguish.

$\mathcal{D}'$ = set of states for which a lower bounds on the stationary state probabilities were obtained in previous steps.
$\mathcal{D}$ = set of states which are the center of attention for this step and for which a lower bound on the stationary state probabilities will be calculated in this step.
$\mathcal{A}$ = the complement of $\mathcal{D}' \bigcup \mathcal{D}$.

Figure 7 illustrates this partitioning of the state space in terms of the transition matrix G. Following the development in [11], we describe a short sequence of transformations to G. Each transformation is such that in the resulting model, the stationary state probabilities for the states in $\mathcal{D}$ are always (individually) bounded from below. We start by constructing the matrix G_1 from G as illustrated in Figure 8. G_1 corresponds to a model in which the states in $\mathcal{D}$ have been replicated. The set of replicate states (referred to as "clones") will be denoted by $\mathcal{C}$. Note that in G_1 the submatrix Q_{CC} is equal to Q_{DD}. We use the notation Q_{CC} because it enhances readability.

Notation: We use the notation π_{D/G_i} to denote the vector of stationary state probabilities for states in a the set of states $\mathcal{D}$ when the transition rate matrix is G_i.

In the previous section while reviewing the results from [14] a similar construction was

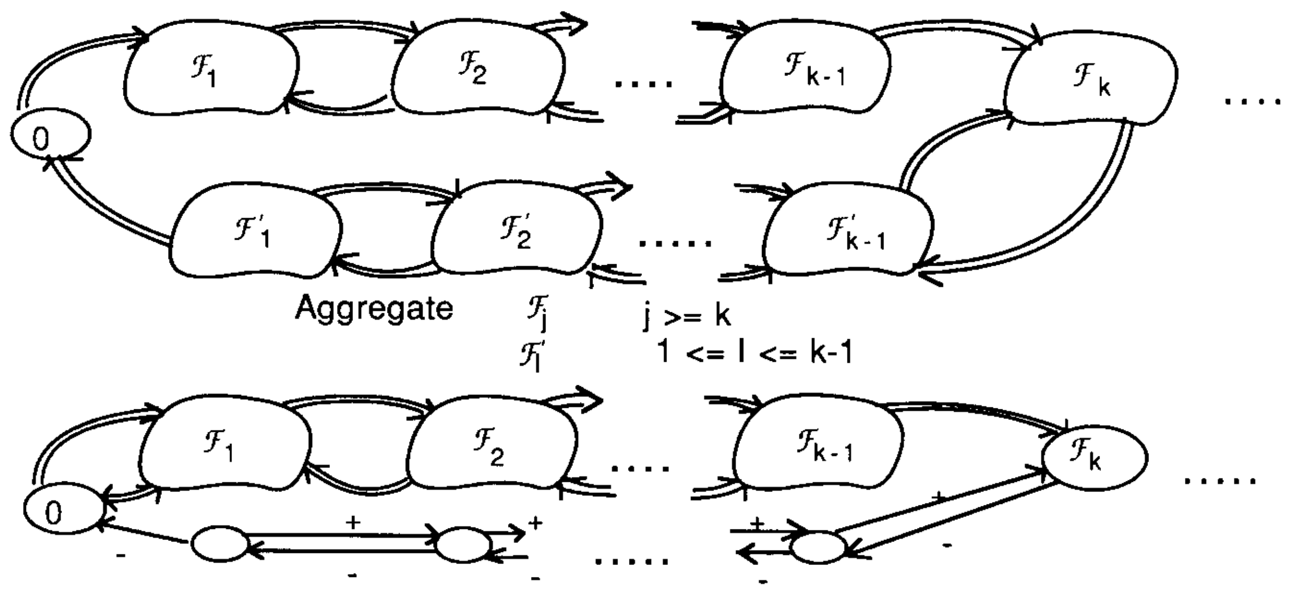

Figure 5: Aggregation of states.

described. From that discussion it is clear that if

$[\pi_{D'/G_1}, \pi_{D/G_1}, \pi_{C/G_1}, \pi_{A/G_1}]$ is the solution of $\pi_1 G_1 = 0$ and $\sum \pi_{G_1}(i) = 1$, then

$[\pi_{D'/G_1}, \pi_{D/G_1} + \pi_{C/G_1}, \pi_{A/G_1}]$ is the solution of $\pi G = 0$ and $\sum \pi_G(i) = 1$.

It follows immediately that $\pi_{D/G_1} \leq \pi_{D/G}$ since $\pi_{C/G_1} \geq 0$.

In the next several transformations we make use of the fact that it is possible to perform exact aggregation of a transition rate matrix. We assume that the reader is familiar with the basic aggregation/disaggregation approximation procedure as described in [4]. Later we show that exact aggregation is not actually required in the computation of the bounds. We merely use exact aggregation in the intermediate steps of the development.

$$\left[\begin{array}{cc|ccccc}
G_{00} & G_{01} & G_{0K+1'} & & \cdots & & G_{0n} \\
G_{10} & G_{11} & G_{1K+1'} & & \cdots & & G_{1n} \\
\hline
- & 0\ldots0 & \bullet & + & \ldots & + & + \\
0 & & - & \bullet & & + & + \\
\vdots & \vdots & 0 & - & \ddots & + & + \\
 & & & 0 & - & \bullet & + \\
0 & 0\ldots0 & \ldots & & 0 & - & \bullet
\end{array}\right]$$

Figure 6: Form of transition matrix after aggregation and putting bounds in rates.

$$\begin{bmatrix} Q_{D'D'} & Q_{D'D} & Q_{D'A} \\ Q_{DD'} & Q_{DD} & Q_{DA} \\ 0 & Q_{AD} & Q_{AA} \end{bmatrix}$$

Figure 7: Initial matrix, G.

$$\begin{bmatrix} Q_{D'D'} & Q_{D'D} & 0 & Q_{D'A} \\ Q_{DD'} & Q_{DD} & 0 & Q_{DA} \\ Q_{DD'} & 0 & Q_{CC} & Q_{DA} \\ 0 & 0 & Q_{AD} & Q_{AA} \end{bmatrix}$$

Figure 8: Introduction of "clone" states. Matrix G_1.

G_2 (see Figure 9) is formed from G_1 by exact aggregation of the states in $\mathcal{D}'$. We will refer to the single state which replaces $\mathcal{D}'$ as d'. Since exact aggregation is assumed we have that $\pi_{D/G_2} = \pi_{D/G_1}$.

G_3 (see Figure 10) is exactly the same as G_2 except that the transitions from d' to states in $\mathcal{D}$ and $\mathcal{C}$ are modified. In G_3 the submatrices $R'_{d'D}$ and $R'_{d'C}$ are required to have non-negative elements and to be such that:

$$R'_{d'D} + R'_{d'C} = R_{d'D} \tag{3}$$

A probabilistic interpretation is that the original transitions from d' to states in $\mathcal{D}$ are each 'split' so that part remains to the state in D and part goes to the corresponding "clone" state in C.

From the construction of G_3 it is easy to show that if

$[\pi_{d'/G_2}, \pi_{D/G_2}, \pi_{C/G_2}, \pi_{A/G_2}]$ is the solution of $\pi G_2 = 0$, $\sum \pi_{G_2}(i) = 1$ and

$[\pi_{d'/G_3}, \pi_{D/G_3}, \pi_{C/G_3}, \pi_{A/G_3}]$ is the solution of $\pi G_3 = 0$, $\sum \pi_{G_3}(i) = 1$ then

$$\begin{bmatrix} \bullet & R_{d'D} & 0 & R_{d'A} \\ Q_{Dd'} & Q_{DD} & 0 & Q_{DA} \\ Q_{Dd'} & 0 & Q_{CC} & Q_{DA} \\ 0 & 0 & Q_{AD} & Q_{AA} \end{bmatrix}$$

Figure 9: After exact aggregation of the states in $\mathcal{D}'$. Matrix G_2.

$$\begin{bmatrix} \bullet & R'_{d'D} & R'_{d'C} & R_{d'A} \\ Q_{Dd'} & Q_{DD} & 0 & Q_{DA} \\ Q_{Dd'} & 0 & Q_{CC} & Q_{DA} \\ 0 & 0 & Q_{AD} & Q_{AA} \end{bmatrix}$$

Figure 10: Modified rates from state d'. Matrix G_3.

$\pi_{d'/G_3} = \pi_{d'/G_2}$;

$\pi_{D/G_3} + \pi_{C/G_3} = \pi_{D/G_2} + \pi_{C/G_2}$; and

$\pi_{A/G_3} = \pi_{A/G_2}$.

The result that we really want is expressed in the following theorem.

Theorem 3.1 $\pi_{D/G_3} \leq \pi_{D/G_2}$

Proof: The proof is given in the [11]. □ .

Now we consider aggregation of the subsets of states in $\mathcal{C}$ and $\mathcal{A}$. By definition :

$\mathcal{D} = \bigcup_{j=L_i}^{H_i} \mathcal{F}_j$; $\qquad \mathcal{C} = \bigcup_{j=L_i}^{H_i} \mathcal{F}'_j$; $\qquad$ and $\qquad \mathcal{A} = \bigcup_{j=H_i+1}^{N} \mathcal{F}_j$.

where L_i and H_i are integers associated with the i^{th} step and denote the minimum and maximum number of failed components for states in $\mathcal{D}$.

We form one aggregate state for each subset $\mathcal{F}'_j$ in $\mathcal{C}$ and $\mathcal{F}_j$ in $\mathcal{A}$. The resulting matrix G_4, is shown in Figure 11. In this figure we have also interchanged the ordering of the state d' and the set of states $\mathcal{D}$. Since we have assumed exact aggregation in forming G_4 it is clear that

$\pi_{D/G_4} = \pi_{D/G_3}$.

Comparing the matrix in Figure 6 with the matrix G_4 in Figure 11 we note that they have the same form. Therefore the result quoted in the Section 2 applies. Specifically, if the elements shown in Figure 12 as '$+$' are replaced by upper bounds on those rates and the elements shown as '$-$' are replaced by lower bounds then the solution for the stationary state probabilities will yield a lower bound for the state probabilities for states in D, i.e. $\pi_{D/G_5} \leq \pi_{D/G_4}$.

In summary, $\pi_{D/G_5} \leq \pi_{D/G}$, and clearly G_5 has a much reduced states space compared to that of G. A remaining issue is the calculation of the upper and lower bounds on the transition rates required for G_5. This issue is addressed and a detailed specification of the multi-step algorithm is given in the following section.

$$\left[\begin{array}{cc|ccccc|ccccc}
Q_{D,D} & Q_{D,d'} & 0 & 0 & \cdots & \cdots & 0 & Q_{D,A_1} & Q_{D,A_2} & \cdots & \cdots & Q_{D,A_n} \\
R'_{d',D} & \bullet & r_{d',C_1} & r_{d',C_2} & \cdots & \cdots & r_{d',C_J} & r_{d',A_1} & r_{d',A_2} & \cdots & \cdots & r_{d',A_n} \\
\hline
0\ldots0 & r_{C_1,D'} & \bullet & r_{C_1,C_2} & \cdots & \cdots & r_{C_1,C_j} & r_{C_1,A_1} & r_{C_1,A_2} & \cdots & \cdots & r_{C_1,A_n} \\
0\ldots0 & 0 & r_{C_2,C_1} & \bullet & \cdots & \cdots & r_{C_2,C_j} & r_{C_2,A_1} & r_{C_2,A_2} & \cdots & \cdots & r_{C_2,A_n} \\
\vdots & \vdots & 0 & r_{C_3,C_2} & \bullet & \cdots & r_{C_3,C_4} & r_{C_3,A_1} & \cdots & \cdots & \cdots & r_{C_2,A_n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0\ldots0 & \vdots & 0 & \cdots & \cdots 0 & r_{C_J,C_{J-1}} & \bullet & r_{C_J,A_1} & \cdots & \cdots & \cdots & r_{C_J,A_n} \\
\hline
0\ldots0 & 0 & 0 & 0 & 0 & \cdots & r_{A_1,C_J} & \bullet & r_{A_1,A_2} & \cdots & \cdots & r_{A_1,A_n} \\
\vdots & 0 & 0 & 0 & 0 & \cdots & 0 & r_{A_2,A_1} & \bullet & \cdots & \cdots & r_{A_2,A_n} \\
\vdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & r_{A_3,A_2} & \bullet & \cdots & r_{A_3,A_n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \ddots & \vdots \\
0\ldots0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & r_{A_n,A_{n-1}} & \bullet
\end{array}\right]$$

Figure 11: Aggregation of states in $\mathcal{C}$ and $\mathcal{A}$. Matrix G_4

3.1 Description of the Algorithm

The bounds on the transition rates in G_5 are of three types. The rates denoted by '$+$' in Figure 12 are to be replaced by upper bounds. A simple upper bound is easily seen to be the sum of the failure rates of all components. Similarly, the rates denoted by '$-$' are to be replaced by lower bounds. A simple lower bound that suffices is the minimum repair rate of all components. However, as described below, we can do munch better.

Based on the construction of the previous section, submatrix $R'_{d'D}$ in G_5 is required to contain lower bounds on the conditional rates from d' to states in $\mathcal{D}$. If $\pi_{D'/G}$ is the vector of stationary state probabilities for states in $\mathcal{D}$ then $(\pi_{D'/G}/\sum_{i\in D'}\pi_{D'/G}(i))\ Q_{D'D}$ is the vector of conditional transition rates to states in $\mathcal{D}$. It follows immediately that if

$\pi'_{D'} \leq \pi_{D'/G}$ then

$\pi'_{D'}Q_{D'D} \leq (\pi_{D'/G}/\sum \pi_{D'/G}(i))Q_{D'D}$

Thus we have the needed lower bound provided only that a lower bound on $\pi_{D'/G}$ is available.

In the multi-step procedure we describe next, at each step the set of states $\mathcal{D}'$ corresponds exactly to the set of states for which a lower bound has been found on previous steps. Therefore these lower bound stationary state probabilities provide the needed data for the following step.

The multi-step bounding procedure is as follows:

$$\left[\begin{array}{cc|ccccc|ccccc}
Q_{D,D} & Q_{D,d'} & 0 & 0 & \cdots & \cdots & 0 & Q_{D,A_1} & Q_{D,A_2} & \cdots & \cdots & Q_{D,A_n} \\
R'_{d',D} & \bullet & + & + & \cdots & \cdots & + & + & + & \cdots & \cdots & + \\
\hline
0\ldots0 & - & \bullet & + & \cdots & \cdots & + & + & + & \cdots & \cdots & + \\
0\ldots0 & 0 & - & \bullet & \cdots & \cdots & + & + & + & \cdots & \cdots & + \\
\vdots & \vdots & 0 & - & \bullet & \cdots & + & + & \cdots & \cdots & \cdots & + \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0\ldots0 & \vdots & 0 & \cdots & \cdots 0 & - & \bullet & + & \cdots & \cdots & \cdots & + \\
\hline
0\ldots0 & 0 & 0 & 0 & 0 & \cdots & - & \bullet & + & \cdots & \cdots & + \\
\vdots & 0 & 0 & 0 & 0 & \cdots & 0 & - & \bullet & \cdots & \cdots & + \\
\vdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & - & \bullet & \cdots & + \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \ddots & \vdots \\
0\ldots0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & - & \bullet
\end{array}\right]$$

Figure 12: Replacement of transition rates with bounds. Matrix G_5.

1. (Step 1) Generate lower bounds on the stationary state probabilities for states $\mathcal{F}_0$ through $\mathcal{F}_{H_1}$ using the results from [14]. This provides lower bounds on the state probabilities of these states as well as an initial set of bounds on steady state availability. If the bounds are tight enough then terminate.

2. (Step $i \geq 2$) Generate the portion of G corresponding to $\mathcal{F}_{L_i}$ through $\mathcal{F}_{H_i}$. Construct the matrix corresponding to G_5 described in the previous section. The submatrices Q_{DD} and $Q_{Dd'}$ are generated from the model definition. Let $\pi'_{D'}$ be the vector of lower bounds on the state probabilities computed from previous steps. Now set the submatrix $R'_{d'D}$ equal to $\pi'_{D'}Q_{D'D}$ and solve for π_{D/G_5} which provides lower bounds on the state probabilities for states in $\mathcal{D}$. Compute upper and lower bounds on steady state availability using Equation 1 from Section 2. If the bounds are tight enough then terminate, else repeat step 2.

4 Error-reduction.

In the previous two sections, we have described the one-step and the multi-step procedures for bounding the steady state availability of a repairable computer system. In these bounding procedures, errors accumulate at each step. These errors can be classified according to their source :

1. by considering the clone states and the aggregate states to have reward of 0 or 1 in the evaluation of the availability bounds.

2. the difference between the lower bounds and the actual stationary state probabilities of the "detailed states" in $\mathcal{D}$.

These errors cannot be completely eliminated unless the complete transition rate matrix is generated and solved (which is what we originally tried to avoid), but we can *reduce* these errors and obtain a tighter availability bound. The error reduction process we propose is iterative in nature. We will first reduce the error in the clone states by finding their lower bound state probabilities. Once we obtain these lower bound state probabilities, an improved estimate of the transition rates out of the aggregate clone states can be computed. With these improved transition rates, we can reduce the error in the stationary state probabilities of the "detailed states" $\mathcal{D}$. An important point is that the error reduction process does not require generating more of the transition matrix. In section 4.1, we will present the approach to reduce the error associated with the clone states and in section 4.2, we will present the approach to obtain improved lower bound state probabilities for the "detailed states", $\mathcal{D}$.

4.1 Error reduction for the clone states.

Until now, we have assigned a reward of 0 or 1 to each clone state in the computation of the availability bound. To reduce this source of error, we would like to obtain a lower bound for the state probability of each individual clone state. We will make use of the fact that the clone states have exactly the same transition structure as the detail states $\mathcal{D}$. As we will show, the portion of transition rate matrix for the detail states, which was generated in the previous step, can be used to compute an improved lower bound on the state probabilities of the individual clone states.

Assume that in the previous step of the multi-step procedure, we have already obtained lower bounds on the state probabilities for all detail states that have $m+1$ failed components and we want to compute the state probabilities for all clone states that have between k and m failures. Figure 13 depicts the situation. Let us define the following notation :

Let us define the following :

${}^{c}\pi_j(i)$	$=$	probability of the i^{th} clone state among those clone states in C_j.
$\lvert C_j \rvert$	$=$	total number of clone states in C_j.
S_j	$=$	set of states with j failures, it is equal to $C_j \cup F_j$.
$\lvert S_j \rvert$	$=$	total number of states in S_j.
$g_{(j,l),(i,m)}$	$=$	transition rate from the j^{th} state in C_l to the i^{th} state in C_m.
$g_{(j,l),(*,*)}$	$=$	transition rate out of the j^{th} state in C_l.

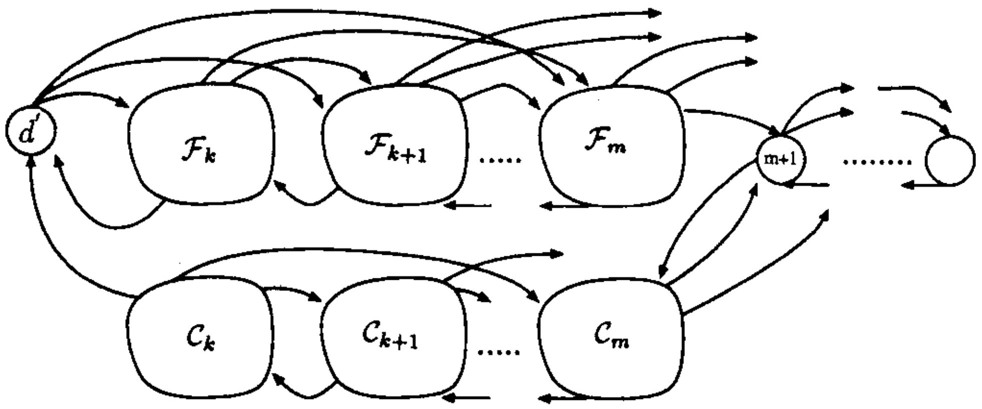

Figure 13: State Transition at $k+1^{th}$ step

We can write the flow equations for the clone states in C_m as follow :

$$
\begin{aligned}
{}^c\pi_m(i) \quad = \quad & \left\{ \sum_{j=1}^{|C_k|} {}^c\pi_k(j) g_{(j,k),(i,m)} + \cdots + \sum_{j=1}^{|C_{m-1}|} {}^c\pi_{m-1}(j) g_{(j,m-1),(i,m)} \right. \\
& \left. + \sum_{j=1}^{|S_{m+1}|} \pi_{m+1}(j) g_{(j,m+1),(i,m)} \right\} \Big/ g_{(i,m),(*,*)} \qquad for \;\; i = 1, \ldots, |C_m|. \qquad (4)
\end{aligned}
$$

For clone states in C_l where $k \; < \; l \; < m$, we have :

$$
\begin{aligned}
{}^c\pi_l(i) \quad = \quad & \left\{ \sum_{j=1}^{|C_k|} {}^c\pi_k(j) g_{(j,k),(i,l)} + \cdots + \sum_{j=1}^{|C_{l-1}|} {}^c\pi_{l-1}(j) g_{(j,l-1),(i,l)} \right. \\
& \left. + \sum_{j=1}^{|C_{l+1}|} {}^c\pi_{l+1}(j) g_{(j,l+1),(i,l)} \right\} \Big/ g_{(i,l),(*,*)} \qquad for \;\; i = 1, \ldots, |C_l|. \qquad (5)
\end{aligned}
$$

And for the clone states in C_k, we have:

$$
{}^c\pi_k(i) \quad = \quad \left\{ \sum_{j=1}^{|C_{k+1}|} {}^c\pi_{k+1}(j) g_{(j,k+1),(i,k)} \right\} \Big/ g_{(i,k),(*,*)} \qquad for \;\; i = 1, \ldots, |C_k|. \qquad (6)
$$

Assume that from the previous step, we have obtained lower bounds on the state probabilities $\pi_{m+1}(i)$ and also the transition structure for states that are between k and m failures. We can then apply the Gauss-Seidel Iterative method for obtaining a lower bound on the state probabilities of the clone states. Let the initial values be :

$$
{}^c\pi_j^{(0)}(i) \quad = \quad 0 \qquad\qquad for \;\; j = k, \ldots, m \; and \;\; i = 1, \ldots, |C_j|. \qquad (7)
$$

Note that $\pi_{m+1}(i)$ are constants in this algorithm(the lower bound state probabilities obtained from the previous bounding step). We can rewrite the flow equations into the

Gauss-Seidel form as follows:

$$\begin{aligned}{}^{c}\pi_m^{(n)}(i) &= \left\{ \sum_{j=1}^{|C_k|} {}^{c}\pi_k^{(n-1)}(j) g_{(j,k),(i,m)} + \cdots + \sum_{j=1}^{|C_{m-1}|} {}^{c}\pi_{m-1}^{(n-1)}(j) g_{(j,m-1),(i,m)} \right. \\ &\quad \left. + \sum_{j=1}^{|S_{m+1}|} \pi_{m+1}^{(n-1)}(j) g_{(j,m+1),(i,m)} \right\} \Big/ g_{(i,m),(*,*)} \qquad for \;\; i = 1, \ldots, |C_m|. \quad (8)\end{aligned}$$

for the clone states in C_l where $k < l < m$, we have :

$$\begin{aligned}{}^{c}\pi_l^{(n)}(i) &= \left\{ \sum_{j=1}^{|C_k|} {}^{c}\pi_k^{(n-1)}(j) g_{(j,k),(i,l)} + \cdots + \sum_{j=1}^{|C_{l-1}|} {}^{c}\pi_{l-1}^{(n-1)}(j) g_{(j,l-1),(i,l)} \right. \\ &\quad \left. + \sum_{j=1}^{|C_{l+1}|} {}^{c}\pi_{l+1}^{(n)}(j) g_{(j,l+1),(i,l)} \right\} \Big/ g_{(i,l),(*,*)} \qquad for \;\; i = 1, \ldots, |C_l|. \quad (9)\end{aligned}$$

And for the states in C_k, we have:

$${}^{c}\pi_k^{(n)}(i) = \left\{ \sum_{j=1}^{|C_{k+1}|} {}^{c}\pi_{k+1}^{(n)}(j) g_{(j,k+1),(i,k)} \right\} \Big/ g_{(i,k),(*,*)} \qquad for \;\; i = 1, \ldots, |C_k|. \quad (10)$$

The iterative procedure defined by these equations has the following characteristics :

1. it *converges* and converges to a *unique* solution.
2. it will converge from below.
3. it converges monotonically.
4. the solution (fixed point) is a lower bound of the exact state probabilities of the clone states.

The second and third characteristics are especially interesting because they indicate that we can terminate the iterative process anytime and still obtain lower bounds on the state probabilities of the clone states. To show the above characteristics hold, we rewrite the iterative equations into matrix form. Define :

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_m} \\ \mathbf{x_{m-1}} \\ \mathbf{x_{m-2}} \\ \mathbf{x_{m-3}} \\ \vdots \\ \mathbf{x_k} \end{bmatrix} \quad and \quad \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{x_i}$ is a column vector for the steady state probabilities for clone states in C_i and $\mathbf{c}$ is a column vector with components $c_i = \sum_{j=1}^{|S_{m+1}|} \pi_{m+1}(j) g_{(j,m+1),(i,m)}$ for all clone states in C_m. The flow equations can be written in matrix form as :

$$\begin{bmatrix} Q_{m,m} & Q_{m-1,m} & \cdots & Q_{k+1,m} & Q_{k,m} \\ Q_{m,m-1} & Q_{m-1,m-1} & \cdots & Q_{k+1,m-1} & Q_{k,m-1} \\ 0 & Q_{m-1,m-2} & \cdots & Q_{k+1,m-2} & Q_{k,m-2} \\ 0 & 0 & \cdots & Q_{k+1,m-3} & Q_{k,m-3} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & Q_{k+1,k} & Q_{k,k} \end{bmatrix} \begin{bmatrix} \mathbf{x_m} \\ \mathbf{x_{m-1}} \\ \mathbf{x_{m-2}} \\ \mathbf{x_{m-3}} \\ \vdots \\ \mathbf{x_k} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

where $Q_{i,i}$ is a diagonal matrix with each element being the rate out from a clone state in C_i, $Q_{i,j}$ is a matrix with its elements being the "negated" transition rates from clone states in C_i to clone states in C_j. Let this transition matrix be $\mathbf{A}$ and in matrix notation, we have :

$$\mathbf{A\ x} \ = \ \mathbf{b} \tag{11}$$

Let $\mathbf{A} = [\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A} - \mathbf{U}_\mathrm{A}]$ where $\mathbf{D}_\mathrm{A}$ is a diagonal matrix and $\mathbf{L}_\mathrm{A}$ and $\mathbf{U}_\mathrm{A}$ are the lower and upper triangular matrix respectively. The Gauss-Seidel iteration can be written as :

$$\mathbf{x}^{(k)} \ = \ [(\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{U}_\mathrm{A}]\mathbf{x}^{(k-1)} \ + \ (\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{b} \tag{12}$$

A necessary and sufficient condition for the above iterative process to converge to an unique solution is for $\rho[(\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{U}_\mathrm{A}]$, the spectral radius of the matrix, be less than 1 [1]. To prove the characteristics of our algorithm, we need the following definition and theorem from [21].

Definition. For n x n real matrices $\mathbf{A}$, $\mathbf{M}$ and $\mathbf{N}$, where $\mathbf{A} = \mathbf{M} - \mathbf{N}$. $(\mathbf{M} - \mathbf{N})$ is a regular splitting of the matrix $\mathbf{A}$ if $\mathbf{M}$ is nonsingular with $\mathbf{M}^{-1} \geq \mathbf{0}$ and $\mathbf{N} \geq \mathbf{0}$.

Theorem 4.1 *If* $\mathbf{A} = \mathbf{M} - \mathbf{N}$ *is a regular splitting of the matrix* $\mathbf{A}$ *and* $\mathbf{A}^{-1} \geq \mathbf{0}$, *then*

$$\rho[\mathbf{M}^{-1}\mathbf{N}] \ < \ 1$$

We proceed by showing that Theorem 4.1 can be applied to $\mathbf{A}$. The first step is to show that $\mathbf{A}^{-1} \geq \mathbf{0}$.

Lemma 4.1 $\mathbf{A}$ *is nonsingular and* $\mathbf{A}^{-1} \geq \mathbf{0}$.

Proof : Let $\mathbf{B} = \mathbf{A}^\mathrm{T}$. Then $\mathbf{B} = (b_{i,j})$ is a n x n matrix where $b_{i,i} > 0$ for all i, $b_{i,j} \leq 0$ for all $i \neq j$, $b_{i,i} \geq \sum_j |b_{i,j}|$ for all i and strict inequality holds for at least one value of i. Let $\mathbf{D}_\mathrm{B}$ be the diagonal matrix of $\mathbf{B}$. Since $\mathbf{D}_\mathrm{B}$ is nonsigular, we can express :

$$\mathbf{C} \ = \ \mathbf{I} - \mathbf{D}_\mathrm{B}^{-1}\mathbf{B}$$

Now $\mathbf{C}$ is a substochastic matrix with $\rho[\mathbf{C}]$ less than 1. From Theorem 3.10 in [21], $\mathbf{B}^{-1} \geq \mathbf{0}$. Since $(\mathbf{A}^\mathrm{T})^{-1} \geq \mathbf{0}$, therefore $\mathbf{A}^{-1} \geq \mathbf{0}$. □

Lemma 4.2 $(\mathbf{D}_A - \mathbf{L}_A)$ *is nonsingular and* $(\mathbf{D}_A - \mathbf{L}_A)^{-1} \geq \mathbf{0}$.

Proof : same as above lemma. □

Lemma 4.3 *The proposed iterative procedure converges to a unique solution.*

Proof : $\mathbf{A}$ and $(\mathbf{D}_A - \mathbf{L}_A)$ are nonsingular and their inverses are nonnegative matrices. From the construction, $\mathbf{U}_A$ is also a nonnegative matrix. Since $\mathbf{A} = (\mathbf{D}_A - \mathbf{L}_A) - \mathbf{U}_A$, it follows that $(\mathbf{D}_A - \mathbf{L}_A)$ and $\mathbf{U}_A$ is a *regular splitting* of the matrix $\mathbf{A}$. From Theorem 4.1, the spectral radius, $\rho[(\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{U}_A] < 1$. □

Lemma 4.4 *The proposed iterative procedure will always converge from below.*

Proof : Since the iterative process converges to a unique solution $\mathbf{x}^*$. from (12), we have :

$$\mathbf{x}^* = \left[(\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{U}_A\right] \mathbf{x}^* + (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{b}$$

Let $\mathbf{e}^{(k)}$ be the error vector at k^{th} iteration. Then :

$$\begin{aligned} \mathbf{e}^{(k)} &= \mathbf{x}^* - \mathbf{x}^{(k)} \\ &= (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{U}_A(\mathbf{x}^* - \mathbf{x}^{(k-1)}) \\ &= (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{U}_A(\mathbf{e}^{(k-1)}) \end{aligned}$$

In order to have the solution approach from below, we need $\mathbf{e}^{(k)} \geq \mathbf{0}$ for all k. Since $(\mathbf{D}_A - \mathbf{L}_A)^{-1}$ and $\mathbf{U}_A$ are nonnegative matrices, if we start with a lower bound vector, then we always approach the solution from below. □

Lemma 4.5 *The proposed iterative procedure converges monotonically.*

Proof : To show that we have an improved bound at each iteration, we have to show $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \geq \mathbf{0}$ for all k. This can be easily proved by induction.

Basis : For $k = 0$. From (7), we see that $\mathbf{x}^{(0)} = \mathbf{0}$. From (12), we see that $\mathbf{x}^{(1)} = (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{b} \geq \mathbf{0}$. Therefore $\mathbf{x}^{(1)} - \mathbf{x}^{(0)} \geq \mathbf{0}$.

Induction : Assume $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \geq \mathbf{0}$ for $k \leq n$. For $k = n+1$, we have:

$$\mathbf{x}^{(n+2)} - \mathbf{x}^{(n+1)} = (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{U}_A(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) \geq \mathbf{0}$$

the inequality holds because $(\mathbf{D}_A - \mathbf{L}_A)^{-1}$ and $\mathbf{U}_A$ are nonnegative matrices. □

Lemma 4.6 *The solution of the proposed procedure is a lower bound on the exact state probabilities of the clone states.*

Proof : Let $\mathbf{x}'$ be the exact state probabilities vector for the clone states. We have to show that $\mathbf{x}' - \mathbf{x}^* \geq \mathbf{0}$. Let $\mathbf{b}'$ contains the *exact* rates into the clone states from states with $m+1$ failed components. Since :

$$\begin{aligned} \mathbf{A}\ \mathbf{x}^* &= \mathbf{b} \\ \mathbf{A}\ \mathbf{x}' &= \mathbf{b}' \\ \mathbf{b}' - \mathbf{b} &\geq \mathbf{0} \end{aligned}$$

The above inequality holds because in the exact model, we have the exact state probabilities for $\pi_{m+1}(j)$. It is easily seen that :

$$\begin{aligned} \mathbf{x}' - \mathbf{x}^* &= \mathbf{b}'\mathbf{A}^{-1} \ - \ \mathbf{b}\mathbf{A}^{-1} \\ &= (\mathbf{b}' - \mathbf{b})\mathbf{A}^{-1} \qquad \geq \mathbf{0} \end{aligned}$$

Since $(\mathbf{b}' - \mathbf{b}) \geq \mathbf{0}$ and $\mathbf{A}^{-1} \geq \mathbf{0}$. □

The algorithm for error reduction for the clone states is stated as follow :

1. let the probabilities for all clone states be zero.
2. apply the Gauss-Seidel iteration until a specified tolerance is satisfied.

4.2 Error reduction for the detail states.

Recall that in computing lower bounds for the state probabilities of the detail states D , we used upper bound failure rates (e.g: sum of the failure rates) and lower bound repair rates (e.g: minimum repair rate) for the aggregates. Using the procedure described in Section 4.1, we can obtain better lower bound estimates of the state probabilities for the clone states. These in turn can be used to generate tighter bounds on the transition rates and thereby obtain improved lower bound state probabilities for the detail states.

Let us define the following notation :

$r(k,l)$	transition rate from the clone aggregate corresponding to C_k to the clone aggregate corresponding to C_l .
$({}^c\pi_k(i))_{lb}$	lower bound state probability for the i^{th} state in C_k.
$\overline{\pi_k}$	sum of all computed lower bound state probabilities except the clone states in C_k.
$\sum_{(i \in k)}$	indicates sum over all states i in C_k.

Since we already computed a lower bound state probability for each clone state, the *improved* lower bound on the "repair rates" between the aggregates is as follow :

$$r(k,k-1)^- \ = \ MAX\left\{ minimum\ repair\ rate,\ \frac{\sum_{(i\in k)}\sum_{(l\in k-1)}({}^c\pi_k(i))_{lb}\ \ g_{(i,k),(l,k-1)}}{1.0\ -\ \overline{\pi_k}} \right\} \tag{13}$$

while the *improved* upper bounds on the failure rate from aggregate k to aggregates m is as follow :

$$r(k,m)^{+} = MIN\left\{ \text{sum of all failure rates}, \frac{\sum_{(i\in k)}\sum_{(j\in m)}\left[({}^{c}\pi_k(i))_{lb} + (1.0 - \overline{\pi_k} - \sum_{(i\in k)}[{}^{c}\pi_k(i)]_{lb})\right] g_{(i,k),(j,m)}}{\sum_{(i\in k)}({}^{c}\pi_k(i))_{lb}} \right\} \tag{14}$$

Equations (13) and (14) follow easily from considering the conditional transition rates between aggregates based on upper and lower bounds conditional state probabilities. For example in equation (13):

$$\frac{({}^{c}\pi_k(i))_{lb}}{(1.0 - \overline{\pi_k})} \tag{15}$$

is a lower bound on the conditional state probability for the i^{th} state in C_k.

4.3 Reduction strategy.

The above error reduction procedure will give us better lower bound state probabilities for detailed and clone states corresponding to *step* k of the bounding process. We can repeat the reduction procedure for detail states and clone states corresponding to steps $k-1$, $k-2$, $\ldots$, 1 of the bounding process. Since we obtain a better bound for all states in bounding step 1, we can go forward again and apply the multi-step bounding procedure to obtain better state probability bounds for states corresponding to bounding steps $2, 3, \ldots, k$. Therefore, the error reduction strategy is :

1. start at bounding step k, reduce the errors in the clone states and then the detail states, then go to bounding step $k-1$, repeat the reduction process until bounding step 1.
2. repeat the multi-step bounding procedure until step k to obtain better lower bounds for the detail states in bounding step $2, 3, \ldots, k+1$.

Clearly this reduction strategy can be applied repeatedly to obtain better availability bounds. Also, each time we apply the iterative procedure to refine the clone states, the starting vector for the clone states can be the fixed point solution vector from the previous iterative procedure. This way we not only minimize the number of iterations but also preserving the characteristics we claimed in Lemma 4.3 to Lemma 4.6. The presevation of the characteristics in Lemma 4.3, 4.4 and 4.6 is trivial. In the following lemma, we show that using the fixed point solution vector from the previous iterative procedure, we preserve the monotonic convergence characteristic.

Let us define the following :

$\mathbf{b}$ = be the vector of conditional rates from $\mathcal{F}_{m+1}$ to $\mathcal{C}_m$ computed by lower bound state probabilities $\pi_{m+1}(i)$.

$\pi'_{m+1}(i)$ = be a new lower bound state probabilites computed in the last step of the reduction strategy, where $\pi'_{m+1}(i) \geq \pi_{m+1}(i) \quad \forall \ i$.

$\mathbf{b}_2$ = be the vector of conditional rates from $\mathcal{F}_{m+1}$ to $\mathcal{C}_m$ computed by $\pi'_{m+1}(i)$.

Lemma 4.7 *Using the fixed point solution vector from the previous iterative procedure, the monotonic convergence characteristic is preserved.*

Proof : According to the last step of the reduction strategy, we obtained a better lower bounds for the detailed states in bounding step $k+1$, This implies that we have a better lower bounds state probabilities for $m+1$ failed components, $\pi'_{m+1}(i)$. Based on these better lower bounds, we can obtain a new vector $\mathbf{b}_2$ which has the following properties :

$$\begin{aligned} \mathbf{b}_2 &\geq \mathbf{b} \\ \mathbf{b}_2 &= \mathbf{b} + \mathbf{b}'_2 \qquad \textit{where} \quad \mathbf{b}'_2 \geq \mathbf{0} \end{aligned}$$

Similar to lemma 4.5, we have to show $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \geq \mathbf{0}$ for all k. Again, we proof this by induction. Let $\mathbf{x}^{(0)} = \mathbf{x}^*$ where $\mathbf{x}^*$ is the fixed point solution from the previous iterative procedure.

Basis : for $k=0$. From (12), we have :

$$\begin{aligned} \mathbf{x}^{(1)} &= \left[(\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{U}_A\right] \ \mathbf{x}^{(0)} + (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{b}_2 \\ &= \left[(\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{U}_A\right] \ \mathbf{x}^{(*)} + (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{b} + (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{b}'_2 \\ &= \mathbf{x}^{(*)} + (\mathbf{D}_A - \mathbf{L}_A)^{-1}\mathbf{b}'_2 \end{aligned}$$

Since $\mathbf{b}'_2 \geq \mathbf{0}$ and $(\mathbf{D}_A - \mathbf{L}_A)^{-1} \geq \mathbf{0}$, therefore $\mathbf{x}^{(1)} - \mathbf{x}^{(*)} \geq \mathbf{0}$.

Induction : The induction step is similar with the induction step in Lemma 4.5. □

Although this reduction strategy can be applied repeatedly, but there is a diminishing return of refining the errors in the clone states and the detailed states. One natural question we have to ask ourselves is when is it better to repeat the error reduction procedure and when it is better to generate more of the transition rate matrix corresponding to unexplored states. We provide a heuristic answer to this question in the following section.

5 Decision for backward iteration or forward generation

Although we will be able to obtain a tighter availability bounds by either going forward (i.e., by generating more of the matrix) or going backward (i.e., by reducing the errors accumulated in the previous steps), the computational cost and potential gain for these two choices are quite different. For the forward algorithm, we have to consider the following :

- computational cost for state generation.
- storage cost for the newly generated transition matrix.
- computational cost for evaluating steady state probabilities in the detail states.

For the backward algorithm, the transition matrices were generated in the previous steps, therefore the only cost is the computation cost of the error reduction process.

In order to decide between the forward and backward algorithm, we have to also compare their respective potential gains. We define the potential gain as the improvement in the spread between the upper and lower availability bounds. For the backward algorithm, the potential gain comes from the error reduction of the detail states and clone states. For the forward algorithm, the potential gain comes from the ability to obtain a better lower bound for the reward contributed by the newly generated states. Although we can apply the backward algorithm repeatedly to reduce the errors, the potential gain for each successive application exhibits diminishing returns : the potential gain for each application decreases geometrically. On the other hand, the forward algorithm will generate more states. But since the state probabilities of the system is skewed (states with more failures will have much smaller probabilities), therefore these newly generated state may not have a significant contribution to the system availability.

Based on the above discussion, we see that the problem of making an optimal decision for either applying forward or backward algorithm is not trivial. The decision algorithm should clearly be much less costly than the forward and backward algorithms themselves. Also, since we always obtain bounds regardless of the decision, there is no serious danger in using a heuristic that sometimes makes a suboptimal decision. The worst possible effect is some inefficiency. Next we will propose a simple heuristic algorithm for deciding the next step in obtaining bounds. The algorithm is as follow :

1. apply the backward algorithm (or the error reduction algorithm).
2. compute the gain (reduction in spread between the upper and lower bounds) we acquired from step 1.
3. if the potential gain is greater than the estimated state probability of the first unexplored aggregate, go to step 1, else apply forward algorithm.

In essence, the decision algorithm is biased toward reducing the accumulated errors from the previous steps. By doing this, it also avoids the state generation cost and the storage cost of the forward algorithm. It will apply forward algorithm only when the projected potential gain is less than the reward of the unexplored states.

Lastly, we define the global bounding algorithm. One important note is that the algorithm can be terminated at any phase depending on the user requirement, this is due to the fact that at all phases, we are providing bounds to the system availability. The global algorithm is :

1. apply one step algorithm. If the bound is satisfied, stop.
2. apply multi-step algorithm. If the bound is satisfied, stop.
3. apply the error refinement algorithm, if the bound is satisfied, stop.
4. apply the decision algorithm, then either go to step 2 or 3.

6 Example.

In this section, we will present an example to illustrate the versatility of the bounding algorithm. The example incorporates the general bounding procedure and a tight bound is obtained. The example is a heterogeneous distributed database system as depicted in Figure 14. The components of this system are: two front-ends, four databases and four processing subsystems consisting of a switch, a memory and two processors. Components may fail and be repaired according to the rates given in Table 1. If the processors of system A and B fail, they will have a 0.05 probability of contaminating the Database A and a. If the processors of system C and D fail, they will have a 0.05 probability of contaminating the Database B and b. Components are repaired by a single repair facility which gives preemptive priority to components in the order: front-end, databases, switches, memories, processors set 1 and lastly processor set 2. (Ties are broken by random selection.) The database system is considered operational if the at least one front-end is operational, at least one database is operational, and at least one processing subsystem is operational. A processing subsystem is operational if the switch, the memory and at least one processor are operational. Also, this system is in *active breakdown mode*, meaning that components fail even when the system is non-operational.

In Table 2, we present the bounds on steady state availability. We note that for each step, the bounds of the availability are significantly tightened. In step one, we apply the one-step bounding algorithm with detail states that are between 0 to 2 failures. In step two, we apply the multi-step bounding algorithm with detail states that have between 3 to 4 failed components. In step three, we apply the backward/forward error reduction algorithm for states that are between 1 to 4 failures. In step four, we apply the multi-step bounding algorithm with detail states that are between 5 to 6 failures. In step five, we apply the backward/forward error reduction algorithm for states that are between 1 to 6 failures.

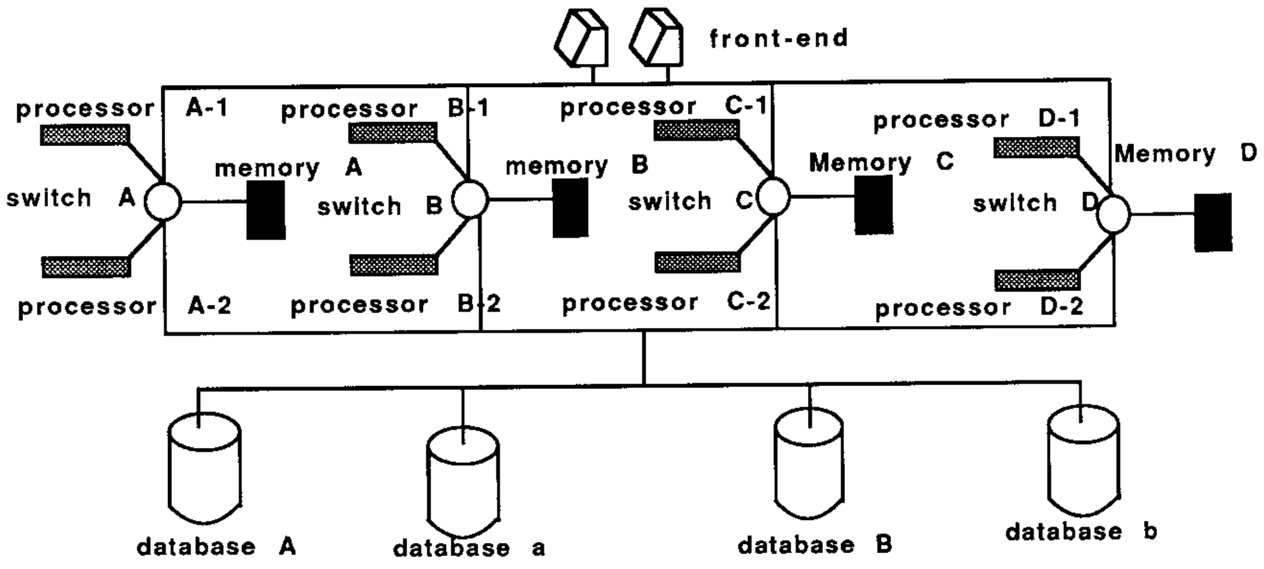

Figure 14: A fault-tolerant heterogeneous distributed database system.

7 Conclusion.

We have developed a methodology for computing bounds on the steady state availability of repairable computer systems. The method provides an efficient way to overcome the large state space problem in evaluating realistic computer systems. We showed that by modification of the original model, bounds can be obtained with much less cost and also state space cardinality was drastically reduced. Depending on the tightness required, the user can tradeoff tightness of the bounds with computational effort.

The development in the paper is couched in terms of models of repairable computer systems and determining bounds on availability. However the methods appear to have promise for other applications. The important property of availability models that was used was that the equilibrium state probabilities were concentrated in very few states. It is reasonable to expect that this same property will hold for example, in models of probabilistic protocol evauation [6,13] and load balancing. In this case of load balancing, there is presumably a policy for balancing the load on the resources in the system. Thus we expect that a large number of possible states of the system will have "small" probability since the scheduler will be biasing the system toward a small number of preferred states. Research into such applications is ongoing.

Components	Mean Failure Rate	Mean Repair Rate
Front-end A	1/4000	2.1
Front-end B	1/8000	2.0
Processor A-1	1/500	2.5
Processor A-2	1/400	2.0
Switch A	1/750	2.7
Memory A	1/750	2.5
Processor B-1	1/450	2.3
Processor B-2	1/450	1.8
Switch B	1/625	2.6
Memory B	1/750	2.4
Processor C-1	1/600	2.3
Processor C-2	1/450	1.7
Switch C	1/625	2.6
Memory C	1/600	2.4
Processor D-1	1/450	2.1
Processor D-2	1/450	1.5
Switch C	1/600	2.1
Memory C	1/600	2.5
Database A	1/5500	2.5
Database a	1/5000	2.2
Database B	1/5000	2.5
Database b	1/4500	2.3

Table 1: Failure and repair rates(per hour).

Step Number	Lower Bound	Upper Bound	Difference in Bounds
1	0.986456955373	0.999999999655	0.013543044282
2	0.990023127431	0.999999999029	0.009976869598
3	0.999995763421	0.999999987643	0.000004224222
4	0.999999246581	0.999999975211	0.000000728630
5	0.999999952345	0.999999952972	0.000000000627

Table 2: Upper and lower bounds on steady state availability of the database system.

References

[1] Richard L. Burden, J. Douglas Faires. *Numerical Analysis, PWS-KENT Publishing Company, 1988.*

[2] J.A. Carrasco, J. Figueras, *METFAC: Design and Implementation of a Software Toll for Modeling and Evaluation of Complex Fault-Tolerant Computing Systems, Proceedings of FTCS-16, pp. 424-429, July 1986.*

[3] A. Costes, J.E. Doucet, C. Landrault, and J.C. Laprie, *SURF: A Program for Dependability Evaluation of Complex Fault-Tolerant Computing Systems, Proceedings of FTCS-11, pp. 72-78, June 1981.*

[4] P. J. Courtois. *Decomposability — queueing and computer system approximation. Academic Press, New York, 1977.*

[5] E. de Souza e Silva and H.R. Gail, *Calculating Cumulative Operational Time Distributions of Repairable Computer Systems, IEEE Transactions on Computers (Special Issue on Fault Tolerant Computing), vol. C-35, no. 4, pp. 322-332, April 1986.*

[6] D.D. Dimitrijevic and M. Chen. *An Integrated Algorithm for Probabilistic Protocol Verification and Evaluation, IBM Tech. Report RC 13901, 1988.*

[7] R. Geist and K. S. Trivedi, *Ultra-High Reliability Prediction for Fault-Tolerant Computer Systems, IEEE Trans. Computer, C-32, 12, pp. 1118-1127, Dec. 1985.*

[8] A. Goyal, W.C. Carter, E. de Souza e Silva, S.S. Lavenberg and K.S. Trivedi, *The System Availability Estimator, Proceedings of the* 6th *Annual International Symposium on Fault-Tolerant Computing Systems (FTCS-16), Vienna, pp. 84-89, July 1986.*

[9] A. Goyal, S.S. Lavenberg and K.S. Trivedi. *Probabilistic Modeling of Computer System Availability, Ann. of Oper. Res., vol. 8, pp. 285-306, 1986.*

[10] P. Heidelberger and A. Goyal, *Sensitivity Analysis of Continuous Time Markov Chains Using Uniformization, Proceedings of the Second International Workshop on Applied Mathematics and Performance/Reliability Models of Computer/Communication Systems, Rome, Italy, May 1987.*

[11] John C.S. Lui, R. R. Muntz. *Evaluating Bounds on Steady State Availability from Markov Models of Repairable Systems, First International Workshop on the Numerical Solution of Markov Chains, 1990.*

[12] S.V. Makam, and A. Avizienis *ARIES 81: A Reliability and Life-Cycle Evaluation Tool for Fault Tolerant Systems, Proceedings of FTCS-12, pp. 276-274, June 1982.*

[13] N.F. Maxemchuk and K. Sabnani. *Probabilistic Verification of Communication Protocols, in Protocol Specification, Testing and Verification, VII, ed. H. Rubin and C.H. West, Elsevier, 1987, pp.307-320*

[14] R. R. Muntz,E. De Souza Silva, A. Goyal. *Bounding Availability of Repairable Computer Systems, SIGMETRICS 1989, pp. 29-38, also to appear in a special issue of IEEE-TC on performance evaluation, Dec. 1989.*

[15] M. F. Neuts. *Matrix-geometric solutions in Stochastic Models — an algorithmic approach. John Hopkins University Press, Baltimore, MD, 1981.*

[16] William J. Stewart, Ambuj Goyal. *Matrix Methods in Large Dependability Models. IBM Research Report, 11485, Nov 4, 1985.*

[17] Y. Takahashi. *Some Problems for Applications of Markov Chains, Ph.D. Thesis, Tokyo Institute of Technology. March 1972.*

[18] Y. Takahashi. *A lumping method for numerical calculations of stationary distributions of Markov chains, Research Reports on Information Sciences, Tokyo Institute of Technology, No.B-18, June 1975.*

[19] K.S. Trivedi, *Probability & Statistics with Reliability, Queuing and Computer Science Applications, Prentice Hall, 1982.*

[20] K.S. Trivedi, J.B. Dugan, R.R. Geist, and M.K. Smotherman, *Hybrid Reliability Modeling of Fault-Tolerant Computer Systems, Comput. Elec. Eng., Vol. 11, pp. 87-108, 1984.*

[21] Richard Varga. *Matrix Iterative Analysis. England Cliffs, N.J., Prentice-Halls, 1962.*