Computing Bounds on Steady State Availability of Repairable Computer Systems

JOHN C. S. LUI AND RICHARD R. MUNTZ

University of California at Los Angeles, Los Angeles, California

Abstract. One of the most important performance measures for computer system designers is system availability. Most often, Markov models are used in representing systems for dependability/availability analysis. Due to complex interactions between components and complex repair policies, the Markov model often has an irregular structure, and closed-form solutions are extremely difficulty to obtain. Also, a realistic system model often has an unmanageably large state space and it quickly becomes impractical to even generate the entire transition rate matrix. In this paper, we present a methodology that can (i) bound the system steady state availability and at the same time, (ii) drastically reduce the state space of the model that must be solved. The bounding algorithm is iterative and generates a part of the transition matrix at each step. At each step, tighter bounds on system availability are obtained. The algorithm also allows the size of the submodel, to be solved at each step, to be chosen so as to accommodate memory limitations. This general bounding methodology provides an efficient way to evaluate dependability models with very large state spaces without ever generating the entire transition rate matrix.

Categories and Subject Descriptors: D.4.8 [**Operating Systems**]: Performance—*stochastic analysis; reliability*

General Terms· Algorithms, Availability, Theory

Additional Key Words and Phrases: Bounds, Markov models, stationary probabilities

1. *Introduction*

Computer systems are widely used in many applications (e.g., air traffic control and banking applications) in which dependability is crucial, and system dependability analysis has long been an active area of research. Techniques, such as combinatoric analysis, Markov or semi-Markov analysis and simulation, have all been used in dependability analysis.[1] In recent years, tools based on these

[1] For Markov or semi-Markov analysis, see de Souza e Silva and Gail [1989], Geist and Trivedi [1985], Goyal et al. [1986], Heidelberger and Goyal [1987], and Trivedi [1982]. For simulation, see Conway and Goyal [1987] and Lewis and Bohm [1984].

This research was supported by the National Science Foundation (NSF) under contract CCR 92-15064.

Authors' address: 3277A Boelter Hall, Computer Science Department, University of California at Los Angeles, Los Angeles, CA 90024.

techniques have been built to aid system designers in evaluating and comparing different architectures during the design process.[2]

There are two major types of dependability measures that are of interest. The first type concerns transient measures, for example, distribution of the number of times the system fails in a certain mode by time t. This type of measure is especially appropriate for mission-oriented systems (e.g., spacecraft computers). The second type concerns steady state dependability measures, such as steady state availability. These measures are appropriate for systems with lifetimes that are long enough to span many failure and repair cycles (e.g., database management systems or telephone systems). Methods to solve transient dependability measures of repairable computer systems have been reported in [de Sousa e Silva and Gail, 1986; 1989]. In this paper, we concentrate on steady state availability.

Markov models are most widely used in analyzing system dependability because of their generality as well as their ability to represent complex interactions among components (e.g., dependent failure rates, complex repair policies, etc.). Because of these complex interactions between components, closed-form solutions are extremely difficult, if not impossible, to obtain. Therefore, numerical solution techniques are often used in analyzing the Markov model. One of the most pervasive limitations of numerical techniques is the inability to handle large models. For models of realistic systems, the state space requirements often vastly exceed the memory and the storage capacity of current (or future) systems [Stewart and Goyal, 1985]. Although approximations (e.g., state space truncation) can be applied to solve the cardinality problem, the errors introduced are difficult to quantify. In this paper, we present a methodology for computing steady state availability that can drastically reduce the state space cardinality and at the same time provide error bounds.

The results recently reported in [Lui and Muntz, 1991; Muntz et al., 1989] provide methods for computing bounds on the steady state availability of repairable systems. In [Muntz et al., 1989], a "one-step" algorithm is proposed that requires the user to make an *a priori* decision concerning the portion of the state transition matrix to be generated, and the steady state availability bounds are computed based on this submodel. In [Lui and Muntz, 1991], a "multi-step" algorithm is proposed that allows stepwise generation of portions of the state transition matrix, such that at each step, we can achieve progressively tighter bounds on the steady state availability.

It is important to note that although each successive application of the algorithm in [Lui and Muntz, 1991] can tighten the availability bounds, the spread between the bounds has a non-zero limiting value, that is, the bounds cannot be made arbitrarily tight by using the multi-step bounding algorithm. In this paper, we present a general bounding methodology that augments the previous results by providing an iterative procedure to refine the bounds to an arbitrary precision.

After any step in the iterative computation, we have generated lower bounds for the stationary state probabilities for a subset of the states of the model. The lower bounds on the state probabilities are used to compute upper and lower

[2] See Carrasco and Figueras [1986], Costes et al. [1981], Goyal et al. [1986], Makam and Avizienis [1982], and Trivedi et al. [1984].

bounds on the steady state availability. The algorithm we present in this paper can then proceed in two ways: (i) another set of states can be explored and lower bounds for their state probabilities can be computed or (ii) the lower bounds on the state probabilities which have been previously computed can be refined, that is, we can obtain improved lower bounds on these state probabilities. We refer to the former as a *forward* generation step and the latter as a *bound spread reduction* or *backward* step. A decision procedure to determine whether to apply a forward step or a backward step is introduced and discussed in a later section.

In Section 2, we introduce the model and the notation. We describe the "one-step" algorithm in Section 3. In Section 4, the "multi-step" bounding algorithm is presented. The bound spread reduction algorithm is presented in Section 5, and Section 6 describes a decision procedure to determine whether to generate more of the transition rate matrix and apply the multi-step bounding algorithm or to apply the bound spread reduction algorithm to refine the bounds for previously generated states. Up to this point in the exposition, we make certain simplifying assumptions about the set of states the compose the submodel at each step in the iteration. Section 7 describes an extension that removes these assumptions and thereby further extends the results in [Lui and Muntz, 1991; Muntz et al., 1989]. An example is discussed in Section 8, and the conclusion is given in Section 9.

2. *Markov Model and Assumptions*

Consider a Markov model of a repairable computer system with state space $\mathscr{S}$. Let $R(i)$ be the reward rate associated with state i, $i \in \mathscr{S}$. Let $\mathscr{R}$ be the expected reward rate of the Markov model. We can express $\mathscr{R}$ as:

$$\mathscr{R} = \sum_{i \in \mathscr{S}} \pi(i) R(i), \tag{1}$$

where $\pi(i)$ is the steady state probability of state i. The system steady state availability, S_A, is a special case of this expected reward rate function where:

$$R(i) = \begin{cases} 1 & \text{if} \quad i \in \mathscr{O}, \\ 0 & \text{if} \quad i \in \mathscr{F}. \end{cases} \tag{2}$$

where $\mathscr{O}$ is the set of states in which the system is "operational" and $\mathscr{F}$ is the set of states in which the system has "failed".

Let N be the number of components in the system being modeled. We can partition the state space $\mathscr{S}$ as follows:

$$\mathscr{S} = \{\mathscr{F}_0 \cup \mathscr{F}_1 \cup \cdots \cup \mathscr{F}_N\}$$

where $\mathscr{F}_i$ contains exactly the states with i failed components. Figure 1 represents the transition rate matrix Q of the model in which $Q_{i,j}$ denotes the submatrix of transitions from $\mathscr{F}_i$ to $\mathscr{F}_j$.

In this paper, we make two assumptions concerning the availability model.

(1) the underlying Markov process is irreducible, and
(2) the underlying Markov process has a block Hessenberg structure, that is, $Q_{i,j} = 0$ for $j < i - 1$.

$$G = \left[\begin{array}{ccccccc}
Q_{00} & Q_{01} & Q_{02} & & \cdots & & Q_{0N} \\
Q_{10} & Q_{11} & Q_{12} & & \cdots & & Q_{1N} \\
0 & Q_{21} & Q_{22} & & \cdots & & \\
0 & 0 & Q_{32} & Q_{33} & & & \cdot \\
\cdot & & 0 & Q_{43} & Q_{44} & & \cdot \\
\cdot & \cdots & & \ddots & & \ddots & \cdot \\
\cdot & & & & & & \\
0 & \cdots & & & 0 & Q_{N,N-1} & Q_{NN}
\end{array}\right]$$

FIG. 1. Transition matrix G.

The second assumption implies that the probability of two or more components becoming operational in an interval of length Δt is $o(\Delta t)$. It is important to note that this assumption does not preclude multiple repair facilities or other common features of dependability models. Also note that a model in which a dormant component becomes active due to the repair of a second component does not violate the assumption. We simply do not consider such a dormant component as failed in the definition of the state partitions.

3. *One-Step Bounding Algorithm*

In this section, we describe a *one-step* bounding algorithm, as reported in [Muntz et al., 1989]. This one-step algorithm will be used as the initial step in the general procedure to be presented later. In [Muntz et al, 1989], several variations of the basic one-step bounding algorithm are described. Here, we will only summarize the one version of the algorithm that is used in developing the results of this paper.

Since computer systems are designed with high availability in mind, it is reasonable to expect that most of the components are operational most of the time. With this in mind, we partition the state space $\mathscr{S}$ as follow:

$$\mathscr{S} = \{\mathscr{D} \cup \mathscr{A}\},$$

where

$$\mathscr{D} = \{\mathscr{F}_0 \cup \mathscr{F}_1 \cdots \mathscr{F}_K\}$$
$$\mathscr{A} = \{\mathscr{F}_{K+1} \cup \mathscr{F}_{K+2} \cdots \mathscr{F}_N\}.$$

The idea we exploit in this section is to represent the transition rates between states in $\mathscr{D}$ exactly and represent the behavior of states in $\mathscr{A}$ approximately via aggregation [Courtois, 1977]. The intent is that $|\mathscr{D}| \ll |\mathscr{A}|$.

Let $\pi_{\mathscr{D}}$ denote the stationary state probabilities for states in $\mathscr{D}$. Suppose that we could determine a vector π' such that $\pi' \leq \pi_{\mathscr{D}}$.[3] Then, we can express bounds on the steady state system availability $\mathscr{S}_A$ in terms of π' as follows:

$$\sum_{i \in \mathscr{D}} \pi'(i)R(i) \leq S_A \leq \sum_{i \in \mathscr{D}} \pi'(i)R(i) + \left(1 - \sum_{i \in \mathscr{D}} \pi'(i)\right) \tag{3}$$

[3] For two vectors x and y, the notation $x\theta y$ means that $x_i\theta y_i$ for all corresponding elements.

where

$$R(i) = \begin{cases} 1 & \text{for operational states} \\ 0 & \text{for non-operational states.} \end{cases}$$

An interpretation of the above expression is that $1 - \Sigma_{i \in \mathscr{D}} \pi'(i)$ is the fraction of time that is not explicitly accounted for by the lower bound state probabilities for states in $\mathscr{D}$. The bounds are obtained by assuming that, in this fraction of time, the system is either always operational (upper bound) or always failed (lower bound).

In the remainder of this section we show how to efficiently obtain a tight lower bound vector, π'. There are two main ideas that are exploited to obtain this lower bound. The first involves the notion of *state cloning* that was introduced in [Muntz et al., 1989] and will be described next. In addition to state cloning, the concept of state aggregation [Courtois, 1977] is adapted to obtain an efficient solution. These ideas are now developed in more detail.

3.1. State Cloning. In [Muntz et al., 1989], the concept of state cloning is introduced which is a modification of the original model. The modified model, with the help of aggregation, can be exploited to obtain bounds. In the following, we summarize the use of state cloning.

Given a model with the transition rate matrix G, as illustrated in Figure 1, we form the following partition of the state space.

$$\begin{aligned} \mathscr{S}_0 &= \{\mathscr{F}_0\} \\ \mathscr{S}_1 &= \{\mathscr{F}_1 \cup \mathscr{F}_2 \cup \cdots \cup \mathscr{F}_K\} \\ \mathscr{S}_2 &= \{\mathscr{F}_{K+1} \cup \mathscr{F}_{K+2} \cup \cdots \cup \mathscr{F}_N\}. \end{aligned}$$

The transition rate matrix can then be conformally partitioned and represented as:

$$G = \begin{bmatrix} G_{00} & G_{01} & G_{02} \\ G_{10} & G_{11} & G_{12} \\ 0 & G_{21} & G_{22} \end{bmatrix}, \tag{4}$$

in which $G_{i,j}$ contains the rates for transitions from states in $\mathscr{S}_i$ to states in $\mathscr{S}_j$.

Figure 2 illustrates the concept of state cloning in which the states of $\mathscr{S}_1$ are replicated. Applying the technique of state cloning to the states in $\mathscr{S}_1$, the states of $\mathscr{S}_1$ are replicated and we refer to the two sets of states as $\mathscr{S}'_{1u}$ and $\mathscr{S}'_{1d}$. The transition rate matrix G' of the modified model is given by

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

where the ordering of the states in $\mathscr{S}_0, \mathscr{S}_{1u}, \mathscr{S}_{1u}, \mathscr{S}_2$.

Each state in $\mathscr{S}_1$ maps to a corresponding state in $\mathscr{S}'_{1u}$ and also to a state in $\mathscr{S}'_{1d}$. An interpretation of cloning is as follows: Assume the system starts in $\mathscr{S}_0$. As components fail and are repaired, the process remains in $\mathscr{S}_0$ and $\mathscr{S}'_{1u}$ until for the first time there are $K + 1$ or more failed components. At this point, the process enters a state in $\mathscr{G}_2$. When the number of failed components again falls to K, the process enters $\mathscr{G}'_{1d}$. The system then evolves in $\mathscr{G}_2$ and $\mathscr{G}'_{1d}$ until the next time it enters $\mathscr{G}_0$.

It is convenient to introduce a bit of notation at this point that will make it easier to express the relationship between the stationary state probability vectors for two Markov chains. Let $\mathscr{E}$ be any subset of states for a Markov chain with transition probability matrix G. The stationary state probability vector for states in $\mathscr{E}$ is denoted by $\pi_{\mathscr{E}/G}$. In the special case that $\mathscr{E}$ is the entire state space we denote the stationary state probability vector by π_G.

We return now to the main line of development. From the construction of G' from G, it is easy to show that the steady state probability vectors for the two Markov chains are related as stated in the following Lemma.

LEMMA 3.1. *Let G and G' be as defined above. Let $\mathscr{D} = \mathscr{G}_0 \cup \mathscr{G}_1$ and $\mathscr{D}' = \mathscr{G}_0 \cup \mathscr{G}'_{1u}$. Then $\pi_{\mathscr{D}'/G'} \leq \pi_{\mathscr{D}/G}$.*

PROOF. If $[\pi_{\mathscr{G}_0/G'}, \pi_{\mathscr{G}'_{1u}/G'}, \pi_{\mathscr{G}'_{1d}/G'}, \pi_{\mathscr{G}'_2/G'}]$ is the solution of $\pi_{G'}G' = 0$ and $\Sigma\pi_{G'}(i) = 1$, then $\pi_G = [\pi_{\mathscr{G}_0/G'}, \pi_{\mathscr{G}'_{1u}/G'} + \pi_{\mathscr{G}'_{1d}/G'}, \pi_{\mathscr{G}_2/G'}]$ is the solution of $\pi_G G = 0$ and $\Sigma\pi_G(i) = 1$.

It follows that $\pi_{\mathscr{G}_0/G} = \pi_{\mathscr{G}_0/G'}$ and, since $\pi_{G'_{1d/G'}} \geq 0$, that $\pi_{\mathscr{G}'_{1u}/G'} \leq \pi_{\mathscr{G}_1/G}$. The lemma follows directly from these observations. □

From the above lemma, it follows that if we could efficiently obtain a solution for $\pi_{\mathscr{G}'_{1u}/G'}$ and $\pi_{\mathscr{G}_0/G'}$, then we would have the lower bound we need. Of course, the matrix G' is larger than the original matrix G, so solving G' directly is not an improvement. We now show that due to (a) the original model being block Hessenberg and (b) the manner in which state cloning altered the model, we will be able to use aggregation to reduce the size of the model that has to be solved and still maintain lower bounds on the steady state probabilities for states in $\mathscr{D}$.

3.2. STATE AGGREGATION AND TRANSITION RATE BOUNDS. Let $\mathscr{G}'_{1d} = \bigcup_{i=1}^{K}\mathscr{F}'_i$ where $\mathscr{F}'_i$, $1 \leq i \leq K$, is the subset of states of $\mathscr{G}'_{1d}$ in which there are exactly i failed components. For each $\mathscr{F}_i, i \geq K + 1$ form one aggregate state and for each $\mathscr{F}'_i, 1 \leq i \leq K$ form one aggregate state. Then the transition rate matrix G'', is depicted in Figure 3 where the $r_{i,j}$ elements represent transition rates between aggregate states. Note that the submatrices $G_{1,j}$, $K < j \leq N$ have single columns and the submatrices $G_{0,0}$ and $G_{0,j} K < j \leq N$ are 1×1 matrices since $\mathscr{G}_0$ contains only a singleton state. Assuming that exact aggregation [Courtois, 1977; 1982] is used to compute the transition rates out of the aggregate states then:

$$\pi_{\mathscr{D}'/G'} = \pi_{\mathscr{D}'/G''}, \tag{5}$$

where $\mathscr{D}' = \mathscr{G}_0 \cup \mathscr{G}'_{1u}$.

Of course, the described construction of G'' assumed exact aggregation that, in general, requires having a knowledge of the conditional state probabilities for each subset of states being aggregated. We now show that bounds, rather than exact values, for the aggregate transition rates will suffice to obtain bounds on $\pi_{\mathscr{D}'/G''}$.

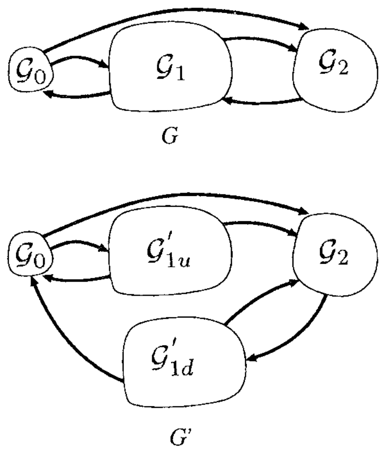

FIG. 2. Relationship of G and G'.

$$G'' = \left[\begin{array}{cc|cccccc}
G_{00} & G_{01} & 0 & \cdots & 0 & G_{0\ K+1} & \cdots & G_{0N} \\
G_{10} & G_{11} & 0 & \cdots & 0 & G_{1\ K+1} & \cdots & G_{1N} \\
\hline
r_{1'0} & 0 & \bullet & \cdots & r_{1'K'} & r_{1'K+1} & \cdots & r_{1'N} \\
0 & 0 & r_{2'1'} & \bullet & \cdots & r_{2'K+1} & \cdots & r_{2'N} \\
0 & 0 & 0 & r_{3'2'} & \cdots & r_{3'K+1} & \cdots & r_{3'N} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & r_{N\ N-1} & \bullet
\end{array}\right]$$

FIG. 3. Form of transition matrix after aggregation.

Consider a Markov process with generator G'' as depicted in Figure 3. Construct a generator matrix G''' from G'' by replacing the non-zero aggregate failure rates (i.e., r_{ij} with $j > i$) by upper bounds and the aggregate repair rates (i.e., r_{ij} with $j = i - 1$) with non-zero lower bounds (see Figure 4). The Markov chain remains irreducible and the following theorem relates $\pi_{\mathcal{D}'/G'''}$ and $\pi_{\mathcal{D}'/G''}$.

THEOREM 3.1. $\pi_{\mathcal{D}'/G'''} \leq \pi_{\mathcal{D}'/G''}$.

PROOF. This is a simple application of Lemma 2 that is given in the Appendix. □

3.3. SUMMARY. In summary, the one-step bounding algorithm can be described as follows:

```
procedure one-step bounding algorithm
begin
  Pick $K$ such that the transition rates in $\mathcal{D}$ can be represented exactly;
  Generate the matrix $G''$ as in Figure 3 (except for the aggregate transition rates).
  Generate upper and lower bound aggregate transition rates to form the matrix $G'''$ as in
     Figure 4.
  Solve the model;
  Based on the computed (lower bound) probabilities for states in $\mathcal{D}$, compute the upper
     and lower bounds on $S_A$ from Eq. 3;
end
```

$$\left[\begin{array}{cc|cccccc}
G_{00} & G_{01} & 0 & \cdots & 0 & G_{0\;K+1} & \cdots & G_{0N} \\
G_{10} & G_{11} & 0 & \cdots & 0 & G_{1\;K+1} & \cdots & G_{1N} \\
\hline
- & 0 & \bullet & \cdots & + & + & \cdots & + \\
0 & 0 & - & \bullet & \cdots & + & \cdots & + \\
0 & 0 & 0 & - & \cdots & + & \cdots & + \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & - & \bullet
\end{array}\right]$$

FIG. 4. Form of rate matrix after bounding transition rates.

4. *Multi-Step Bounding Algorithm*

In the "one-step" bounding algorithm described in the previous section, the dimension of the submatrix $G_{\mathscr{D},\mathscr{D}}$ is specified a priori and once the steady state availability bounds have been calculated, there is no method provided for further tightening of the bounds. In this section, we describe a *multi-step* bounding algorithm [Lui and Muntz, 1991], which allows the bounds to be tightened by investing more time in computation. The multi-step algorithm allows incremental generation of more of the transition rate matrix, that is, 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 provides lower bounds on the stationary state probabilities for an additional set of states. This allows incremental improvement of the bounds on system availability.

We introduce the following notation for the multi-step bounding algorithm:

$\mathscr{D}_i$ = the set of states which are generated in the ith step of the bounding procedure and for which lower bounds on the stationary state probabilities are to be calculated.
$\mathscr{C}_i$ = clone states for the states in $\mathscr{D}_i$.
$\mathscr{C}_i^j$ = set of states in $\mathscr{C}_i$ with exactly j failed components.
c_i^j = aggregate state corresponding to $\mathscr{C}_i^j$.
$\mathscr{D}_i'$ = $\{\mathscr{D}_1 \cup \cdots \cup \mathscr{D}_{i-1} \cup \mathscr{C}_1 \cup \cdots \cup \mathscr{C}_{i-1}\}$.
d' = aggregate state corresponding to $\mathscr{D}'$
$\mathscr{A}_i$ = the complement of $\{\mathscr{D}_i' \cup \mathscr{D}_i\}$ (i.e., the portion of the state space that is unexplored.)
a_j = aggregate state for all the states in $\mathscr{A}_i$ that have exactly j failed components.
L_i = integer associated with the ith step that denotes the minimum number of failed components for states in $\mathscr{D}_i$.
H_i = integer associated with the ith step that denotes the maximum number of failed components for states in $\mathscr{D}_i$.
$Q_{\mathscr{D}_i,\mathscr{D}_i}$ = transition rate matrix containing transition rates between states in $\mathscr{D}_i$.
$Q_{\mathscr{D}_{i,j}}$ = transition rate vector from states in $\mathscr{D}_i$ to state j.
$R_{d',\mathscr{D}_i}$ = row vector of transition rates from aggregate state d' to states in $\mathscr{D}_i$.
$R_{d',\mathscr{C}_i}$ = row vector of transition rates from aggregate state d' to states in $\mathscr{C}_i$.
$r_{i,j}$ = transition rate from aggregate state i to aggregate state j.

At the ith step of the "multi-step" bounding algorithm, there are three disjoint sets of states. They are: $\mathscr{D}_i'$, $\mathscr{D}_i$, and $\mathscr{A}_i$. During the ith step of the algorithm, $\mathscr{D}_i$ is composed of all the states that have between L_i and H_i failed components. Figure 5 illustrates this partitioning of the state space in terms of the transition matrix G.

FIG. 5. Rate matrix G. Initial matrix.

$$\begin{bmatrix} Q_{\mathcal{D}'\mathcal{D}'} & Q_{\mathcal{D}'\mathcal{D}_t} & Q_{\mathcal{D}'\mathcal{A}} \\ Q_{\mathcal{D}_t\mathcal{D}'} & Q_{\mathcal{D}_t\mathcal{D}_t} & Q_{\mathcal{D}_t\mathcal{A}} \\ 0 & Q_{\mathcal{A}\mathcal{D}_t} & Q_{\mathcal{A}\mathcal{A}} \end{bmatrix}$$

To illustrate how the multi-step bounding algorithm provides lower-bound state probabilities for states in $\mathscr{D}_t$, we describe a sequence of transformations of the rate matrix G such that for each state space transformation, state probabilities for states in $\mathscr{D}_t$ in the original model are individually bounded from below by the state probabilities for $\mathscr{D}_t$ in the transformed model. During this sequence of state space transformations, we use the basic aggregation/disaggregation technique described in [Courtois, 1988]. Of course, just as in the one-step bounds procedure, exact aggregation is *not actually required* in the computation of the steady state system availability bounds. We merely use the *existence* of an exact aggregation in the intermediate steps of the development. In the end, we only need bounds on transition rates out of the aggregate states.

Figure 6 depicts the rate matrix G_1 which results from the first transformation of G. G_1 corresponds to the cloning of the states in $\mathscr{D}_t$ and the set of cloned states is denoted by $\mathscr{C}_t$. Note that in the rate matrix G_1, the submatrix $Q_{\mathscr{C}_t,\mathscr{C}_t}$ is equal to $Q_{\mathscr{D}_t,\mathscr{D}_t}$. A similar transformation was described in the previous section and based on that discussion, it is clear that:

if $[\pi_{\mathscr{D}'/G_1}, \pi_{\mathscr{D}_t/G_1}, \pi_{\mathscr{C}_t/G_1}, \pi_{\mathscr{A}/G_1}]$ is the solution of $\pi_{G_1}G_1 = 0$ and $\Sigma\pi_{G_1}(i) = 1$,
then $[\pi_{\mathscr{D}'/G_1}, \pi_{\mathscr{D}_t/G_1} + \pi_{\mathscr{C}_t/G_1}, \pi_{\mathscr{A}/G_1}]$ is the solution of $\pi_G \mathrm{G} = 0$ and $\Sigma\pi_G(i) = 1$.
Since $\pi_{\mathscr{C}_t/G_1} \geq 0$, the following relationship holds:

$$\pi_{\mathscr{D}_t/G_1} \leq \pi_{\mathscr{D}_t/G} \tag{6}$$

Figure 7 depicts the next transformation. In this transformation, G_2 is formed from G_1 by applying exact aggregation to the states in $\mathscr{D}'$. Let d' be the state that represents the aggregation of all states in $\mathscr{D}'$. Because exact aggregation is assumed, we have the following relationship:

$$\pi_{\mathscr{D}_t/G_2} = \pi_{\mathscr{D}_t/G_1}. \tag{7}$$

G_3 in Figure 8 is the result of the next transformation on G_2. G_3 has a structure similar to that of G_2 except that the transitions from d' to states in $\mathscr{D}_t$ and $\mathscr{C}_t$ are modified. The submatrices $R'_{d'\mathscr{D}_t}$ and $R'_{d'\mathscr{C}_t}$ in G_3 and $R_{d'\mathscr{D}_t}$ in G_2 are required to satisfy the following constraints:

$$R'_{d'\mathscr{D}_t} \geq \mathbf{0}$$

$$R'_{d'\mathscr{C}_t} \geq \mathbf{0}$$

$$R'_{d'\mathscr{D}_t} + R'_{d'\mathscr{C}_t} = R_{d'\mathscr{D}_t}. \tag{8}$$

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

Based on the definition of G_3, we have the following theorem:

THEOREM 4.1. $\pi_{\mathscr{D}_t/G_3} \leq \pi_{\mathscr{D}_t/G_2}$

$$\begin{bmatrix} Q_{\mathcal{D}'\mathcal{D}'} & Q_{\mathcal{D}'\mathcal{D}_i} & 0 & Q_{\mathcal{D}'\mathcal{A}} \\ Q_{\mathcal{D}_i\mathcal{D}'} & Q_{\mathcal{D}_i\mathcal{D}_i} & 0 & Q_{\mathcal{D}_i\mathcal{A}} \\ Q_{\mathcal{D}_i\mathcal{D}'} & 0 & Q_{\mathcal{C}_i\mathcal{C}_i} & Q_{\mathcal{D}_i\mathcal{A}} \\ 0 & 0 & Q_{\mathcal{A}\mathcal{D}_i} & Q_{\mathcal{A}\mathcal{A}} \end{bmatrix}$$

FIG. 6. Rate matrix G_1. Introduction of clone states.

$$\begin{bmatrix} \bullet & R_{d'\mathcal{D}_i} & 0 & R_{d'\mathcal{A}} \\ Q_{\mathcal{D}_i d'} & Q_{\mathcal{D}_i\mathcal{D}_i} & 0 & Q_{\mathcal{D}_i\mathcal{A}} \\ Q_{\mathcal{D}_i d'} & 0 & Q_{\mathcal{C}_i\mathcal{C}_i} & Q_{\mathcal{D}_i\mathcal{A}} \\ 0 & 0 & Q_{\mathcal{A}\mathcal{D}_i} & Q_{\mathcal{A}\mathcal{A}} \end{bmatrix}$$

FIG. 7. Rate matrix G_2. After exact aggregation of the states in $\mathscr{D}'$.

$$\begin{bmatrix} \bullet & R'_{d'\mathcal{D}_i} & R'_{d'\mathcal{C}_i} & R_{d'\mathcal{A}} \\ Q_{\mathcal{D}_i d'} & Q_{\mathcal{D}_i\mathcal{D}_i} & 0 & Q_{\mathcal{D}_i\mathcal{A}} \\ Q_{\mathcal{D}_i d'} & 0 & Q_{\mathcal{C}_i\mathcal{C}_i} & Q_{\mathcal{D}_i\mathcal{A}} \\ 0 & 0 & Q_{\mathcal{A}\mathcal{D}_i} & Q_{\mathcal{A}\mathcal{A}} \end{bmatrix}$$

FIG. 8. Rate matrix G_3. Modified rates from state d'.

PROOF. First, the following equations are easily seen to hold:

$$\pi_{d'/G_2} = \pi_{d'/G_3},$$
$$\pi_{\mathscr{D}_i/G_2} + \pi_{\mathscr{C}_i/G_2} = \pi_{\mathscr{D}_i/G_3} + \pi_{\mathscr{C}_i/G_3},$$
$$\pi_{\mathscr{A}/G_2} = \pi_{\mathscr{A}/G_3}.$$

The flow conservation equation for states in $\mathscr{D}_i$ corresponding to G_2 and G_3 are:

$$\pi_{\mathscr{D}_i/G_2} = \pi_{d'/G_2} R_{d'\mathscr{D}_i}\left(-Q_{\mathscr{D}_i\mathscr{D}_i}\right)^{-1},$$
$$\pi_{\mathscr{D}_i/G_3} = \pi_{d'/G_3} R'_{d'\mathscr{D}_i}\left(-Q_{\mathscr{D}_i\mathscr{D}_i}\right)^{-1}.$$

Observe that since $(-Q_{\mathscr{D}_i\mathscr{D}_i})$ is a non-singular M-matrix [Varga, 1962], it's inverse is a non-negative matrix and since $R_{d'\mathscr{D}_i} \geq R_{d'\mathscr{D}_i}$, it follows immediately that $\pi_{\mathscr{D}_i/G_2} \geq \pi_{\mathscr{D}_i/G_3}$. □

Theorem 4.1 holds for any vectors $R'_{d'\mathscr{D}_i}$ and $R'_{d'\mathscr{C}_i}$ which satisfy Eq. (8). In particular, we now define G'_3 by selecting specific values for these vectors. The following provides a definition of a particular value for $R'_{d',\mathscr{D}_i}$, which is computable at the ith step.

$$\begin{aligned} R_{d',\mathscr{D}_i} &= [\pi_{\mathscr{D}'}\mathbf{1}]^{-1}\pi_{\mathscr{D}'}Q_{\mathscr{D}',\mathscr{D}_i} \\ &\geq [\pi'_{\mathscr{D}'}\mathbf{1} + (1-a)]^{-1}\pi'_{\mathscr{D}'}Q_{\mathscr{D}',\mathscr{D}_i} = R'_{d',\mathscr{D}_i}, \end{aligned} \tag{9}$$

where:

$\pi_{\mathscr{D}'}$ = exact steady state probability vector for states in $\mathscr{D}'$.
$\pi'_{\mathscr{D}'}$ = lower bound steady state probability vector for states in $\mathscr{D}'$ (which we have computed in previous steps).
a = sum of the lower bound state probabilities for all those states we have generated thus far.
$\mathbf{1}$ = column vector of 1's.

We also set $R'_{d',\mathscr{C}_l} = R_{d',\mathscr{D}_l} - R'_{d',\mathscr{D}_l}$. Of course, $R_{d',\mathscr{D}_l}$ is not known so that this can only be done in the abstract. (After the next transformation we will show that the exact value for $R_{d',\mathscr{D}_l}$ does not have to be known to obtain bounds.) It follows from the definition of G'_3 that $\pi_{\mathscr{D}_l/G'_3} \leq \pi_{\mathscr{D}_l/G_2}$.

In the next transformation, exact aggregation is applied to the states in $\mathscr{C}_l$ and $\mathscr{A}$. One aggregate state is formed for each subset $\mathscr{F}'_j$ in $\mathscr{C}_l$ and $\mathscr{F}_j$ in $\mathscr{A}$. The result of this transformation is the rate matrix G_4 as depicted in Figure 9. Note that the relative ordering of the state d' and the set of states $\mathscr{D}_l$ has been permuted. This will make it a bit simpler to describe the next transformation. Since exact aggregation is assumed in forming G_4, the following relationship holds:

$$\pi_{\mathscr{D}_l/G_4} = \pi_{\mathscr{D}_l/G'_3}. \tag{10}$$

In the next transformation G_5 is constructed from G_4 by replacing some of the aggregate transition rates by upper bounds and some by lower bounds. Specifically, (see Figure 10) the transition rates from state d' to the aggregate states $c_l^{L_l}, \ldots, c_l^{H_l}$ and to the aggregate states $a_l^{L_l}, \ldots, a_n$ are replaced by upper bounds. Similarly the aggregate "repair rates", $r_{c_l}^{L_l+1}, r_{c_l}^{L_l}, \ldots, r_{a_{H_l+1}, c_l^{H_l}}, \ldots, r_{a_{H_l+1}, c_l^{H_l}}, \ldots, r_{a_n, a_{n-1}}$ are replaced by lower bounds. (How to compute bounds is discussed below.) The relationship between G_4 and G_5 satisfies the requirements of Lemma 2 in the Appendix and therefore we have:

$$\pi_{\mathscr{D}_l/G_5} \leq \pi_{\mathscr{D}_l/G_4}. \tag{11}$$

In the sequence of model transformations that has been described, each yields a lower bound on the stationary state probabilities for states in $\mathscr{D}_l$ compared with the previous model and therefore:

$$\pi_{\mathscr{D}_l/G_5} \leq \pi_{\mathscr{D}_l/G}. \tag{12}$$

In terms of state space cardinality, clearly G_5 has a much reduced state space compared with that of G. The remaining question is how to provide bounds for the transition rates indicated in Figure 10.

Preliminary to computing bounds for the transition rates r_{d',d_l^j}, $L_l \leq j \leq H_l$, let us define the following:

$\mathscr{D}_l^j$ = states in $\mathscr{D}_l$ with exactly j failed components.
$R'_{d',\mathscr{D}_l^j}$ = transition rate vector from state d' to states in $\mathscr{D}_l^j$.
$Q_{\mathscr{D}',\mathscr{D}_l^j}$ = transition rate matrix from states $\mathscr{D}'$ to states in $\mathscr{D}_l^j$.
$r_{\max}(i,j)$ = maximum entry in vector $Q_{\mathscr{D}'_l,\mathscr{D}_l^j}\mathbf{1}$.
a = sum of the lower bound state probabilities for all those states we have generated thus far.

The rate from aggregate state d' to aggregate state c_l^j, r_{d',c_l^j}, is easily seen to satisfy the relationship:

$$r_{d',c_l^j} \leq \min\Big\{\text{sum of all failure rates},$$
$$\left[\pi'_{\mathscr{D}'_l}\mathbf{1}\right]^{-1}\left[\pi'_{\mathscr{D}'_l}Q_{\mathscr{D}'_l,\mathscr{D}_l^j}\mathbf{1} + (1-a)r_{\max}\right] - R'_{d',\mathscr{D}_l^j}\mathbf{1}\Big\}, \tag{13}$$

which provides an upper bound for transition rates from state d' to each of the aggregate states c_l^j, $L_l \leq j \leq H_l$. Bounds on the transition rates from d' to the

$$
\left[\begin{array}{cc|cccc|cccc}
Q_{\mathcal{D}_\iota,\mathcal{D}_\iota} & Q_{\mathcal{D}_\iota,d'} & 0 & 0 & \cdots & 0 & Q_{\mathcal{D}_\iota,a_{H_\iota+1}} & Q_{\mathcal{D}_\iota,a_{H_\iota+2}} & \cdots & Q_{\mathcal{D}_\iota,a_n} \\
R'_{d',\mathcal{D}_\iota} & \bullet & r_{d',c_\iota^{L_\iota}} & r_{d',c_\iota^{L_\iota+1}} & \cdots & r_{d',c_\iota^{H_\iota}} & r_{d',a_{H_\iota+1}} & r_{d',a_{H_\iota+2}} & \cdots & r_{d',a_n} \\
\hline
0\ldots0 & r_{c_\iota^{L_\iota},d'} & \bullet & r_{c_\iota^{L_\iota},c_\iota^{L_\iota+1}} & \cdots & r_{c_\iota^{L_\iota},c_\iota^{H_\iota}} & r_{c_\iota^{L_\iota},a_{H_\iota+1}} & r_{c_\iota^{L_\iota},a_{H_\iota+2}} & \cdots & r_{c_\iota^{L_\iota},a_n} \\
0\ldots0 & 0 & r_{c_\iota^{L_\iota+1},c_\iota^{L_\iota}} & \bullet & \cdots & r_{c_\iota^{L_\iota+1},c_\iota^{H_\iota}} & r_{c_\iota^{L_\iota+1},a_{H_\iota+1}} & r_{c_\iota^{L_\iota+1},a_{H_\iota+2}} & \cdots & r_{c_\iota^{L_\iota+1},a_n} \\
0\ldots0 & 0 & 0 & r_{c_\iota^{L_\iota+2},c_\iota^{L_\iota+1}} & \bullet & r_{c_\iota^{L_\iota+2},c_\iota^{H_\iota}} & r_{c_\iota^{L_\iota+2},a_{H_\iota+1}} & r_{c_\iota^{L_\iota+2},a_{H_\iota+2}} & \cdots & r_{c_\iota^{L_\iota+2},a_n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0\ldots0 & \vdots & 0 & \cdots & \cdots & \bullet & r_{c_\iota^{H_\iota},a_{H_\iota+1}} & \cdots & \cdots & r_{c_\iota^{H_\iota},a_n} \\
\hline
0\ldots0 & 0 & 0 & 0 & 0 & r_{a_{H_\iota+1},c_\iota^{H_\iota}} & \bullet & r_{a_{H_\iota+1},a_{H_\iota+2}} & \cdots & r_{a_{H_\iota+1},a_n} \\
\vdots & 0 & 0 & 0 & 0 & 0 & r_{a_{H_\iota+2},a_{H_\iota+1}} & \bullet & \cdots & r_{a_{H_\iota+2},a_n} \\
\vdots & 0 & 0 & 0 & 0 & 0 & 0 & r_{a_{H_\iota+3},a_{H_\iota+2}} & \bullet & r_{a_{H_\iota+3},a_n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
0\ldots0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \bullet
\end{array}\right]
$$

FIG. 9. Rate matrix G_4. Exact aggregation of states in $\mathscr{C}_\iota$ and $\mathscr{A}_\iota$.

$$
\left[\begin{array}{cc|ccccc|ccccc}
Q_{\mathcal{D}_\iota,\mathcal{D}_\iota} & Q_{\mathcal{D}_\iota,d'} & 0 & 0 & \cdots & \cdots & 0 & Q_{\mathcal{D}_\iota,A_{H_\iota+1}} & Q_{\mathcal{D}_\iota,A_{H_\iota+2}} & \cdots & \cdots & Q_{\mathcal{D}_\iota,A_N} \\
R'_{d',\mathcal{D}_\iota} & \bullet & + & + & \cdots & \cdots & + & + & + & \cdots & \cdots & + \\
\hline
0\ldots0 & - & \bullet & + & \cdots & \cdots & + & + & + & \cdots & \cdots & + \\
0\ldots0 & 0 & - & \bullet & \cdots & \cdots & + & + & + & \cdots & \cdots & + \\
\vdots & \vdots & 0 & - & \bullet & \cdots & + & + & \cdots & \cdots & \cdots & + \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0\ldots0 & \vdots & 0 & \cdots & \cdots 0 & - & \bullet & + & \cdots & \cdots & \cdots & + \\
\hline
0\ldots0 & 0 & 0 & 0 & 0 & \cdots & - & \bullet & + & \cdots & \cdots & + \\
\vdots & 0 & 0 & 0 & 0 & \cdots & 0 & - & \bullet & \cdots & \cdots & + \\
\vdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & - & \bullet & \cdots & + \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \ddots & \vdots \\
0\ldots0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & - & \bullet
\end{array}\right]
$$

FIG. 10. Rate matrix G_5. Replacement of transition rates with bounds.

aggregate states $a_k, H_\iota < k \leq N$, have an analogous expression in which the appropriate submatrix is substituted for $Q_{\mathscr{D}',\mathscr{D}_k^\iota}$, that is, the submatrix containing the transition rates from $\mathscr{D}_\iota$ to $\mathscr{A}_\iota^k$ (the set of states in $\mathscr{A}$ with k failed components). The required bounds on the rates out of the remaining aggregate states can be found as described in the section on the one-step bounds.

The multi-step bounding algorithm can be summarized as follows:

```
procedure   multi-step bounding algorithm
begin
  Aggregate all states in $\mathscr{D}_\iota'$ to a single state;
  Decide on the dimension of $\mathscr{D}_\iota$ and generate the transition rates for states in $\mathscr{D}_\iota$;
  Generate upper and lower bound aggregate transition rates using Eqs. (9) and (13) for the
    rates out of $d'$ and simple bounds for the rates out of the other aggregate states, that
    is, minimum repair rate for lower bounds and sum of failure rates for upper
    bounds.
    Solve the model;
    Based on the lower bound state probability vectors for $\mathscr{D}_i$ and $\mathscr{D}_i'$, compute the improved
        bounds on system availability $\mathscr{S}'_{\mathscr{A}}$ using Eq. 3;
end
```

5. *Bound Spread Reduction*

In the previous two sections, we have described a *one-step* and a *multi-step procedure* for bounding the steady state availability of a repairable computer system. The multi-step bounding algorithm provides only a partial answer to further tightening of the system availability bounds. It is not a complete solution because errors are introduced at each step that are never recovered in later steps. The implication is that the bound spread will not approach zero using the procedures that we have developed thus far. At each step of the multi-step procedure, there are two components to the bound spread that are introduced and which are not later reduced by further steps. The two components are:

(1) in considering the clone states, $\mathscr{C}_i$ to have reward of 0 or 1 in the evaluation of the availability bounds
(2) in the difference between the lower bounds and the actual values of the stationary state probabilities of the "detailed states" in $\mathscr{D}_i$

The contribution to the bound spread by components (1) and (2) are not reduced by successive applications of the multi-step algorithm. In this section, we show how we can *reduce* these errors and obtain tighter bounds by re-evaluating previously calculated bounds for state probabilities. The bound spread reduction algorithm we propose is iterative in nature. We first reduce the error associated with clone states by obtaining lower bounds on their stationary 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" $\mathscr{D}_i$. An important point is that the bound spread reduction algorithm does not require generation of more of the transition matrix (it does require reusing previously generated portions of the transition rate matrix). In Section 5.1, we present the approach to reducing the error associated with clone states and in Section 5.2, we present the approach to obtaining improved lower bound state probabilities for the "detailed states", $\mathscr{D}_i$.

5.1. Bound Spread Reduction for the Clone States. Until now, we have assigned a reward of 0 or 1 to all aggregate clone states in the computation of the system availability bounds. In this way, we never had to be concerned with the state probabilities for individual clone states in $\mathscr{C}_i$. To reduce this source of error, we would like to obtain a lower bound for the state probability of each individual clone state and incorporate this information into the bounds. To do this, we make use of the fact that the clone states $\mathscr{C}_i$, have exactly the same transition structure as the detailed states, $\mathscr{D}_i$. Since this portion of the transition rate matrix is already generated, we can use it to compute a lower bound on the state probabilities of the individual clone states.

Assume that at the $i + 1$st step of the multi-step procedure, we have already obtained lower bounds on the state probabilities for all states in $\mathscr{D}_{i+1}$ and we now want to find lower bounds on the steady state probabilities for the clone states in $\mathscr{C}_l$. Let us define the following notation:

$\pi_{\mathscr{C}_l}$ = steady state probability vector for the clone states in $\mathscr{C}_l$

$\pi_{\mathscr{D}_{l+1}^{L_{l+1}}}$ = steady state probability vector for the detailed states in $\mathscr{D}_{l+1}$ with exactly L_{l+1} failed components.

$Q_{\mathscr{D}_{l+1}^{L_{l+1}},\mathscr{C}_l}$ = portion of the transition rate matrix corresponding to transition from states in $\mathscr{D}_{l+1}^{L_{l+1}}$ to states in $\mathscr{C}_l$.

From flow conservation we have:

$$\pi_{\mathscr{C}_l} Q_{\mathscr{D}_l,\mathscr{D}_l} + \pi_{\mathscr{D}_{l+1}^{L_{l+1}}} Q_{\mathscr{D}_{l+1}^{L_{l+1}},\mathscr{C}_l} = \mathbf{0} \tag{14}$$

The first term holds because $Q_{\mathscr{D}_l,\mathscr{D}_l} = Q_{\mathscr{C}_l,\mathscr{C}_l}$. In the previous step (of the multi-step procedure), we have obtained lower bounds on the state probabilities $\pi_{\mathscr{D}_{l+1}^{L_{l+1}}}$. We now show that by applying an iterative solution method (e.g., Jacobi or Gauss–Seidel Iterative method [Varga, 1962]), we obtain a lower bound on the state probabilities of the clone states. In the remainder of this section, we formulate the iterative procedure using the Gauss–Seidel iterative method. Note that $\pi_{\mathscr{D}_{l+1}^{L_{l+1}}}$ is constant in this algorithm (the lower bound state probabilities obtained from the previous bounding step). We show that the iterative procedure:

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

These characteristics are especially interesting because they indicate that the iterative process can be *terminated at any step* and the current values are guaranteed to be lower bounds on the state probabilities of the clone states. Once the lower-bound state probabilities of the clone states are known, we can obtain tighter system availability bounds.

Let us rewrite Eq. (14) as:

$$-Q_{\mathscr{D}_l,\mathscr{D}_l}^T \pi_{\mathscr{C}_l}^T = Q_{\mathscr{D}_{l+1}^{L_{l+1}},\mathscr{C}_l}^T \pi_{\mathscr{D}_{l+1}^{L_{l+1}}}^T, \tag{15}$$

which has the form of a linear system $\mathbf{Ax} = \mathbf{b}$ (with $\mathbf{x} = \pi_{\mathscr{C}_l}^T$). Note that each diagonal element of $\mathbf{A}$ is the absolute value of the transition rate out of the associated clone state and the off diagonal elements are the "negated" transition rates from one clone state to another clone state. 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 lower and upper triangular matrices, respectively. The Gauss–Seidel iteration can be written as[4]:

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

with

$$\mathbf{x}^{(0)} = \mathbf{0} \tag{16}$$

A necessary and sufficient condition for the above iterative process to converge to a unique solution is for the spectral radius $\rho[(\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{U}_\mathrm{A}]$ to be less than 1 [Burden and Faires, 1989]. Since $\mathbf{A}$ and $(\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})$ are nonsingular M-matrices, their inverses are nonnegative matrices, and $(\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})$ and $\mathbf{U}_\mathrm{A}$

[4] In order to guarantee the claimed characteristics, an initial vector of zeros is chosen. There exist other initial vectors that can speed up the convergence rate but the zero vector is the only known lower bound vector the first time we apply this procedure.

form a regular splitting of the matrix $\mathbf{A}$ [Varga, 1962]. Hence, the spectral radius, $\rho[(\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{U}_\mathrm{A}]$, is less than 1. Therefore the iterative process does converge to a unique solution. In the following, we show that the iterative procedure also has the other claimed characteristics. To show the convergence characteristics of our algorithm, we need the following theorem from [Berman and Plemmons, 1979]:

THEOREM 5.1. *Consider a linear system $Ax = b$ and the iterative formula*

$$x^{(k+1)} = Hx^{(k)} + C$$

Let $A = M - N$ with A and M being nonsingular and $H \geq 0$ where $H = M^{-1}N$. If there exists $x^{(0)}$ such that $x^{(0)} \leq x^{(1)}$ is computed from the iterative formula, then:

$$x^{(0)} \leq x^{(1)} \leq \cdots \leq x^{(i)} \leq \ldots A^{-1}b.$$

COROLLARY 5.1. *The proposed iterative procedure converges monotonically.*

PROOF. The result follows directly by making the following associations with the terms in Theorem 5.1.

$$\begin{aligned} M &= (\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A}) \\ N &= \mathbf{U}_\mathrm{A} \\ C &= (\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{b} \\ \mathbf{x}^{(0)} &= 0. \end{aligned}$$

□

LEMMA 5.1. *The fixed point x^* of the proposed iterative procedure is a lower bound on the exact state probabilities of the clone states.*

PROOF. Let $\mathbf{x}'$ be the exact state probability vector for the clone states. We have to show that $\mathbf{x}' - \mathbf{x}^* \geq \mathbf{0}$. Let $\mathbf{b}'$ be a row vector containing the *exact* rates into the clone states from states in $\mathscr{D}_{t+1}^{L_{t+1}}$. Then:

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

The above inequality holds because we computed $\mathbf{b}$ from lower bound state probabilities for states in $\mathscr{D}_t^{L_{t+1}}$. It is easily seen that:

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

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

To summarize, the algorithm for tightening the bounds on the clone states in $\mathscr{C}_t$ is as follows[5]:

```
procedure   Bound Spread Reduction for $\mathscr{C}_t$
begin
  Let $\pi_{\mathscr{C}_t}^{(0)} = \mathbf{0}$;
  do
    apply the iterative procedure embodied in Eq. (16);
  while (specified tolerance is not satisfied)
end
```

[5] We are actually computing tighter lower bounds on the states in $\mathscr{C}_t$ but we refer to this as "bound spread reduction" since tighter bounds on the state probabilities for $\mathscr{C}_t$ lead directly to tighter system availability bounds.

5.2. Bound Spread Reduction for the Detailed States. Recall that in computing lower bounds for the state probabilities of the detailed states $\mathscr{D}_i$, 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 5.1, we obtain improved lower bounds for the stationary state probabilities of the clone states. In this section, we show that these lower bounds on the clone state probabilities can be used to generate tighter bounds on the aggregate transition rates out of the clone states and thereby obtain improved lower bounds for the state probabilities of the detailed states, $\mathscr{D}_i$.

Let us define the following notation:

$r(c_i^j, c_i^k)$ = transition rate from the clone aggregate c_i^j to clone aggregate c_i^k.
a = sum of the lower bound state probabilities for all states generated thus far.
$\pi_{\mathscr{C}_i^k}$ = lower bound state probability vector for clone states in $\mathscr{C}_i$ with exactly k failed components.
$r_{\max}^{ikm}$ = maximum entry in vector $Q_{\mathscr{D}_i^k, \mathscr{D}_i^m}\mathbf{1}$.

Since we already have a lower bound for the state probability of each clone state, *improved* lower bounds on the "repair rates" between the aggregates can be obtained as follows:

$$r\left(c_i^k, c_i^{k-1}\right)^{-} = \max\left\{\text{minimum repair rate}, \left[\pi_{\mathscr{C}_i^k}\mathbf{1} + (1-a)\right]^{-1}\pi_{\mathscr{C}_i^k}Q_{\mathscr{D}_i^k, \mathscr{D}_i^{k-1}}\mathbf{1}\right\}. \tag{17}$$

Similarly, an *improved* upper bound on the failure rate from aggregate c_i^k to aggregate c_i^m is as follows:

$$r\left(c_i^k, c_i^m\right)^{+} = \min\Big\{\text{sum of all failure rates},$$
$$\left(\pi_{\mathscr{C}_i^k}\mathbf{1}\right)^{-1}\left[\pi_{\mathscr{C}_i^k}Q_{\mathscr{D}_i^k, \mathscr{D}_i^m}\mathbf{1} + (1-a)r_{\max}^{ikm}\right]\Big\}. \tag{18}$$

Equations (17) and (18) follow easily from considering the conditional transition rates between aggregate based on upper and lower bounds on the conditional state probabilities. For example, in Eq. (17):

$$\left[\pi_{\mathscr{C}_i^k}\mathbf{1} + (1-a)\right]^{-1}\pi_{\mathscr{C}_i^k}$$

is a lower bound on the conditional state probability vector for states in $\mathscr{C}_i^k$.

To summarize, the bound spread reduction algorithm for states in $\mathscr{D}_i$ is as follows:

```
procedure    Bound Spread Reduction for $\mathscr{D}_i$
begin
  Based on the rate matrix $G_5$ in Section 4,
  for each pair $(c_i^j, c_i^{j-1})$ where $c_i^j, c_i^{j-1} \in \mathscr{C}_i$, compute the improved clone aggregate repair
      rate $r(c_i^j, c_i^{j-1})$ using Eq. (17);
  for each transition from a state $c_i^{L_i}$ to aggregate state $d'$, compute the improved clone
      aggregate repair rate $r(c_i^{L_i}, d')$ using Eq. (17);
  for each pair $(c_i^j, c_i^k)$ where $c_i^j, c_i^k \in \mathscr{C}_i$, compute the improved clone aggregate failure
      rate $r(c_i^j, c_i^k)$ using Eq. (18);
  compute an improved lower bound on the state probabilities by solving $G_5$ with the
      improved bounds on the aggregate transition rates
end
```

5.3. Bound Spread Reduction Algorithm. The above bound spread reduction procedure will give us improved lower bounds on the state probabilities for detailed states in $\mathscr{D}_i$ and clone states in $\mathscr{C}_i$ corresponding to step i of the bounding process. We can repeat the bound spread reduction procedure for detailed states and clone states corresponding to steps $i - 1, i - 2, \ldots, 1$ of the bounding process. Since we then obtain improved bounds for states in $\mathscr{D}_1$, we can go forward again and apply the multi-step bounding procedure to obtain tighter lower bounds for states corresponding to steps $2, 3, \ldots, i + 1$. The complete bound spread reduction procedure can therefore be stated as follows:

```
procedure Bound Spread Reduction Procedure
begin
  set $i$ = the index of the last forward step.
  for $j = i - 1$ to 1 do
  begin
    Apply the algorithm in Section 5.1 to tighten the bounds for clone states in $\mathscr{C}_j$;
    Apply the algorithm in Section 5.2 to tighter lower bounds for states in $\mathscr{D}_j$;
  end
  for $j = 1$ to $i$ do
  begin
    Apply the multi-step bounding algorithm described in Section 4 to obtain tighter lower
       bounds for states in $\mathscr{D}_j$;
  end
end
```

Clearly, the bound spread reduction procedure can be applied *repeatedly* to obtain tighter availability bounds. Also, each time we apply the bound spread reduction algorithm to improve the lower bounds on the stationary state probabilities for the clone states, the initial vector for the clone state probabilities can be set to the estimated fixed point solution vector from the previous bound spread reduction algorithm. Since this estimated fixed-point solution vector is a lower bound for the exact solution vector, by using it as a starting vector, we not only reduce the number of iterations but also preserve the monotonicity characteristics we claimed for the algorithm. In the following, we show that by using the estimated fixed-point solution vector from the previous bound spread reduction procedure, we preserve the monotonic convergence characteristic.

Corollary 5.2. *Using the estimated fixed point solution vector from the previous bound spread reduction procedure, the monotonic convergence characteristic is preserved.*

Proof. Let us define the following

b = the vector of conditional rates from $\mathscr{D}_{i+1}$ to $\mathscr{C}_i$ computed using the lower bound state probabilities $\pi_{\mathscr{D}_{i+1}^{L_{i+1}}}$.

$\pi'_{\mathscr{D}_{i+1}^{L_{i+1}}}$ = the vector of new lower bound state probabilities computed in the last step of the bound spread reduction procedure, where $\pi'_{\mathscr{D}_{i+1}^{L_{i+1}}} \geq \pi_{\mathscr{D}_{i+1}^{L_{i+1}}}$

$\mathbf{b}_2$ = the vector of conditional rates from $\mathscr{D}_{i+1}$ to $\mathscr{C}_i$ computed using $\pi'_{\mathscr{D}_{i+1}^{L_{i+1}}}$

In the previous step of the bound spread reduction procedure, we obtained improved lower bounds for the detailed states $\mathscr{D}_{i+1}$. This implies that we have improved lower bounds for the stationary state probabilities of states with L_{i+1}

failed components, $\pi'_{\mathscr{D}_{i+1}^{L_i+1}}$. Based on these improved lower bounds, we can obtain a new vector $\mathbf{b}_2$, which can be expressed as:

$$\mathbf{b}_2 = \mathbf{b} + \mathbf{b}'_2,$$

where

$$\mathbf{b}'_2 \geq \mathbf{0}$$

Let $\mathbf{x}^{(0)} = \tilde{\mathbf{x}}$ where $\tilde{\mathbf{x}}$ is the estimated fixed point solution from the previous iterative procedure. Since $\tilde{\mathbf{x}}$ is a lower bound state probability vector for the clone states, $\mathbf{x}^1$ can be expressed as (see Eq. (16)):

$$\mathbf{x}^{(1)} = \left[(\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{U}_\mathrm{A}\right]\tilde{\mathbf{x}} + (\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{b} + (\mathbf{D}_\mathrm{A} - \mathbf{L}_\mathrm{A})^{-1}\mathbf{b}'_2$$

in which $\mathbf{x}^1 \geq \tilde{\mathbf{x}}$. Note that the sum of the first two terms on the right-hand side must be greater than $\tilde{\mathbf{x}}$ by Theorem 5.1. The last term is nonnegative and therefore $\mathbf{x}^{(1)} \geq \tilde{\mathbf{x}}$. Applying Theorem 5.1, the result follows. □

Lastly, we estimate the cost of the bound spread reduction procedure after the kth step of the bounding process. The cost of one iteration of the bound spread reduction procedure is:

$$O\left(\sum_{j=1}^{k} k_j |\mathscr{D}_j|\right),$$

where k_j is a constant multiplier which is a function of the number of nonzero entries in the matrix $\mathscr{D}_j$ and the number of iterations for the algorithm to converge. In [Stewart and Goyal, 1985] it is reported that the number of iterations often ranges between 20 and 100. In our case, since the starting probability vector is the estimated fixed point solution vector computed from the previous step, in successive steps the number of iterations required for a specified tolerance can be expected to be significantly reduced in most cases.

Although the bound spread reduction algorithm can be applied repeatedly, there are diminishing returns in successive iterations. It is of interest then to estimate when it is better to repeat the bound spread reduction algorithm and when it is better to generate more of the transition rate matrix corresponding to unexplored states. This issue is discussed in the next section.

6. *Decision Criteria for Backward Iteration or Forward Generation*

Although tighter availability bounds can be obtained either by going forward (i.e., by generating more of the transition rate matrix) and applying the multi-step bounding algorithm or by going backward (i.e., reducing the errors accumulated in the previous steps) and applying the bound spread reduction algorithm, the computational cost and potential gain for these two choices can be quite different. For the multi-step bounding algorithm, we have to consider the following:

—computational cost of state generation,
—storage cost of the newly generated transition matrix,
—computational cost of evaluating steady-state probabilities for the detailed states.

For the bound spread reduction algorithm, all the transition matrices used are generated in the previous steps. Therefore, the only cost is the computation cost of the bound spread reduction process and the cost of retrieving transition matrices from secondary storage, if they do not all fit in main memory concurrently.

In order to decide which algorithm to apply, we have to also compare their respective *potential gains*. We define the potential gain as the fractional improvement in the spread between the upper and lower availability bounds. For the bound spread reduction algorithm, the potential gain comes from improved bounds on the detailed states and clone states. In the forward direction, the potential gain comes from the ability to obtain lower bounds for additional states. Although we can apply the bound spread reduction algorithm repeatedly to reduce the errors, the potential gain for each successive application exhibits diminishing returns. On the other hand, going forward will require generating more of the transition rate matrix. But since the distribution of state probabilities is skewed, these newly generated states may not make a significant contribution to bound reduction.

Based on the above discussion, we see that the problem of making an optimal decision is not trivial. One important requirement for the decision algorithm is that its computation cost should be much less than that of the multi-step bounding or the bound spread reduction algorithms themselves. Since we always obtain improved bounds regardless of the decision, the worst possible effect is some inefficiency. Although finding an optimal decision algorithm is an interesting theoretical issue, we conjecture that truly optimal decision algorithm will be very costly to implement and finding the optimal decision algorithm is beyond the scope of this paper. Nevertheless, a decision algorithm is required to implement the complete bounding methodology. The heuristic decision algorithm we present is not meant to be optimal in any sense. It is a simple, common sense heuristic that we have found to work well in practice. In the following, we described this heuristic decision criteria.

Let us define the following notation:

$s_f(i)$ = difference between the upper and lower bounds on availability after the ith application of the forward-step algorithm.
$s_b(i)$ = difference between the upper- and lower-bound availability after the ith application of the bound spread reduction algorithm.
s_c = current difference between the upper and lower bounds on availability.
q'_f = estimated gain if the multi-step bounding algorithm is applied (forward step).
g'_b = estimated gain if the bound spread reduction algorithm is applied (backward step).

In deciding whether to apply the forward step algorithm for the $(i + 1)$st time, we estimate the gain g'_f as follows:

$$g'_f = \frac{\left[s_f(i-1) - s_f(i)\right]}{s_f(i-1)} s_c \tag{19}$$

with the first term representing the fractional potential gain from the previous application of the forward step algorithm. In essence, g'_f estimates that, if the multi-step algorithm is applied, the gain will be the same fraction of the bound spread as was achieved by the previous step.

The estimated gain for going backward after the jth application of the bound spread reduction bounding algorithm, g'_b is as follows:

$$g'_b = \frac{[s_b(j-1) - s_b(j)]}{s_b(j-1)} s_c \qquad \text{for} \quad j \geq 1 \tag{20}$$

with $s_b(0) = s_c$. The first term represents the fractional potential gain for the previous iteration of the bound spread reduction algorithm and g'_b estimates that another iteration of the bound spread reduction algorithm will reduce the bound spread by the same fraction as the previous iteration.

In the following, we propose a simple heuristic for deciding the next step in the procedure. The algorithm is applied after each forward step. Note that one (backward) bound spread reduction is always applied before a decision is made.

```
procedure decision algorithm
begin
  Apply the bound spread reduction algorithm described in Section 5.3 for one iteration;
  Compute $g'_f$ and $g'_b$;
  while ($g'_f \leq g'_b$)
  begin
    Apply bound spread reduction algorithm described in Section 5.3 once;
    Compute $g'_f$ and $g'_b$
  end
  Apply the multi-step algorithm from Section 4
end
```

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 multi-step bounding algorithm. The decision algorithm cost is clearly trivial. A detailed example in Section 8 illustrates the application and effectiveness of the heuristic.

6.1. Global Bounding Algorithm. The global algorithm is as follows:

```
procedure Global Bounding Algorithm
begin
  $i = 1$;
  Based on the one-step algorithm described in Section 3, generate lower bounds on the
     stationary state probabilities for states in $\mathscr{F}_0$ through $\mathscr{F}_{H_1}$ and compute the system
     availability bounds;
  if (system availability bound is tight enough)
    stop
  while (system availability bound not tight enough) do
  begin
    $i = i + 1$;
    Based on the multi-step algorithm described in Section 4, generate the portion of the
       rate matrix corresponding to $\mathscr{F}_{L_i}$ to $\mathscr{F}_{H_i}$ (with $L_i = H_{i-1} + 1$) and compute the
       system availability bounds;
    if (system availability bound is tight enough)
      stop
    Apply Bound Spread Reduction algorithm described in 5.3 and compute the system
       availability bounds;
```

```
        if (system availability bound is tight enough)
            stop;
        compute $g'_f$ and $g'_b$;
        while ($g'_f \leq g'_b$)
        begin
            Apply Bound Spread Reduction algorithm described in 5.3 and compute the system
                availability bounds;
            if (system availability bound is tight enough)
                stop;
            recompute $g'_f$ and $g'_b$
        end
    end
end
```

Thus, the global bounding algorithm provides a method for piecewise generation of the transition matrix such that at each step, tighter system availability bounds can be obtained. One important note is that the algorithm can be terminated at any phase depending on the tightness of bounds required. This is due to the fact that at all times, we have bounds on the system availability.

7. *Partial State Generation*

In the previous sections, we developed an algorithm to compute bounds on the steady state availability bounds of repairable computer systems. The algorithm assumes a rate matrix generation process in which it is feasible to generate portions of the matrix in chunks which correspond to sets of states of the form $\cup_{i=L}^{H}\mathscr{F}_i$ for some L and H. However it may be that $|\mathscr{F}_i|$ can be too large to be solved in one step (e.g., for a model of a system with 100 components, $|\mathscr{F}_{10}| > 10^9$). When this situation occurs, $\mathscr{D}_t$ will have to be a proper subset of states in $\mathscr{F}_i$ for some i. In this section, we extend the algorithm so that it can accommodate this generality.

Assume we want to compute lower bounds for the state probabilities of states in $\{\mathscr{F}_0 \cup \mathscr{F}_1 \cup \cdots \mathscr{F}_K\}$ using the one-step bounding algorithm. Let G in Figure 1 be the original transition rate matrix. After applying cloning to all states in $\{\mathscr{F}_1 \cup \cdots \cup \mathscr{F}_K\}$, we obtain the transition rate matrix G_1 as depicted in Figure 11.

Assume that due to memory limitations, we can handle the portion of the rate matrix corresponding to $\mathscr{D} = \{\mathscr{F}_0 \cup \cdots \cup \mathscr{F}_{K-1} \cup \mathscr{F}'_K\}$, where $\mathscr{F}'_K$ is a proper subset of $\mathscr{F}_K$. Let $\mathscr{F}''_K = \{\mathscr{F}_K - \mathscr{F}'_K\}$ where the corresponding transition rate matrix G_2 is depicted in Figure 12.

In order to compute lower bounds for the state probabilities of states in the set $\mathscr{D}_t$, we transform the rate matrix G_2 to G_3 (depicted in Figure 13). This transformation has the following probabilistic interpretation. Whenever there is a transition from state j in $\mathscr{D}_t$ to state k in $\mathscr{F}''_K$, this is made a transition from state j in $\mathscr{D}_t$ to state k in $\mathscr{C}''_K$ instead. Since we apply cloning to states in $\{\mathscr{F}_1 \cup \cdots \cup \mathscr{F}_K\}$, there is a natural one-to-one mapping from any state in $\mathscr{D}_t$ to its clone state.

As in previous sections, let $\pi_{\mathscr{D}/G_i}$ be the vector of stationary probabilities of states in $\mathscr{D}$ with rate matrix G_i. The following theorem shows that the solution of the transformed matrix provides lower bounds for the state probabilities of states in $\mathscr{D}$.

$$\left[\begin{array}{ccccc|cccc|ccc}
Q_{00} & Q_{01} & \cdots & Q_{0,K-1} & Q_{0,K} & 0 & \cdots & 0 & 0 & \cdots & Q_{0,N-1} & Q_{0,N} \\
Q_{10} & Q_{11} & \cdots & Q_{1,K-1} & Q_{1,K} & 0 & \cdots & 0 & 0 & \cdots & Q_{1\ N-1} & Q_{1,N} \\
0 & Q_{21} & \cdots & Q_{2,K-1} & Q_{2,K} & 0 & \cdots & 0 & 0 & \cdots & Q_{2,N-1} & Q_{2,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & 0 & \cdots & 0 & 0 & \cdots & \vdots & \vdots \\
0 & 0 & \cdots & Q_{K,K-1} & Q_{K,K} & 0 & \cdots & 0 & 0 & \cdots & Q_{K,N-1} & Q_{K,N} \\
\hline
Q_{10} & 0 & \cdots & 0 & 0 & Q_{11} & \cdots & Q_{1,K-1} & Q_{1,K} & \cdots & Q_{1,N-1} & Q_{1,N} \\
0 & 0 & \cdots & 0 & 0 & Q_{21} & \cdots & Q_{2,K-1} & Q_{2,K} & \cdots & Q_{2,N-1} & Q_{2,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{K,K-1} & Q_{K,K} & \cdots & Q_{K,N-1} & Q_{K,N} \\
\hline
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & Q_{K+1,K} & \cdots & Q_{K+1,N-1} & Q_{K+1,N} \\
 & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & Q_{N,N-1} & Q_{N,N}
\end{array}\right]$$

FIG. 11. Transition matrix G_1.

$$\left[\begin{array}{ccccc|c|ccccc|ccc}
Q_{00} & Q_{01} & \cdots & Q_{0,K-1} & Q_{0,K'} & Q_{0,K''} & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{0,N-1} & Q_{0,N} \\
Q_{10} & Q_{11} & \cdots & Q_{1,K-1} & Q_{1,K'} & Q_{1,K''} & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{1,N-1} & Q_{1,N} \\
0 & Q_{21} & \cdots & Q_{2,K-1} & Q_{2,K'} & Q_{2,K''} & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{2,N-1} & Q_{2,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & Q_{K',K-1} & Q_{K',K'} & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{K',N-1} & Q_{K',N} \\
\hline
0 & 0 & \cdots & Q_{K'',K-1} & 0 & Q_{K'',K''} & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{K'',N-1} & Q_{K'',N} \\
Q_{10} & 0 & \cdots & 0 & 0 & 0 & Q_{11} & \cdots & Q_{1,K-1} & Q_{1,K'} & Q_{1,K''} & \cdots & Q_{1,N-1} & Q_{1,N} \\
0 & 0 & \cdots & 0 & 0 & 0 & Q_{21} & \cdots & Q_{2,K-1} & Q_{2,K'} & Q_{2,K''} & \cdots & Q_{2,N-1} & Q_{2,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & 0 & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & Q_{K',K-1} & Q_{K',K'} & 0 & \cdots & Q_{K',N-1} & Q_{K',N} \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & Q_{K'',K-1} & 0 & Q_{K'',K''} & \cdots & Q_{K'',N-1} & Q_{K'',N} \\
\hline
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & Q_{K+1,K'} & Q_{K+1,K''} & \cdots & Q_{K+1,N-1} & Q_{K+1,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{N,N-1} & Q_{N,N}
\end{array}\right]$$

FIG. 12. Transition matrix G_2.

$$\left[\begin{array}{ccccc|ccccc|ccc}
Q_{00} & Q_{01} & \cdots & Q_{0,K-1} & Q_{0,K'} & 0 & \cdots & 0 & 0 & Q_{0,K''} & \cdots & Q_{0,N-1} & Q_{0,N} \\
Q_{10} & Q_{11} & \cdots & Q_{1,K-1} & Q_{1,K'} & 0 & \cdots & 0 & 0 & Q_{1,K''} & \cdots & Q_{1,N-1} & Q_{1,N} \\
0 & Q_{21} & \cdots & Q_{2,K-1} & Q_{2,K'} & 0 & \cdots & 0 & 0 & Q_{2,K''} & \cdots & Q_{2,N-1} & Q_{2,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & Q_{K',K-1} & Q_{K',K'} & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{K',N-1} & Q_{K',N} \\
\hline
Q_{10} & 0 & \cdots & 0 & 0 & Q_{11} & \cdots & Q_{1,K-1} & Q_{1,K'} & Q_{1,K''} & \cdots & Q_{1,N-1} & Q_{1,N} \\
0 & 0 & \cdots & 0 & 0 & Q_{21} & \cdots & Q_{2,K-1} & Q_{2,K'} & Q_{2,K''} & \cdots & Q_{2,N-1} & Q_{2,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{K',K-1} & Q_{K',K'} & 0 & \cdots & Q_{K',N-1} & Q_{K',N} \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{K'',K-1} & 0 & Q_{K'',K''} & \cdots & Q_{K'',N-1} & Q_{K'',N} \\
\hline
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & Q_{K+1,K'} & Q_{K+1,K''} & \cdots & Q_{K+1,N-1} & Q_{K+1,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & Q_{N,N-1} & Q_{N,N}
\end{array}\right]$$

FIG. 13. Transition matrix G_3.

THEOREM 7.1. $\pi_{\mathscr{D}/G_3} \leq \pi_{\mathscr{D}/G_2}$

PROOF. First, we define the following set of states:

$$\begin{aligned}
\mathscr{D} &= \{\mathscr{F}_0 \cup \cdots \cup \mathscr{F}_{K-1} \cup \mathscr{F}'_K\},\\
d' &= \mathscr{F}_0,\\
\mathscr{D}' &= \mathscr{D} - d',\\
\mathscr{F}''_K &= \mathscr{F}_K - \mathscr{F}'_K.
\end{aligned}$$

It is easy to observe that the following equations hold:

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

$$\pi_{\mathscr{F}''_K/G_2} \geq \mathbf{0}$$

The flow conservation equation for states in $\mathscr{D}'$ for both G_2 and G_3 are:

$$\pi_{\mathscr{D}'/G_2} = \left(\pi_{d'/G_2} Q_{d'\mathscr{D}'} + \pi_{\mathscr{F}''_K/G_2} Q_{\mathscr{F}''_K\mathscr{D}'}\right)[-Q_{\mathscr{D}'\mathscr{D}'}]^{-1},$$

$$\pi_{\mathscr{D}'/G_3} = \left(\pi_{d'/G_3} Q_{d'\mathscr{D}'}\right)[-Q_{\mathscr{D}'\mathscr{D}'}]^{-1}.$$

Since $(-Q_{\mathscr{D}'\mathscr{D}'})$ is an M-matrix, it's inverse is a nonnegative matrix, therefore $\pi_{\mathscr{D}/G_3} \leq \pi_{\mathscr{D}/G_2}$. □

We can now transform G_3 to G_4 (depicted in Figure 14) by exact aggregation. We aggregate all the states in $\mathscr{F}_i$ for $i > K$ and all clone states according to number of failed components. Since we apply exact aggregation, we have $\pi_{\mathscr{D}/G_3} = \pi_{\mathscr{D}/G_4}$.

At this point, we can apply Theorem 3.1 and replace the aggregate transition rates by appropriate *bounds*. The final transition rate matrix, G_5 is given in Figure 15. It is easy to see that $\pi_{\mathscr{D}/G_5} \leq \pi_{\mathscr{D}/G_4}$ and therefore $\pi_{\mathscr{D}/G_5} \leq \pi_{\mathscr{D}/G}$. With this modification, we overcome the partial state generation problem, and we also maintain the block Hessenberg property.

For the multi-step bounding algorithm, we may encounter either of the following two situations:

(1) we can generate all the $\mathscr{F}''_K$ from the previous step and perhaps some states with more failures,
(2) we can only generate a proper subset of $\mathscr{F}''_K$.

At the ith step, we can apply basically the same transformations we used in the one-step bounding algorithm discussed earlier in this section. The modification are:

(1) Let $\mathscr{F}''_K$ be the states in $\mathscr{F}_K$ that are not in $\mathscr{D}_i$. All transitions from $\mathscr{D}_i$ to any states in $\mathscr{F}''_K$ are modified to become transitions to the corresponding clone states.
(2) Computer upper and lower bound transition rates for the aggregate and the clone states.

It is easy to show that this modification provides a transition rate matrix whose solution vector provides lower bound steady state probabilities for the states in $\mathscr{D}_i$.

For the bound spread reduction process, we see that to refine the errors for clone states that have K or less failures, we have to have lower bounds for the

$$\left[\begin{array}{ccccc|cccc|ccc}
Q_{00} & Q_{01} & \cdot\cdot & Q_{0,K-1} & Q_{0,K'} & 0 & \cdot\cdot & 0 & Q_{0,K''} & \cdot\cdot & Q_{0,N-1} & Q_{0,N} \\
Q_{10} & Q_{11} & \cdot\cdot & Q_{1,K-1} & Q_{1,K'} & 0 & \cdot\cdot & 0 & Q_{1,K''} & \cdot\cdot & Q_{1,N-1} & Q_{1,N} \\
0 & Q_{21} & \cdot\cdot & Q_{2,K-1} & Q_{2,K'} & 0 & \cdot\cdot & 0 & Q_{2,K''} & \cdot\cdot & Q_{2,N-1} & Q_{2,N} \\
\vdots & \vdots & \cdot\cdot & \vdots & \vdots & \vdots & \cdot\cdot & \vdots & \vdots & \cdot\cdot & \vdots & \vdots \\
0 & 0 & \cdot\cdot & Q_{K',K-1} & Q_{K',K'} & 0 & \cdot\cdot & 0 & 0 & \cdot\cdot & Q_{K',N-1} & Q_{K',N} \\
\hline
r_{10} & 0 & \cdot\cdot & 0 & 0 & \bullet & \cdot\cdot & r_{1,K-1} & r_{1,K} & \cdot\cdot & r_{1,N-1} & r_{1,N} \\
0 & 0 & \cdot\cdot & 0 & 0 & r_{21} & \cdot\cdot & r_{2,K-1} & r_{2,K} & \cdot\cdot & r_{2,N-1} & r_{2,N} \\
\vdots & \vdots & \cdot\cdot & \vdots & \vdots & \vdots & \cdot\cdot & \vdots & \vdots & \cdot\cdot & \vdots & \vdots \\
0 & 0 & \cdot\cdot & 0 & 0 & 0 & \cdot\cdot & r_{K,K-1} & \bullet & \cdot\cdot & r_{K',N-1} & r_{K,N} \\
\hline
0 & 0 & \cdot & 0 & 0 & 0 & \cdot\cdot & 0 & r_{K+1,K} & \cdot\cdot & r_{K+1,N-1} & r_{K+1,N} \\
\vdots & \vdots & \cdot\cdot & \vdots & \vdots & \vdots & \cdot\cdot & \vdots & \vdots & \cdot\cdot & \vdots & \vdots \\
0 & 0 & \cdot\cdot & 0 & 0 & 0 & \cdot\cdot & 0 & 0 & \cdot\cdot & r_{N,N-1} & \bullet
\end{array}\right]$$

FIG. 14. Transition matrix G_4.

$$\left[\begin{array}{ccccc|cccc|ccc}
Q_{00} & Q_{01} & \cdot\cdot & Q_{0,K-1} & Q_{0,K'} & 0 & \cdot\cdot & 0 & Q_{0,K''} & \cdot\cdot & Q_{0,N-1} & Q_{0,N} \\
Q_{10} & Q_{11} & \cdot\cdot & Q_{1,K-1} & Q_{1,K'} & 0 & \cdot\cdot & 0 & Q_{1,K''} & \cdot\cdot & Q_{1,N-1} & Q_{1,N} \\
0 & Q_{21} & \cdot\cdot & Q_{2,K-1} & Q_{2,K'} & 0 & \cdot\cdot & 0 & Q_{2,K''} & \cdot\cdot & Q_{2,N-1} & Q_{2,N} \\
\vdots & \vdots & \cdot\cdot & \vdots & \vdots & \vdots & \cdot\cdot & \vdots & \vdots & \cdot\cdot & \vdots & \vdots \\
0 & 0 & \cdot\cdot & Q_{K',K-1} & Q_{K',K'} & 0 & \cdot\cdot & 0 & 0 & \cdot\cdot & Q_{K',N-1} & Q_{K',N} \\
\hline
- & 0 & \cdot\cdot & 0 & 0 & \bullet & \cdot\cdot & + & + & \cdot\cdot & + & + \\
0 & 0 & \cdot\cdot & 0 & 0 & - & \cdot\cdot & + & + & \cdot\cdot & + & + \\
\vdots & \vdots & \cdot\cdot & \vdots & \vdots & \vdots & \cdot\cdot & \vdots & \vdots & \cdot\cdot & \vdots & \vdots \\
0 & 0 & \cdot\cdot & 0 & 0 & 0 & \cdot\cdot & - & \bullet & \cdot\cdot & + & + \\
\hline
0 & 0 & \cdot\cdot & 0 & 0 & 0 & \cdot\cdot & 0 & - & \bullet & + & + \\
\vdots & \vdots & \cdot\cdot & \vdots & \vdots & \vdots & \cdot\cdot & \vdots & \vdots & \cdot\cdot & \vdots & \vdots \\
0 & 0 & \cdot\cdot & 0 & 0 & 0 & \cdot\cdot & 0 & 0 & \cdot\cdot & - & \bullet
\end{array}\right]$$

FIG. 15. Transition matrix G_5.

state probabilities for all the states with $K + 1$ failures. We can obtain lower bounds on the state probabilities for the states with $K + 1$ failures using the multi-step algorithm. Once we have the lower bounds on the state probabilities for the states with $K + 1$ failures, the iterative refinement process described in Section 5 can be applied.

8. *Example*

In this section, we present an example to illustrate the application of the bounding algorithm. The example is a heterogeneous distributed database system as depicted in Figure 16. The components of this system are: two front-ends, four databases, and four processing subsystems each consisting of a switch, a memory and two processors. Components may fail and be repaired according to the rates given in Table I. If either processor of subsystem A or B fails, it has a 0.05 probability of contaminating both database A and database B. (To contaminate the database means to leave it in a inconsistent state, and in this case the database is considered to have "failed". Repair of the database requires the execution of a recovery procedure.) If either processor of

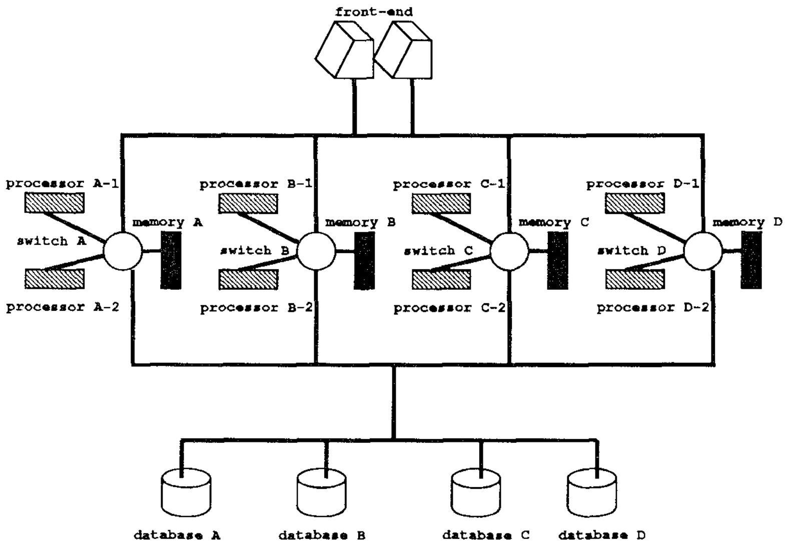

FIG. 16. A fault-tolerant heterogeneous distributed database system.

TABLE I. FAILURE AND REPAIR RATES (PER HOUR)

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 D	1/600	2.1
Memory D	1/600	2.5
Database A	1/5500	2.5
Database B	1/5000	2.2
Database C	1/5000	2.5
Database D	1/4500	2.3

TABLE II. UPPER AND LOWER BOUNDS ON STEADY STATE AVAILABILITY OF THE DATABASE SYSTEM

		Availability	Spread in
Step	Algorithm	(Upper Bound	Availability
Number	Applied	Lower Bound)	Bounds
1	one-step	0.986456955373 0.999999999655	0.013543044282
2	multi-step (one application)	0.990023127431 0.999999999029	0.009976869598
3	bound spread reduction (applied twice)	0.999995763421 0.999999987643	0.000004224222
4	multi-step (one application)	0.999999246581 0.999999975211	0.000000728630
5	bound spread reduction (applied twice)	0.999999952345 0.999999952972	0.000000000627

subsystem C or D fails, it has a 0.05 probability of contaminating both database C and database D. Components are repaired by a single repair facility that gives preemptive priority to components in the order: front-ends, databases, switches, memories, processor set 1, and, lastly, processor set 2. (Ties are broken by random selection.) The database system is considered operational if 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 nonoperational.

In Table II, we present the bounds on steady state availability for several steps of the bounding procedure. We note that in each step, the bounds are significantly tightened. In Step (1), we apply the one-step bounding algorithm with detailed states that have 0 to 2 failed components. The Markov chain that has to be solved in this step has 276 states. In Step (2), we apply the multi-step bounding algorithm for detailed states that have between 3 and 4 failed components. In this step, we have to solve a model with 8,876 states. In Step (3), we apply the bound spread reduction algorithm described in Section 5.3 for states that have between 0 and 4 failed components. In Step (4), we apply the multi-step bounding algorithm with detailed states that have between 5 and 6 failed components. The number of states in the model at this step is 100,966. In Step (5), we apply the bound spread reduction algorithm for states that have between 1 and 6 failed components.

In Table III and Table IV, we illustrate the individual contributions to the bound spread reduction by the clone states and detailed states when the bound spread reduction algorithm is applied. This data shows that most of the improvement comes from recouping some of the unclaimed reward of the clone states. As expected, the majority of the gain comes from obtaining lower bounds on the clone states when the bound spread reduction is applied for the first time.

TABLE III. CONTRIBUTION BY CLONES STATES AND DETAIL STATES IN STEP (3)

total reduction of bound spread in step 3	9.9726×10^{-3}
contribution by clone states $\mathcal{C}_1 \cup \mathcal{C}_2$	8.2304×10^{-3}
contribution by detailed states $\mathcal{D}_0 \cup \cdots \cup \mathcal{D}_4$	1.7422×10^{-3}

TABLE IV. CONTRIBUTION BY CLONES STATES AND DETAILED STATES IN STEP (5)

total reduction of bound spread in step 5	7.2800×10^{-7}
contribution by clone states $\mathcal{C}_1 \cup \mathcal{C}_2$	5.0818×10^{-8}
contribution by clone states $\mathcal{C}_3 \cup \mathcal{C}_4$	5.8400×10^{-7}
contribution by detailed states $\mathcal{D}_0 \cup \cdots \cup \mathcal{D}_2$	3.5391×10^{-9}
contribution by detailed states $\mathcal{D}_3 \cup \mathcal{D}_4$	7.3522×10^{-8}
contribution by detailed states $\mathcal{D}_5 \cup \mathcal{D}_6$	1.6122×10^{-8}

TABLE V. DECISION ALGORITHM ILLUSTRATION

g'_f	g'_b	Algorithm Applied	Resulting Spread in Availability Bounds	Algorithm Applied	Resulting Spread in Availability Bounds
—	—	bound spread reduction	5.6821×10^{-6}	multi-step bounding	9.9764×10^{-3}
1.4962×10^{-6}	5.6788×10^{-6}	bound spread reduction	4.2242×10^{-6}	multi-step bounding	4.4781×10^{-6}
1.1126×10^{-6}	1.0838×10^{-6}	multi-step reduction	7.2863×10^{-7}	bound spread reduction	3.7822×10^{-6}

Table V shows details of the reduction in the spread of the availability bounds in Step (3) of the global algorithm. The first two columns show the estimated gain in each direction, the third and fourth columns show the spread in the availability bounds if we follow the decision algorithm. The last two columns show the spread in the availability bounds if the decision algorithm is not followed (this is recorded just to illustrate what would have happened if the opposite decision was made.) In the first row, we observe that by applying the bound spread reduction algorithm, the spread in the availability bounds is significantly reduced compared to the spread in the availability bounds if the multi-step bounding (or forward) algorithm is applied. In the second row, since the estimated gain g'_b is greater than g'_f, we apply the bound spread reduction algorithm again. The reduction in the spread of the availability bounds is comparable in either direction, but since the cost of forward generation is higher, it pays to apply the bound spread reduction algorithm. In the last row (which corresponds to Step (4) of the global algorithm), since the estimated gain g'_f is greater than g'_b, we apply the multi-step bounding algorithm, and we obtain a significant reduction in the spread of the availability bounds.

9. *Conclusion*

We have developed a methodology for computing bounds on the steady state availability of repairable computer systems. The method provides an efficient computational procedure to overcome the large state space problem in evaluating steady state availability of realistic systems. We showed that by modifying the original model, bounds can be obtained by solving a sequence of smaller models, each of which has a state space cardinality chosen to be small enough to match memory limitations. The method also supports trading off tightness of the bounds against computational effort.

The development in the paper is couched in terms of models of repairable computer systems and determination of bounds on availability. However, the methods appear to have promise for other applications. The important property of availability models that is used is that the equilibrium state probabilities are concentrated in very few states. It is reasonable to expect that this same property will hold for example, in models of probabilistic protocol evaluation [Dimitrijevic and Chen, 1988; Maxemchuk and Sabnani, 1987] and load balancing. In the case of load balancing, the routing policy is designed to balance the load on the resources in the system. Thus, we expect that a large number of possible states 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.

Appendix

LEMMA A.1. *Let G be the generator matrix for an irreducible Markov chain with the upper Hessenberg form as illustrated in Figure* 17. *Let G′ be a matrix with the same structure as G, but where* $0 < g'_{i,i-1} \le g_{i,i-1}$ *and* $g'_{i,j} \ge g_{i,j}$ *for* $0 \le i < j \le N$. *G′ is illustrated in Figure* 18, *where the non-zero lower bound rates are denoted by "−" and the upper bound rates are denoted by "+". Then* $\pi_{0/G'} \le \pi_{0/G}$.

PROOF. The proof is by contradiction.

Assume that $\pi_{0/G'} > \pi_{0/G}$. Consider the flow balance equations, for each Markov chain, which equate the flow between the set of states numbered 0 to $i - 1$ and the set of states numbered i to N.

$$\pi_{i/G} g_{i,i-1} = \sum_{j=0}^{i-1} \sum_{k=i}^{N} \pi_{j/G} g_{j,i} \tag{A.1}$$

$$\pi_{i/G'} g'_{i,i-1} = \sum_{j=0}^{i-1} \sum_{k=i}^{N} \pi_{j/G'} g'_{j,i} \tag{A.2}$$

It is a simple induction argument using the above equations to show that if, $\pi_{j/G'} > \pi_{j/G}$ for $j < i$, then $\pi_{i/G'} > \pi_{i/G}$. But then, from the assumption that $\pi_{0/G'} > \pi_{0/G}$, it follows that

$$\sum_{j=0}^{N} \pi_{j/G'} > \sum_{j=0}^{N} \pi_{j/G}. \tag{A.3}$$

This is impossible since both sums must equal 1. We conclude that it is not possible for $\pi_{0/G'} > \pi_{0/G}$, and the lemma is proved. □

FIG. 17. Transition matrix G.

$$G = \begin{bmatrix} \bullet & g_{01} & g_{02} & g_{03} & \cdots & g_{0N} \\ g_{10} & \bullet & g_{12} & g_{13} & \cdots & g_{1N} \\ 0 & g_{21} & \bullet & g_{23} & \cdot & \vdots \\ 0 & 0 & g_{32} & \bullet & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & \cdot & 0 & g_{N,N-1} & \bullet \end{bmatrix}$$

FIG. 18. Transition matrix G'.

$$G' = \begin{bmatrix} \bullet & + & + & + & \cdot\cdot & + \\ - & \bullet & + & + & \cdots & + \\ 0 & - & \bullet & + & \ddots & \vdots \\ 0 & 0 & - & \bullet & \ddots & \vdots \\ \cdot & & \cdot & \ddots & \ddots & \\ 0 & \cdots & \cdots & 0 & - & \bullet \end{bmatrix}$$

$$G = \begin{bmatrix} G_{\mathcal{U},\mathcal{U}} & G_{\mathcal{U},\mathcal{V}} \\ G_{\mathcal{V},\mathcal{U}} & G_{\mathcal{V},\mathcal{V}} \end{bmatrix}$$

$$= \left[\begin{array}{ccccc|cccccc} & & G_{\mathcal{U},\mathcal{U}} & & & & & & G_{\mathcal{U},\mathcal{V}} & & \\ \hline 0 & \cdots & g_{v_0,u_k} & \cdots & 0 & \bullet & g_{v_0,v_1} & g_{v_0,v_2} & g_{v_0,v_3} & \cdot\cdot & g_{v_0,v_{K_v}} \\ 0 & \cdots & 0 & \cdots & 0 & g_{v_1,v_0} & \bullet & g_{v_1,v_2} & g_{v_1,v_3} & \cdots & g_{v_1,v_{K_v}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & q_{v_2,v_1} & \bullet & q_{v_2,v_3} & \ddots & \vdots \\ \cdot & \vdots & \vdots & \vdots & \vdots & 0 & 0 & g_{32} & \bullet & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & \cdots & \cdots & 0 & g_{v_{K_v},v_{K_v-1}} & \bullet \end{array}\right]$$

FIG. 19. Transition matrix G.

LEMMA A.2. *Let G be the generator matrix for an irreducible Markov chain as illustrated in Figure* 19. *Let the state space be partitioned into two subsets, $\mathcal{U}$ and $\mathcal{V}$. As illustrated in Figure* 19, *the generator matrix is conformally partitioned. Let the states in $\mathcal{U}$ be denoted $u_0, u_1, \ldots, u_{K_u}$ and the states in $\mathcal{V}$ be denoted $v_0, v_1, \ldots, v_{K_v}$. As illustrated in Figure* 19, *$G_{\mathcal{V},\mathcal{U}}$ has all zero entries except for one non-zero transition rate from v_0 to some state u_k, in $\mathcal{U}$. Also, $G_{\mathcal{V},\mathcal{V}}$ has the upper Hessenberg form.*

Let G' be defined as equal to G except that (*see Figure* 20):

$$0 < g'_{v_0,u_k} \leq g_{v_0,u_k}$$

$$0 < g'_{v_i,v_{i-1}} \leq g_{v_i,v_{i-1}}$$

$$g'_{v_i,v_j} \geq g_{v_i,v_j}, \qquad K_v \geq j > i \geq 0$$

$$G'_{\mathcal{U},\mathcal{V}} \geq G_{\mathcal{U},\mathcal{V}}$$

Then, $\pi_{\mathcal{U}/G'} \leq \pi_{\mathcal{U}/G}$.

$$G' = \begin{bmatrix} G'_{\mathcal{U},\mathcal{U}} & G'_{\mathcal{U},\mathcal{V}} \\ G'_{\mathcal{V},\mathcal{U}} & G'_{\mathcal{V},\mathcal{V}} \end{bmatrix} = \left[\begin{array}{c|c} G'_{\mathcal{U},\mathcal{U}}\,(= G_{\mathcal{U},\mathcal{U}}) & \begin{array}{cccccc} + & + & \cdots & \cdots & + & + \\ \vdots & \vdots & & & & \\ + & + & \cdots & \cdots & + & + \end{array} \\ \hline \begin{array}{ccccc} 0 & \cdots & - & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{array} & \begin{array}{cccccc} \bullet & + & + & + & \cdots & + \\ - & \bullet & + & + & \cdots & + \\ 0 & - & \bullet & + & \ddots & \vdots \\ 0 & 0 & - & \bullet & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & \cdots & 0 & - & \bullet \end{array} \end{array}\right]$$

FIG. 20. Transition matrix G'.

$$G_{agg} = \begin{bmatrix} \bullet & q_{a,v_0} & q_{a,v_1} & \cdots & \cdots & \cdots & q_{a,v_{K_v}} \\ q_{v_o,a} & \bullet & q_{v_0,v_1} & q_{v_0,v_2} & q_{v_0,v_3} & \cdots & q_{v_0,v_{K_v}} \\ 0 & q_{v_1,v_0} & \bullet & q_{v_1,v_2} & q_{v_1,v_3} & \cdots & q_{v_1,v_{K_v}} \\ 0 & 0 & q_{v_2,v_1} & \bullet & q_{v_2,v_3} & \ddots & \vdots \\ 0 & 0 & 0 & q_{32} & \bullet & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & q_{v_{K_v},v_{K_v-1}} & \bullet \end{bmatrix}$$

FIG. 21. Transition matrix G_{agg}.

PROOF. For any vector x, let $\hat{x}$ denote the normalized version of x which is scaled so that the sum of the elements is 1.

Since (1) there is only a single state, call it u_k, in $\mathcal{U}$ by which the system can enter $\mathcal{U}$ from $\mathcal{V}$, and (2) $G_{\mathcal{U},\mathcal{U}} = G'_{\mathcal{U},\mathcal{U}}$, it follows that $\hat{\pi}_{\mathcal{U}/G'} = \hat{\pi}_{\mathcal{U}/G}$ [Courtois and Semal, 1986].

Based on the equality of the conditional state probabilities for $\mathcal{U}$ and $\mathcal{U}'$, it follows immediately that $\pi_{\mathcal{U}/G'} \leq \pi_{\mathcal{U}/G}$ if and only if $\pi_{\mathcal{U}/G'}\mathbf{1} \leq \pi_{\mathcal{U}/G}\mathbf{1}$.

Assuming that we know the conditional state probabilities for $\mathcal{U}$, we can aggregate the states in $\mathcal{U}$ in both G and G'. The aggregated transition rate matrices are illustrated in Figures 21 and 22. Let a denote the aggregate state. We note that since $\hat{\pi}_{\mathcal{U}/G'} = \hat{\pi}_{\mathcal{U}/G}$ and $G'_{\mathcal{U},\mathcal{V}} \geq G_{\mathcal{U},\mathcal{V}}$ it follows that $g'_{a,v_j} \geq g_{a,v_j}$, for $0 \leq j \leq K_v$ and therefore the aggregate matrices satisfy the conditions of Lemma 1. Applying Lemma A.1 we have $\pi_{a/G'_{\text{agg}}} \leq \pi_{a/G_{\text{agg}}}$. But since we assumed exact aggregation $\pi_{a'/G'_{\text{agg}}} = \pi_{\mathcal{U}'/G'}\mathbf{1}$ and $\pi_{a/G_{\text{agg}}} = \pi_{\mathcal{U}/G}\mathbf{1}$. It follows that $\pi_{\mathcal{U}'/G'}\mathbf{1} \leq \pi_{\mathcal{U}/G}\mathbf{1}$, and we are done. □

ACKNOWLEDGMENTS. The authors wish to thank the anonymous referees for their comments and suggestions that substantially improved the quality of this paper.

FIG. 22. Transition matrix G'_{agg}.

$$G'_{agg} = \begin{bmatrix} \bullet & + & + & .. & \cdots & \cdots & + \\ - & \bullet & + & + & + & \cdots & + \\ 0 & - & \bullet & + & + & \cdots & + \\ 0 & 0 & - & \bullet & + & \ddots & \vdots \\ 0 & 0 & 0 & - & \bullet & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & - & \bullet \end{bmatrix}$$

REFERENCES

NOTE: The following references are not cited in the text: [COURTOIS AND SEMAL 1984]; [KEMENY AND SNELL, 1969]; [NEUTS, 1981]; [ROSS, 1983]; [TAKAHASHI, 1972, 1975]; AND [VAN DIJK 1988, 1989].

BERMAN, A., AND PLEMMONS, R. J. 1979. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, Orlando, Fla.

BURDEN, R. L., AND FAIRES, J. D. 1989. *Numerical Analysis*. PWS-KENT Publishing Company, Boston, Mass.

CARRASCO, J. A., AND FIGUERAS, J. 1986. METFAC: Design and implementation of a software toll for modeling and evaluation of complex fault-tolerant computing systems. In *Proceedings of the 16th Annual International Symposium on Fault-Tolerant Computing Systems (FTCS-16)* (Vienna, Austria, July). IEEE Computer Society Press, Washington, D.C., pp. 424–429.

CONWAY, A. E., AND GOYAL, A. 1987. Monte Carlo simulation of computer system availability and reliability models. In *Proceedings of the 17th Annual International Symposium on Fault-Tolerant Computing Systems (FTCS-17)* (June). IEEE Computer Society Press, Washington, D.C., pp. 230–235.

COSTES, A., DOUCET, J. E., LANDRAULT, C., AND LAPRIE, J. C. 1981. SURF: A program for dependability evaluation of complex fault-tolerant computing systems. In *Proceedings of the 11th Annual International Symposium on Fault-Tolerant Computing Systems (FTCS-11)*. (June). IEEE, Computer Society Press, Washington, D.C., pp. 72–78.

COURTOIS, P. J. 1977. *Decomposability—Queueing and Computer System Applications*. Academic Press, Orlando, Fla.

COURTOIS, P. J. 1982. Error Minimization in Decomposable Stochastic Models. In *Applied Probability—Computer Science: The Interface Vol. I*. R. L. Disney and T. Ott, eds. Birkhauser.

COURTOIS, P. J., AND SEMAL, P. 1984. Bounds for the positive eigenvectors of nonnegative matrices and for their approximations by decomposition. *J. ACM 31*, 4 (Oct.), 804–825.

COURTOIS, P. J., AND SEMAL, P. 1986 Computable bounds for conditional steady-state probabilities in large markov chains and queueing models. *J. Select. Areas Commun. 4*, 6 (Sept.), 926–937.

DE SOUZA E SILVA, E., AND GAIL, H. R. 1986. Calculating cumulative operational time distributions of repairable computer systems. *IEEE Trans. Comput. (Special Issue on Fault Tolerant Computing), C-35*, 4 (Apr.), 322–332.

DE SOUZA E SILVA, E., AND GAIL, H. R. 1989. Calculating availability and performability measures of repairable computer systems using randomization. *J. ACM 36*, 1 (Jan.), 171–193.

DIMITRIJEVIC, D. D., AND CHEN, M. 1988. An integrated algorithm for probabilistic protocol verification and evaluation. IBM Tech. Rep, RC 13001. T. J. Watson Research Center, IBM, Yorktown Heights, N.Y.

GEIST, R., AND TRIVEDI, K. S. 1985. Ultra-High Reliability Prediction for Fault-Tolerant Computer Systems. *IEEE Trans. Comput. C-32*, 12 (Dec.), 1118–1127.

GOYAL, A., CARTER, W. C., DE SOUZA E SILVA, E., LAVENBERG, S. S., AND TRIVEDI, K. S. 1986. The System Availability Estimator. In *Proceedings of the 16th Annual International Symposium on Fault-Tolerant Computing Systems (FTCS-16)* (Vienna, Austria, (July). pp. 84–89.

GOYAL, A., LAVENBERG, S. S., AND TRIVEDI, K. S. 1986. Probabilistic modeling of computer system availability. *Ann. Oper. Res. 8* (1986), 285–306.

HEIDELBERGER, AND GOYAL, A. 1987. Sensitivity Analysis of Continuous Time Markov Chains Using Uniformization. In *Proceedings of the 2nd International Workshop on Applied Mathematics and Performance/Reliabilty Models of Computer/Communication Systems* (Rome, Italy, May) North Holland, Amsterdam, The Netherlands.

KEMENY, J. G., AND SNELL, J. L. 1960. *Finite Markov Chains*. Van Nostrand Company. Princeton, N.J.

LEWIS, E. E., AND BOHM, F. 1989. Monte Carlo simulation of Markov unreliability models. *Nucl. Eng. Des. 77*, 1, 49–62.

LUI, J. C. S., AND MUNTZ, R. R. 1991. Evaluating Bounds on Steady State Availability from Markov Models of Repairable Systems. In *Numerical Solution of Markov Chains*, William J. Stewart, ed. Marcel Dekker, New York, pp. 435–454.

MAKAM, S. V., AND AVIZIENIS, A. 1982. ARIES 81: A reliability and life-cycle evaluation tool for fault tolerant systems. In *Proceedings of the 12th Annual International Symposium on Fault-Tolerant Computing Systems (FTCS-12)*. (June), pp. 273–274.

MAXEMCHUK, N. F., AND SABNANI, K. 1987. Probabilistic verification of communication protocols. In *Protocol Specification, Testing and Verification*, vol. VII, H. Rubin and C. H. West, eds. Elsevier, Amsterdam, The Netherlands, pp. 307–320.

MUNTZ, R. R., DE SOUZA E SILVA, E., AND GOYAL, A. 1989. Bounding availability of repairable computer systems. In *Proceedings of 1989 ACM SIGMETRICS and PERFORMANCE '89* (Berkeley, Calif., May 23–26. ACM New York, pp. 29–38. (also in a special issue of IEEE-TC on performance evaluation, Dec. 1989, pp. 19–30).

NEUTS, M. F. 1981. *Matrix-geometric Solutions in Stochastic Models—An Algorithmic Approach*. Johns Hopkins University Press, Baltimore, Md.

ROSS, S. M. 1983. *Stochastic Processes*. Wiley series in probability and mathematical statistics. Wiley, New York.

STEWART, W. J., AND GOYAL, A. 1985. Matrix methods in large dependability models, IBM Res. Rep. 11485 (Nov. 4, 1985). T. J. Watson Research Center, IBM, Yorktown Heights, N.Y.

TAKAHASHI, Y. 1972. Some problems for applications of Markov chains. Ph.D. dissertation, Tokyo Institute of Technology, Tokyo, Japan, March.

TAKAHASHI, Y. 1975. A lumping method for numerical calculations of stationary distributions of markov chains. Research Reports on Information Sciences, No. B-18. Tokyo Institute of Technology, Tokyo, Japan, June.

TRIVEDI, K. S. 1982. *Probabılity and Statistics wıth Reliabilıty, Queuing and Computer Science Applications*. Prentice-Hall, Englewood Cliffs, N. J.

TRIVEDI, K. S., DUGAN, J. B., GEIST, R. R., AND SMOTHERMAN, M. K. 1984. Hybrid reliability modeling of fault-tolerant computer systems. *Comput. Elec. Eng., 11*, 87–108.

VAN DIJK, N. M. 1988. Simple bounds for queueing systems with breakdowns. *Perf. Eval. 8*, 117–128.

VAN DIJK, N. M. 1989. The importance of bias-terms for error bounds and comparison results. Tech. Rep. 1989-36. Free University, Amsterdam, The Netherlands.

VARGA, R. 1962. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, N.J.

RECEIVED APRIL 1990; REVISED NOVEMBER 1992; ACCEPTED MAY 1993