An Eclectic Survey of Bounding Methods for Markov Chain Models

Invited paper, presented to honor Edward G. Coffman, Jr. on his 60th Birthday

R. R. Muntz

Computer Science Department
UCLA
Los Angeles, CA 90024

John C.S. Lui

Computer Science Department
Chinese University of Hong Kong
New Territory, Hong Kong

Abstract

Markov models are often used for performance modeling. However most models do not have closed form solutions, and numerical solutions are often not feasible due to the large (or even infinite) state space of models of practical interest. One can sometimes take advantage of special structures such as nearly completely decomposable models or models with matrix analytic solutions. This paper presents a brief survey of some work over the past few years on finding bounds on performance measures for models that are otherwise intractable. The bounds are found by modifying the original model such that: (a) the modified model is efficiently solvable and (b) the modified model is proven to provide an upper (lower) bound on the value of the performance measure in the original model. We have successfully applied this methodology in a number of studies which are briefly surveyed here.

This is not meant to be a broad, inclusive survey but rather we concentrate on an approach that the authors' have been exploring.

1 Introduction

The complexity of the new generation of highly concurrent systems that are now being developed has made the use of sophisticated modeling tools for performance and dependability analysis a high priority enterprise [10]. This is particularly true in life critical situations, e.g., air traffic control, or in situations where the economic stakes are high. Models that can capture the essential features of a design and solution techniques of sufficient power to analyze realistic models are required to permit evaluation early in the system design phase.

Continuous time Markov models are still the most general class of analytic models widely used for performance and dependability analysis. This class of models is robust in terms of its ability to represent a very broad class of models of interest. Unfortunately the characteristics of many models (complex interactions between components, sophisticated scheduling policies, synchronization among parallel tasks, etc.) preclude the possibility of closed form solutions, in general. Numerical solution methods are the most general solution methods [15]. The most pervasive practical limitation to the use of numerical solution techniques is the size of the state space of realistic models. State space cardinalities of real models can outstrip current memory and processor capabilities by orders of magnitude; i.e., while tens of thousands of states can be effectively handled by today's systems, real models often have more than tens of millions of states. Even the realization of the most optimistic predictions for advances in hardware will not eliminate the state space explosion problem, since there is an inexhaustible supply of larger and larger problems. One natural way to deal with this issue is via state space reduction techniques, i.e., a transformation of the model to one with fewer states. Another is to modify the model to one with a a structure which enables efficient solution. The transformed model in general is an approximation of the original model. Several classes of such transformations are described below. A major difficulty (as with many approximation methods), is providing a bound on the error that is introduced. Without being able to bound the error introduced, an approximation is much less satisfying.

2 State Space Reduction Techniques

There are several methods of state space reduction that are applicable to Markov chain models. One such method is coalescing "lumpable states" [6]. When lumping applies, it is exact, but only partial information is retrievable from the solution except in very special circumstances, i.e., usually only the sum of the state probabilities for states in $\mathcal{S}_i, 1 \leq i \leq K$, can be obtained from the reduced model and not the individual state probabilities. Also, lumping is only valid for models that have certain very limited types of structure.

Another method for dealing with a large state space is aggregation/disaggregation [1]. This applies particularly well to cases in which the system model can be viewed as an interacting set of tightly coupled subsystems. The solution method is generally an iterative one in which submodels are solved and the results used to adjust the submodels repetitively until a convergence criteria is met. This is an effective procedure if: (a) the model is decomposable into tightly coupled submodels, (b) the state space of each submodel is within the range that is practically solvable, and (c) the number of such submodels is not too large. The method is not directly applicable if the total state space is too large or if the model does not decompose in the required manner.

There is an additional property of many models that is not capitalized on by any of the above methods. In many important cases, although the state space is extremely large, the stationary state probability distribution is highly skewed, i.e., only a small subset of the states account for the vast majority of the probability mass. This is easily illustrated by considering the nature of several modeling application areas. Consider, for example, a model consisting of a set of servers and a load balancing algorithm, such as "join-the-shortest-queue" scheduling. While the system state space can be very large (even infinite), the very nature of the scheduling algorithm attempts to keep the queue lengths balanced so that states with very imbalanced queue lengths have a very small probability. As another example, in a dependability model, each state of the Markov chain corresponds to a system state in which some components have failed (and so, need some form of repair) and other components are operational. Real systems are designed to have a "high level" of dependability, and so it is expected that most of the time the majority of the system components are operational. This observation indicates that most of the probability mass is concentrated in a relatively small number of states in comparison to the total number of states in the model. During its lifetime, the system spends most of its time in this relatively small subset of states and only rarely reaches other states.

One might use the above observation to motivate truncation of the model state space, i.e., choose to only represent a subset of the more "popular" states and ignore the remainder, deleting transitions from the represented states to the "discarded states". Some form of heuristic is often used to decide which states to retain and which to discard. For dependability models a simple heuristic might be to discard all states in which the system has more than a certain number of failed components. A more sophisticated heuristic for determining which states to retain is described in [5]. There is also the issue of how much error is introduced by state space truncation. For some transient measures, error bounds are easily obtained by introducing trap states [13]. For example, in dependability analysis we may be interested in interval reliability. For this purpose one can introduce a trap state and change all transitions to "discarded states" into transitions to the trap state. Then the magnitude of the error is a direct function of the probability that the system is in the trap state at the end of the interval. For other transient measures, and particularly for steady state measures, the issue of error bounds can be considerably more difficult. (Clearly the trap state idea will not work for steady state measures, since with probability one the system enters the trap state eventually.)

A goal of our research has been to explore methods of solution for large complex Markov chain models that arise in various areas of computer system performance analysis. The theme of the research is centered around the following basic ideas.

- While the state space is often large or even infinite, the probability mass is often concentrated in relatively few states. This was discussed above with respect to load balancing and dependability models. Similar arguments can be made for models from many other application areas (e.g., communication protocol models).

- To capitalize on the skewed distribution of the state probabilities, it is possible in many cases to represent the portion of the model corresponding to the most "popular" states in detail and to introduce a modified representation of the remainder of the state space. The modification should enhance the solvability of the model by either a state space reduction (e.g., through aggregation or even truncation) or by approximating this portion of the state space with a special structure that enables efficient solution.

- It is highly desirable that the approximations that are introduced are provably upper or lower bounds and further, the goal is to provide (tight) upper *and* lower bounds in all cases.

3 Techniques for Proving Bounds

There are a number of approaches available for computing bounds or for proving that one model provides a bound (with respect to some specified performance measure) compared with some other model. Among these are:

- the method of Courtois and Semal [2, 3].
- the methods based on sample path analysis and stochastic ordering [12, 16].
- the bias terms approach of van Dijk [4].

Space limitations prohibit a discussion of these methods. We have found all three methods useful, although we have most often used the first two.

4 Example Applications

4.1 Steady State Availability for Repairable Systems

A method is presented in [11] by which steady state availability can be approximated, and moreover the error introduced by the approximation can be bounded. In addition, this method provides the means for avoiding generation of the entire state transition matrix; often only a small fraction of the transition rate matrix is required. This method also provides the flexibility to tradeoff computational resources and error; if one can generate more of the state transition matrix, then better error bounds can be obtained.

We start by discussing the intuitive motivation behind the method. We mentioned in the previous section that dependability models generally have the property that only a small subset of the states accounts for the vast majority of the state probability mass. With the above observation in mind, the approach, in brief, is to maintain a detailed description of the system model for those states with few failed components (the states with the largest probability mass) and represent the complement of this subset (the remainder of the system description) in a simplified way. This simplified representation of the complement is achieved by aggregation (i.e., coalescing) of states. As was demonstrated in [11], the aggregates and transitions between aggregates can be chosen such that upper and lower bounds are obtained for steady state availability. A typical example is given in [11] in which the complete model has more than 10^9 states, but bounds that have a spread of only 10^{-6} can be obtained by solving models with only 1800 states [11].

The basic method described in [11] has a drawback that it is a "one shot" bound, i.e., a portion of the transition rate matrix is generated, the bounds computed, and if they are not tight enough for the purposes at hand the only remedy is to start over and generate more of the model. Not only is this wasteful, but there is generally a limitation on the size of the submodel that can be solved in any step. This limitation and several other extensions are reported in [8]. The major extension is a block iterative method that enables the bounds to be iteratively tightened to an arbitrary relative accuracy and where the computation can be organized to limit the block size, so that memory limitations can be respected.

4.2 Shortest Delay Routing

In [7] bounds are obtained on mean response time for shortest delay routing models with heterogeneous, exponential servers with unbounded queues. The state space is clearly infinite and these models have, in their full generality, defied exact solution. The skew in the state space probability distribution in these models is toward the more "balanced states", where the expected unfinished work in the queues is more or less equal. An upper bound model is formed by truncating the state space by not allowing departures that would cause the queues to become unbalanced beyond some specified amount. This still leaves the state space infinite. Further restrictions on departures are made which partition the state space into subsets of states $\mathcal{S}_1, \mathcal{S}_2, \ldots$, such that there is only one transition from $\mathcal{S}_i$ to $\mathcal{S}_{i+1}$ and only one transition from $\mathcal{S}_{i+1}$ to $\mathcal{S}_i$. This makes it easy to solve for the conditional state probabilities. From these, the partitions can be aggregated to form a birth-death process which is easily solved. Also, the $\mathcal{S}_i$'s for $i > 0$ all have identical structure so that only two models have to be solved to determine the conditional state probabilities.

Two lower bound models are discussed in the paper, one for homogeneous servers and one for heterogeneous servers. (The one for homogeneous servers is tighter.) Space limitations preclude much discussion of these. In short, they are based on modifying the model such that a departure from one queue which creates an imbalance greater than a certain threshold, triggers a departure from another queue instead. Efficient solutions are obtained with upper and lower error bounds within a few percent of each other. See [7] for details.

4.3 Fork-Join Queueing Model

In [9] we have explored the application of the above methodology to fork-join queues with exponential service times and unbounded queues. In these models, each job splits into statistically identical tasks, one for each server. A job completes when all of its tasks have been completed. The usual performance measure is the mean job response time. As with the shortest mean delay scheduling model, the more popular states are those with nearly equal queue lengths. The modified models are based on leaving the states and transitions rates alone for these states and modifying the model for states beyond some threshold of imbalance. For the upper bound model, departures are delayed that would cause the threshold to be violated. For the lower bound model, departures that would violate the threshold trigger other departures that will maintain the state within the threshold. These modifications are also carefully constructed such that the resulting aggregated process has a *tree-typed structure* so that efficient solution of the models can be obtained. As with the other studies, numerical studies showed that tight bounds were possible with reasonable computation costs. The reader is referred to [9] for further details.

5 A Suggestion for Further Research.

A complex model is usually defined via some specification language or notation (e.g., stochastic Petri nets). The underlying Markov chain has to be generated, and in the case of highly skewed stationary state distributions, we would like to be able to dynamically generate the "most popular" states first. Recent work reported in [14] explores alternative methods with two (not independent) goals: (a) consuming less resources (both CPU time and memory) and (b) more accurately selecting the states with the higher probabilities. This is still a largely unexplored area, and there are a number of possible directions for research, but our main interest here is in bounding methods. There is often an interplay between the bounding methods and dynamic state generation as, for example, in our work on bounds for steady state availability. For the sake of clarity in the earlier discussion of the availability bounds we discussed only the basic results, and left out a number of details and generalizations. One assumption was that the states of the model were generated by order of the number of failed components. In some cases this may not be the best strategy for minimizing the spread in the bounds (for the same number of states generated). For example, if the failure of some components can cause other components to fail, it may be that some states with $n+k$ failed components will have higher probabilities than some other states with n failed components. It is therefore of interest to be able to have the flexibility to select the portion of the state space that corresponds to the more "popular" states no matter which they might be. In [8] we have shown that the bounding methodology for steady state availability can be generalized to accommodate any ordering of state generation. For other problems the situation is different, but there still appears to be an opportunity to exploit the dynamic state generation idea. In the shortest-queue routing model, for example, recall that we chose a modification of the model to obtain the upper bounds. That modification involved selection of some portion of the state space which would be accurately represented in the Markov chain (other transitions and states were eliminated). The choice of which states to represent accurately and which to discard is currently based on a heuristic notion of which are the states with highest probability (those states for which the difference between the largest and the smallest queue length is below some threshold). We plan to investigate the possibility of dynamic generation of the states, *but* incorporate into the dynamic generation a set of constraints which have to be satisfied to guarantee that the upper bound can still be proved. So, the requirement that an upper bound be obtained and the desire to generate the set of states with the highest steady state probabilities have to be balanced against each other. (A similar approach should be applicable to the lower bound as well.)

In general, the methods for computing bounds involve the representation of some (high probability) states in detail and then modification of the remainder of the model to accommodate efficient solution. Both have to be done in such a way that a (provable) bound is obtained on the performance metric of interest. The question of interest is, "For which cases can we use dynamic generation of the portion of the state transition matrix that will be represented in detail and still satisfy the conditions that allow bounds to be obtained?". Attacking this problem requires obtaining a formal characterization (the loosest possible) of what conditions the subset of "detailed states" must satisfy such that the bounds are still provable.

6 Conclusion

Many models have a large (or infinite) state space but where most of the states have very low probability. This is exploited in the examples given in this paper where the model modifications only affect the less popular states and therefore have less effect on the performance measures. This approach has proven successful in a number of applications but still requires some ingenuity to apply in a new situation. A major improvement would be to automate some of the work in finding (and verifying) upper and lower bounds.

References

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

[2] P.J. Courtois and P. Semal. *Bounds for the Positive Eigenvectors of Nonnegative Matrices and for Their Approximations.* JACM, Vol. 31, No. 4, Oct. 1984, pp. 804- 825.

[3] P. J. Courtois, P. Semal. *Computable Bounds for Conditional Steady-State Probabilities in Large Markov Chains and Queueing Models.* IEEE JSAC, Vol 4, number 6, September, 1986.

[4] Nico M. van Dijk. *The Importance of Bias-terms for Error Bounds and Comparison Results.* Proceedings of the 1st International Conference on Numerical Solutions of the Markov Chains, North Carolina State University, January, 1989.

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

[6] J. G. Kemeny and J. L. Snell. *Finite Markov Chains.* Van Nostrand Reinhold, New Jersey, 1960.

[7] John C.S. Lui, R.R. Muntz and D. Towsley. *Bounding the Response Time of a Minimum Expected Delay Routing System: An Algorithmic Approach.* Accepted for publication in IEEE Trans. on Computers.

[8] John C.S. Lui, R.R. Muntz. *Computing Bounds on Steady State Availability of Repairable Computer Systems.* Journal of ACM, Vol 41, No. 4, pp. 676-707, July 1994.

[9] John C.S. Lui, R.R. Muntz and D. Towsley. *Computing Performance Bounds for Fork-join Queueing Models* Technical Report CSD-940034, Computer Science Dept. UCLA, Oct. 1994.

[10] A.M.Johnson and M. Malek. *Survey of Software Tools for Evaluating Reliability, Availability and Serviceability* ACM Computing Surveys, Dec. 1988, pp. 227-271.

[11] R.R. Muntz, E. De Souza E Silva and A. Goyal. *Bounding Availability of Repairable Computer Systems.* In Proceedings of 1989 ACM SIGMETRICS and PERFORMANCE '89, also in special issue of IEEE Trans. Computers, Dec. 1989, pp. 19-30.

[12] S.M. Ross. *Stochastic Processes.* John Wiley and Sons, 1983.

[13] E. de Souza e Silva and H. R. Gail. *Calculating Availability and Performability Measures of Repairable Computer Systems Using Randomization.* JACM, Vol. 36, No 1, Jan. 1989, pp. 419-430.

[14] E. de Souza e Silva and P. Mejia. *State Space Exploration in Markov Models.* SIGMETRICS/Performance '92, pp. 152-166.

[15] W.J. Stewart and A. Goyal. *Matrix Methods in Large Dependability Models.* Technical Report RC 11485, IBM T. J. Watson Research Center, 1985.

[16] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models.* John Wiley and Sons, 1983.