Fast Decomposition in Large Stochastic Models

Alexandre Brandwajn
School of Engineering, Computer Engineering
University of California at Santa Cruz

Abstract

We propose a novel approach to the decomposition of large probabilistic models. The goal of our method is to avoid the evaluation of the subnetworks obtained by decomposition for all values of the state description vector, as would be necessary with a standard aggregation and decomposition approach. Instead, we propose a fixed-point iteration that requires the evaluation of the subnetwork for only a subset of the population levels. Outside the evaluated points, simple upper and lower linear approximations are used resulting in bounds for overall system performance measures. We concentrate the evaluation of the subnetworks in the regions where the difference between the lower and upper bound is most likely to impact the accuracy of the result.

1. Introduction

Decomposition methods have been used in queueing models to overcome problems introduced by state dependencies that preclude analytical solution, and also as a way of approaching systems with a large state space. In particular, a family of state aggregation (equivalence) methods divides the model into subsystems whose interaction is represented via conditional throughputs. The latter are obtained by analyzing the subsystems in isolation [5, 4, 3]. An interesting feature of this approach is that it generates the exact solution when applied to a product-form queueing network [10].

Figure 1 shows an example of the application of this type of decomposition [2]. The system under consideration is a closed queueing network in which a total of N customers circulate alternating between a delay server and a subnetwork. The subnetwork may exhibit global dependencies in that some of its parameters (such as service completion rates) may depend on the current total number of customers in the subnetwork.

In the decomposition approach, the subnetwork is analyzed in isolation to produce the conditional throughput u(n) for n=1,…,N. u(n) represents the rate at which customers leave the subnetwork given that the current number of customers in the subnetwork is n. The values of u(n) are then used in the simpler equivalent (aggregated) model. Some authors refer to the subnetwork in isolation as the "inner model", and to the state equivalent as the "outer model".

As outlined, this approach requires that the inner model be solved for all values of the state vector retained in the outer model, i.e. all population levels in our case. Clearly, for a large population, this can represent a significant computation effort. Note that there exists an alternate iterative method [8, 1] that does not require the evaluation of the inner model for all population levels. However, this method is not readily applicable to queueing systems with global dependencies.

For large systems with many customers it seems reasonable to think that only a relatively small subset of all possible states would correspond to the "operating point" of the systems. In other words, the state probabilities in many instances tend to be skewed.

As an example, consider the network of exponential servers shown in Figure 2. Without global dependencies, this network possesses a product form solution. Let p(n) be the probability that there are a total of n customers outside the delay servers. Figure 3 shows examples of the shape of p(n) as a function of n, and also the corresponding conditional throughputs as a function of m, the number of users admitted into the subnetwork of queues. These results were obtained using classical decomposition, i.e. the subnetwork in isolation was evaluated for all population levels m=1,…,M where $M \leq N$.

The probability mass tends to concentrate within a relatively small subset of possible values for n. No doubt, this subset corresponds to the most probable "operating region(s)" of the system. This observation suggests that the precise evaluation of the conditional throughputs u(.) should matter close to the operating point(s) of the system, and just a reasonable approximation for u(.) should suffice elsewhere. By "reasonable" we mean one that would not alter excessively the shape of the probability distribution p(n), and, in particular, the location of the operating regions.

Clearly, when dealing with a queueing system we generally don't know where its most likely points of operation are. It is possible however, to design a simple fixed-point iteration that quickly finds the most important points. Note that if, outside of the points for which the inner model is actually solved, we use an upper and lower bound for the conditional throughput u(.), we will obtain in turn an optimistic and a pessimistic bound for overall performance measures such as system response time or expected number of customers in the system.

In the next section we consider the bounds for the conditional throughputs u(.) in the common case when u(.) possesses a single type convexity. Section 3 is devoted to the fixed-point iteration, and sample results. In Section 4, finally, we present conclusions of this paper, and we consider the application of our method to systems where the conditional throughputs exhibit inflexion points.

2. Bounding of the conditional throughput function

We now restrict our attention to systems for which the conditional throughput u(.) does not exhibit inflexion points,

akin to the function considered in Figure 3. Note that the conditional throughput for the sample network of Figure 2 conforms to this shape, even in the case of admission control, if we consider it as a function of m, the number of users in the system proper. Our goal is to develop reasonable simple lower and upper bounds knowing some number of points on the curve u(m) (m=1,…,M).

We assume in our discussion that the values of u(m) have been evaluated for the bounds of the domain of interest (m=1 and m=M), as well as for a few intermediate points. Let k be the number of intervals defined by the known points. Consider points x_i and x_{i+1} defining interval i, and denote by c_i the chord through these points. It is clear that c_i defines a lower bound for u(x), for x contained in the interval $[x_i, x_{i+1}]$. It is also clear from the convexity of u(.) that the two chords c_{i-1} and c_{i+1} each define an upper bound for u(.) in the same interval. To obtain the tightest bound, we select the lower of the two values. For the first interval, we use c_2 as the upper bound, and for the last interval we use the smaller of c_{k-1} (chord through the next to last interval) and the value u(M). This is summarized below:

for $x \in (x_i, x_{i+1})$: $f_{inf}(x) = c_i(x)$,
$f_{sup}(x) = \min\{ c_{i-1}(x), c_{i+1}(x)\}$,

for $x \in (1, x_2)$: $f_{inf}(x) = c_1(x)$, $f_{sup}(x) = c_2(x)$,

for $x \in (x_{k-1}, M)$: $f_{inf}(x) = c_k(x)$,
$f_{sup}(x) = \min\{ c_{k-1}(x), M\}$.

For the type of queueing systems under consideration, it is clear that solving the outer model using the lower bound on the conditional throughput will produce pessimistic bounds for overall system performance. Conversely, using the upper bound for u(.), will produce optimistic values for system performance measures. The next section presents a fixed-point approach to the selection of evaluation points for u(.).

3. Fixed-point iteration

To keep the computational effort low, it seems natural to attempt to select the evaluation points so that they contribute most to the accuracy of the result. Since the most likely operating regions are not known initially, we propose the following procedure. We start by computing the values of u(.) for m=1, m=M, and some small number of intermediate points. We use four evenly spaced intermediate points. This allows us to create the upper and lower bounds for all missing values of u(.), and results in two probability distributions for the number of customers in the system in the outer model.

Let q(n) and r(n) denote the distribution obtained using the upper and the lower bound for u(n), respectively. We select the additional evaluation point for u(.) as being the point for which the difference between the upper and lower bounds weighted by the probability of occurrence of the given state is the largest in absolute value, i.e., which maximizes $|u_{upper}(n)\, q(n) - u_{lower}(n)\, r(n)|$. Let j be the corresponding population level. To reduce the number of iterations needed, we also compute the conditional throughputs for $j-\Delta$, and $j+\Delta$, with Δ set to some relatively small number (3 in our case). If the selected j value corresponds to a point already evaluated, we may select a somewhat lower or a somewhat higher value with the allowed offset arbitrarily limited to 5 in our case.

We repeat this procedure until either the consecutive values of a selected performance measure have stabilized to within a predefined accuracy or no new points are selected for the evaluation of u(.). In the many trials, the procedure tended to converge within just a few iterations. The total number of points evaluated for u(.) depends on the particular set of system parameters, and represents typically a small fraction of the total number of population levels. Somewhat surprisingly, the bounding turns out to be generally quite tight, the difference between performance measures obtained using the upper and lower bounds rarely exceeding a couple of percent.

Table 1 shows examples of our results. The expected total number of customers in the system was used as the system performance measure of interest. The third row in this table shows one of the less accurate cases.

4. Conclusions

The proposed approach uses sampling of the conditional throughputs that represent a subnetwork in the outer model. This sampling is guided towards the estimated largest "payoff" in terms of minimizing the difference between the upper and lower bounds for the conditional throughputs. The use of bounds for the conditional throughputs allows us to obtain bounds on overall system performance measures as opposed to "one-sided" approximations.

Experimental evidence seems to suggest that this approach tends to produce accurate results at a fraction of the cost of a full-blown decomposition. It has the added advantage that accuracy can be arbitrarily refined, should the need occur, by simply considering additional points for the conditional throughput function.

The bounding of the conditional throughputs presented in Section 2 relies on the convexity of the throughput function u(.). There are queueing systems for which the throughput exhibits an inflexion point. This is the case, for example, for systems with thrashing such as virtual memory systems or certain computer networks. Clearly, to guarantee bounding of the conditional throughput function, the intervals containing an inflexion point must be detected and dealt with appropriately. Preliminary investigation indicates that this appears possible without undue computational effort, so that our method should apply to those systems as well.

In conclusion, we have presented a bounding method designed to reduce the computational cost inherent in the application of decomposition methods to large queueing systems. The resulting method is a quickly converging fixed-point iteration, and the bounds produced are generally tight enough to be used as a quality approximation. The approach we propose is not limited to queueing models. The type of decomposition methods considered in this paper finds its application also in econometrics, so that our method can be applied in this context, as well as in areas such as Petri nets or reliability models.

References

[1] B. Baynat, Y. Dallery, “Approximate techniques for general closed queueing networks with subnetworks having population constraints”, *European J. of Operational Res. 69*, 250-264 (1993).

[2] A. Brandwajn, “A model of a time sharing virtual memory system solved using equivalence and decomposition methods”, *Acta Informatica 4*, 11-47 (1974).

[3] A. Brandwajn, “Equivalence and Decomposition in Queueing Systems – A Unified Approach”, *Performance Evaluation 5*, 175-185 (1985).

[4] K. M. Chandy, U. Herzog, L. Woo, “Parametric analysis of queueing networks”, *IBM J. Res. Develop. 19*, 36-42 (1975).

[5] P. J. Courtois, *Decomposability: Queueing and Computer Applications*, Academic Press, New York, 1977.

[6] E. de Souza e Silva, P. Mejia Ochoa, “State Space Exploration in Markov Models”, *Performance Evaluation Review 21*, 152-166 (1992).

[7] D. D. Dimitrijevic, M.S. Chen, “Dynamic state exploration in quantitative protocol analysis”, in *Protocol Specification, Testing and Verification IX*, 327-338, North Holland, 1990.

[8] R. A. Marie, “An approximate analytical method for general queueing networks”, *IEEE Trans. on Software Engin. SE-5*, 530-538 (1979).

[9] R. R. Muntz, E. de Souza e Silva, A. Goyal, “Bounding availability of repairable computer systems, *IEEE Trans. on Computers 38*, 1714-1723 (1989).

[10] H. Vantilborgh, “Exact aggregation in exponential queueing networks”, *J. ACM 25*, 620-629 (1978).

[11] C. L. Yang, P. Kubat, “Efficient computation of most probable states for communication networks with multimode components”, *IEEE Trans. on Communications 37*, 535-538 (1989).

Acknowledgement

The author thanks Fabrice Clérot of France Télécom CNET Lannion (France) for fruitful discussions of linear bounds for the conditional throughput function.

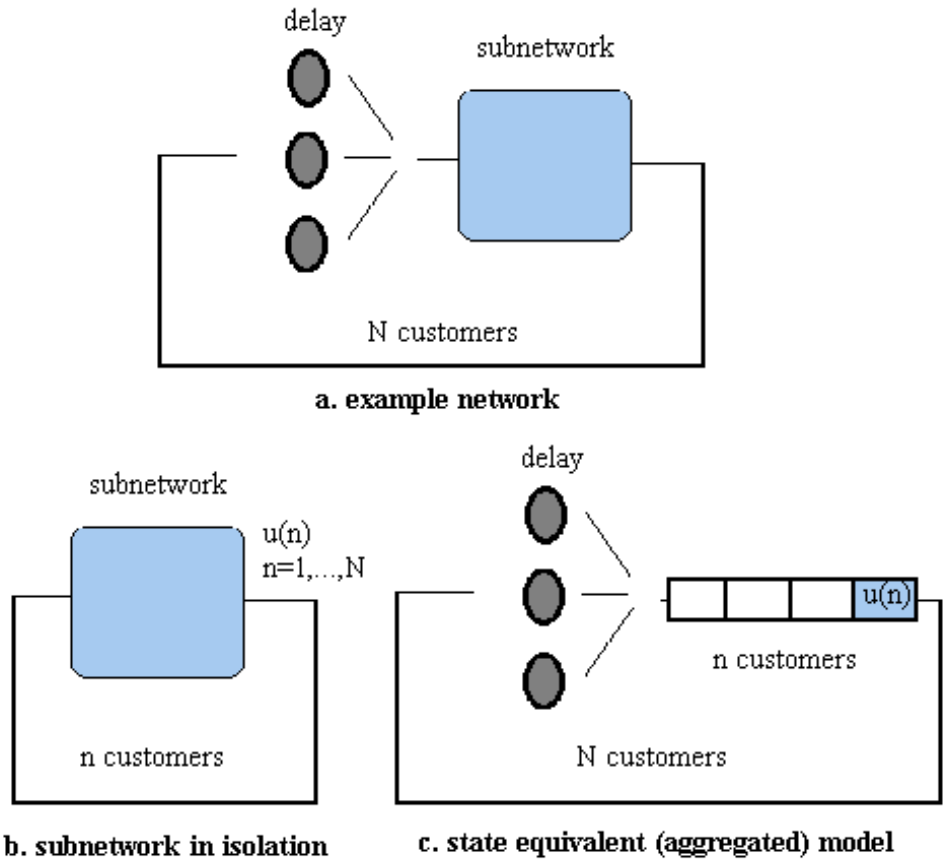

Figure 1: decomposition in a queueing network

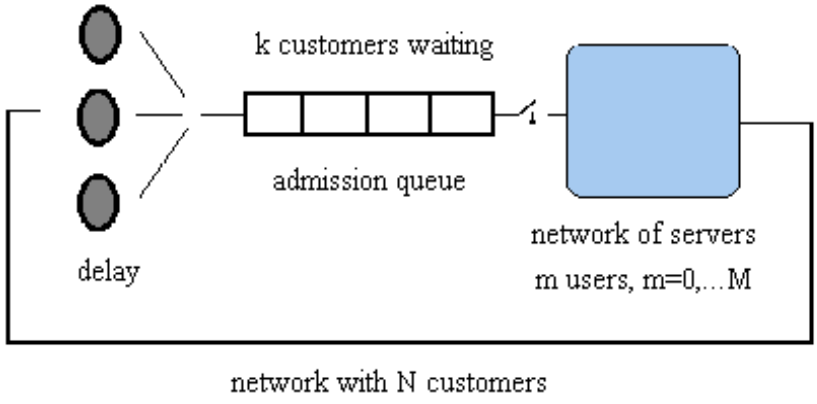

Figure 2: example of queueing network

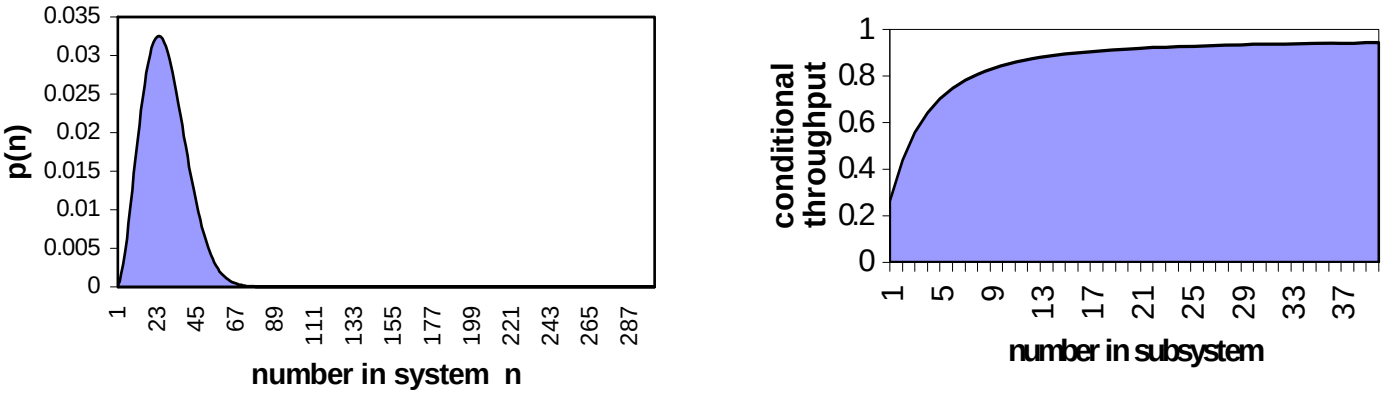

Figure 3: example of state probability and conditional throughput

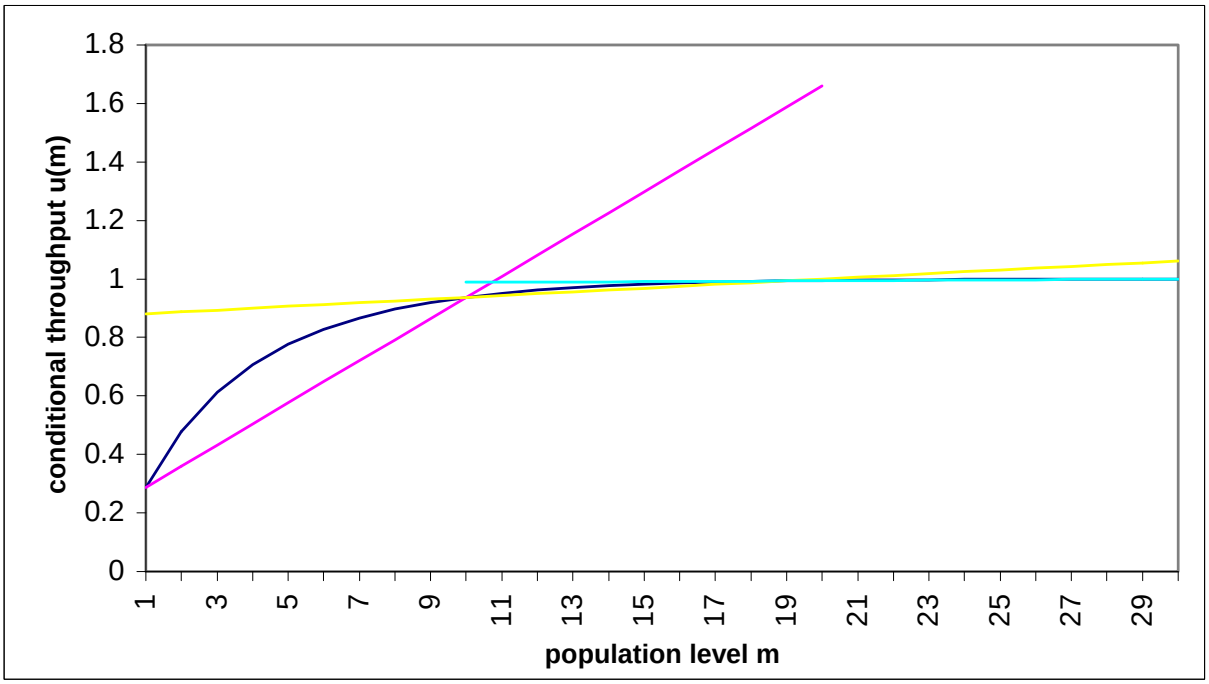

Figure 4: bounding lines for conditional throughput

Table 1: examples of results

total number of customers (N)	maximum number admitted (M)	mean number in system	number of points evaluated	number of iterations
300	300	16.968 17.765	33	11
300	300	85.714 86.056	9	3
300	100	27.658 30.740	15	5
300	100	69.231 69.244	9	3