

# Horner's Rule for the Evaluation of General Closed Queueing Networks

M. Reiser and H. Kobayashi  
 IBM Thomas J. Watson Research Center

The solution of separable closed queueing networks requires the evaluation of homogeneous multinomial expressions. The number of terms in those expressions grows combinatorially with the size of the network such that a direct summation may become impractical. An algorithm is given which does not show a combinatorial operation count. The algorithm is based on a generalization of Horner's rule for polynomials. It is also shown how mean queue size and throughput can be obtained at negligible extra cost once the normalization constant is evaluated.

**Key Words and Phrases:** Queueing networks, queueing theory, Horner's rule, evaluation of multinomial sums, load-dependent service rate

**CR Categories:** 5.12, 5.5, 8.1, 8.3

Queueing networks provide important models for complex computer systems. The most general class of analytically solvable queueing networks is characterized by (1) servers with memoryless service time distributions and (2) Markovian routing. The solution is known as *product form solution*. Recent progress in extending the scope of the product form solution is found in [1, 2]. Well-known special cases of the class of networks treated in [1] are the exponential server networks as described in [3, 4]. Although the product form solution is quite simple mathematically, a numerical evaluation requires a summation of the product terms over the entire state space, which exhibits a combinatorically exploding size. The great interest in applying large queueing network models has led to several solutions of the computational problem [5-7]. It is the object of this paper to describe an algorithm which is based on a

Copyright © 1975, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery.

Authors' address: IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

*multidimensional Horner scheme*. Our algorithm allows evaluation of the most general case with load-dependent servers. It is faster than previously published algorithms of the same generality (although it has the same asymptotic growth of the operations count).

We consider a closed queueing network with  $M$  servers and  $N$  customers which has a product form solution. For such a network, the quantities of interest (i.e. normalization constant and marginal distributions) are given by homogeneous multinomial expressions of the form

$$G(M, N) = \sum_{\mathbf{n} \in D(M, N)} \prod_{m=1}^M \prod_{n=1}^{n_m} \tau_{mn} \quad (1)$$

with

$$\tau_{mn} = e_m / \mu_m(n), \quad (2); \quad \mathbf{e} = \mathbf{e}P \quad (3)$$

where  $\mathbf{n}$  is the state vector of queue lengths  $\mathbf{n} = (n_1, n_2, \dots, n_M)$ ,  $D(M, N)$  is the feasible state space defined by  $D(M, N) = \{\mathbf{n}; \mathbf{n} \geq 0 \text{ and } \sum_i n_i = N\}$ ,  $\mu_m(n)$  is the rate of server  $m$  as a function of its local queue size  $n$ , and  $P$  is the routing matrix. The quantities  $\mathbf{e} = (e_1, e_2, \dots, e_M)$  defined by (3) are proportional to the throughput of each of the servers. Note that  $\mathbf{e}$  is not uniquely determined by (3). The solution (1), however, is unique after normalization. For more details we refer to the original literature [1-4]. The basic idea for evaluating the sum (1) is to partition the state space into mutually exclusive subsets as follows:

$$D(M, N) = \bigcup_{i=0}^N D(M-1, N-i), \quad (4)$$

$D(M-1, N-i) = \{\mathbf{n}; \mathbf{n} \geq 0 \wedge n_M = i \wedge \sum_i n_i = N-i\}$ . Then a factor  $\prod_{n=1}^i \tau_{Mn}$  can be factored out of the sums over the subdomains  $D(M-1, N-i)$  yielding

$$G(M, N) = \sum_{i=0}^N G(M-1, N-i) \prod_{n=1}^i \tau_{Mn}. \quad (5)$$

(Note that empty products have the standard value 1 and therefore the value of  $G(M, 0)$  is 1.) Expressions of the form (5) are most efficiently evaluated by means of Horner's rule [8], e.g. for  $M=3$  and  $N=3$ ,

$$G(3, 3) = G(2, 3) + \tau_{31} [G(2, 2) + \tau_{32} [G(2, 1) + \tau_{33}]]. \quad (6)$$

Equation (5) could in principle be implemented as a recursive subroutine. This, however, would yield a exponentially growing operation count. Inspection of the recursive tree shows that the same subexpressions are reevaluated repetitively and that a row-wise construction of the array  $G(m, n)$ ,  $m = 1, 2, \dots, M$  and  $n = 1, 2, \dots, N$  similar to [6] avoids the exponential growth. We summarize our algorithm as follows.

*Step 1.* Initialize first row by

$$G(1, n) = \prod_{i=1}^n \tau_{1i} \quad \text{for } n = 1, 2, \dots, N. \quad (7)$$

*Step 2.* For each level  $m = 2, 3, \dots, M$  compute row-

rise the values  $G(m, n)$ ,  $n = 1, 2, \dots, N$  by means of Horner's rule:

$$G(m, n) = G(m-1, n) + \tau_{m1} [G(m-1, n-1) + \tau_{m2} [G(m-1, n-2) + \tau_{m3} [\dots + \tau_{m, n-1} [G(m-1, 1) + \tau_{mn}] \dots]] \quad (8)$$

The operation count for this algorithm is  $(1/2)(M-2)(N-1)N + 2(N-1) = O(MN^2)$  essential operations (i.e. multiplications and divisions) as compared to  $2M(N+1)$  for the previously published algorithms [7]. The storage requirement is  $2N$  cells.

In the special case of constant processor speed (i.e.  $\tau_{m1} = \tau_{m2} = \dots = \tau_{mM} = \tau_m$ ). Step 2 of the algorithm simplifies to

Step 2'. For each level  $m = 2, 3, \dots, M$  compute row-wise the values

$$G(m, 1) = \sum_{i=1}^m \tau_{mi} \quad (9)$$

$$G(m, n) = G(m, n-1) + \tau_m G(m-1, n) \quad \text{for } m = 2, 3, \dots, M. \quad (10)$$

The operation count reduces to  $M(N-1) = O(MN)$ . Step 2' yields the first algorithm of [7]. It is now, however, a special case of the general algorithm, whereas in [7] the two algorithms had no connection.

We conclude with two formulas which allow an especially efficient evaluation of queue statistics. For the throughput at server  $m$  we find

$$T_m = e_m G(M, N-1)/G(M, N). \quad (11)$$

The mean queue size of a server with constant rates is given by

$$E\{n_m\} = \tau_m G(M+1, N-1)/G(M, N) \quad (12)$$

where  $G(M+1, N-1)$  is obtained from the array  $G(M, n)$ ,  $n \in [1, N]$  by applying once more step 2' with the parameter  $\tau_m$ . Note that (12) avoids time-consuming computation of the entire marginal distribution.

Received February 1974; revised April 1975

#### References

1. Baskett, F., Chandy, K.M., Muntz, R.R., and Palacios, J.G. Open, closed and mixed networks of queues with different classes of customers. *J. ACM* 22,2 (Apr. 1975), 248-260.
2. Posner, M., and Bernholtz, P. Closed finite queueing networks with time lags and with several classes of units. *Op. Res.* 16 (1968), 977-985.
3. Jackson, J.R. Jobshop-like queueing systems, *Management Sci.* 10 (Oct. 1963), 131-142.
4. Gordon, W.T., and Newell, G.F. Closed queueing systems with exponential servers. *Op. Res.* 15 (Apr. 1967), 254-265.
5. Moore, R.I. Computation model of a closed queueing network with exponential servers. *IBM J. Res. Dev.* 16 (Nov. 1972), 567-572.
6. Reiser, M., and Kobayashi, H. Recursive algorithms for general queueing networks with exponential servers. *IBM Res. Rep. RC 4254*, March 1973.
7. Buzen, T.P. Computational algorithms for closed queueing networks with exponential servers, *Comm. ACM* 16, 9 (Sept. 1973), 527-531.
8. Knuth, D.E. *The Art of Computer Programming, Vol. 2*. Addison-Wesley, Reading, Mass. 1969.

# Available in Microfiche

For libraries supporting instruction in the Computer Sciences, these publications form the nucleus for a basic, technical collection.  
—Magazines for Libraries, 2nd Edition

#### Association for Computing Machinery: Journal

Volumes 1-18. New York. 1954-1972. \$180.00  
per volume \$ 9.50

One of the foremost journals on mathematical modeling and numerical analysis, its research articles on these topics and on programming theory and languages, logical design and numerical mathematics are contributed by university and industry scientists throughout the world.

#### Association for Computing Machinery: Communications

Volumes 1-15. New York. 1958-1972. \$165.00  
per volume \$ 11.00

Spanning the spectrum of the computing sciences, the *Communications* acts as a technical forum for reflecting the current trends in research and development in all areas of information processing. Papers deal with the subjects of computer systems, operating systems, programming techniques, management/data base systems, information retrieval, scientific applications and education.

#### Computing Reviews

Volumes 1-13. New York. 1960-1972. \$123.50  
per volume \$ 9.50

One of the most prominent publications in the computing sciences, *Computing Reviews* produces critical reviews, not merely abstracts, in all branches of the literature. Within the scope of these reviews are books, articles (selected from over 230 journals), conference proceedings, selected theses, or anything which the editorial board considers significant for its readership. The emphasis is on reviewing and not on timely documentation of the literature.

#### Computing Surveys

Volumes 1-4. New York. 1969-1972. \$ 30.00  
per volume \$ 7.50

The general review publication of the Association, *Computing Surveys* carries two to three survey papers in every issue of more than topical interest to a wide clientele such as "Computers in the Humanities and Fine Arts." Tutorial articles elucidate well known theories or applications of an old subject such as "Flow Charting." All are written by well known specialists.



**JOHNSON ASSOCIATES INC.**

P.O. Box 1017 Greenwich, Conn. 06830 • (203) 661-7802