FINITE QUEUES IN SERIES WITH EXPONENTIAL
OR ERLANG SERVICE TIMES

BY
FREDERICK S. HILLIER and RONALD W. BOLING

TECHNICAL REPORT NO. 88
March 18, 1966

SUPPORTED BY THE ARMY, NAVY, AIR FORCE AND NASA UNDER
CONTRACT N000-225(53) (NR-042-002)
WITH THE OFFICE OF NAVAL RESEARCH

DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA
FINITE QUEUES IN SERIES WITH EXPONENTIAL OR ERLANG SERVICE TIMES

by

Frederick S. Hillier and Ronald W. Boling

TECHNICAL REPORT NO. 88

March 18, 1966

Supported by the Army, Navy, Air Force, and NASA under Contract Nonr-225(53) (NR-042-002) with the Office of Naval Research

Gerald J. Lieberman, Project Director

Reproduction in Whole or in Part is Permitted for any Purpose of the United States Government

DEPARTMENT OF STATISTICS

STANFORD UNIVERSITY

STANFORD, CALIFORNIA
FINITE QUEUES IN SERIES WITH EXPONENTIAL

OR ERLANG SERVICE TIMES*

by

Frederick S. Hillier and Ronald W. Boling

1. Introduction

Queueing systems with a number of service facilities in series have received considerable analytical study in recent years. However, only a relatively few of the studies have imposed the restriction that only finite queues are allowed, i.e., a specified finite bound is placed on the allowable size of the queue in front of each of the facilities (except perhaps the first one). A pioneering investigation that did consider this case was performed by Hunt [9], who derived the maximum possible utilization of the system and the corresponding expected number of customers in the system under the assumption of exponential service times. Unfortunately, Hunt was only able to obtain relatively limited results, and little progress has since been reported in extending these results. The purpose of this paper is to present some new extensions of Hunt's work, both in terms of analytical procedures and numerical results.

The motivations for this work were two-fold. First, the authors feel that the problem considered by Hunt is a relatively important one.

*The authors wish to express their appreciation to Reuven Amir, I. F. Burns, Stephen F. Love, and Charles Mylander for their assistance on computational aspects of the problem, and to the computation centers at Stanford University and the University of Tennessee for providing computer time to develop the results given here.
Queueing systems involving finite queues in series are rather common in practice. For example, most production line systems, and quality control systems with a sequence of inspection stations, are of this type. The results sought by Hunt could be most useful when designing such systems, e.g., for determining the amount of storage space and the number of stations to provide. The second motivation was that the authors have been conducting an independent investigation of the effect of unbalancing a production line with variable operation times by assigning unequal expected operation times to the respective stations. Preliminary results reported elsewhere [7] demonstrate that, in some cases, unbalancing such a production line actually can increase its efficiency. In order to conduct this investigation satisfactorily, it became necessary to develop and apply some of the procedures presented here.

The queueing system to be studied here consists of \( N \) service channels in series in a steady-state condition. Thus, every customer must be processed through each of the \( N \) single-server service facilities in the same fixed sequence. The service times at the \( j^{th} \) facility either have an exponential distribution or an Erlang distribution with shape parameter \( k_j \), with a mean of \( \frac{1}{\mu_j} > 0 \) (\( j = 1, 2, \ldots, N \)). All service times are independent. There is always a customer available to be processed at the first facility. This includes the case of a Poisson input process with an infinite queue before the first facility where the mean arrival rate equals the maximum mean output of the system. The maximum allowable queue size before the \( j^{th} \) facility (not counting either a customer being held at the \((j-1)^{st}\) facility or being served
at the $j^{th}$ facility is $S_j$ ($j = 2, 3, \ldots, N$). Thus, if a customer completes service at the $(j-1)^{st}$ facility when the queue before the $j^{th}$ facility is full, that customer must be held at the $(j-1)^{st}$ facility (without service beginning for the next customer) until the queue is no longer full. Because of such blocking, the mean output rate of the system is less than the mean service rates. The objectives of this study are to determine this mean output rate, $R$, and (secondarily) to obtain the mean number of customers in the system (not counting those waiting to be served at the first facility), $L$. When the mean service rates are equal (to $\mu$), the utilization, $\rho_{\text{MAX}} = \frac{R}{\mu}$, will be found instead of $R$ in order to correspond to Hunt's results. The interpretation of $\rho_{\text{MAX}}$ is that, when the mean input rate is a free variable, $\rho_{\text{MAX}}$ is the maximum possible ratio of $R$ (which equals the effective mean arrival rate) to $\mu$.

Saaty [13] has presented a survey of the work done in the general area of queues in series. Most of this work studies queueing systems with a Poisson input, infinite queues in series, and identical exponential service times. A fundamental result for this case that was reported by Burke [3] is that, for each facility, the steady-state output process (and therefore the input process to the next facility) also is Poisson. This result provided the motivation for the approximate procedure

---

Notice that this includes the case where the first queue also is finite. For this case, the first facility would be interpreted as an input source generating arrivals into the queueing system consisting of $(N-1)$ service channels in series. Excluding the pauses when the first queue is full (i.e., contains $(S_2 + 1)$ customers), the interarrival times would therefore have either an exponential or Erlang distribution.
presented here for the corresponding case with finite queues in series. There also has been other relevant work reported recently. Patterson [11] has discussed some analytical methods in the study of finite queues in series. Avi-Itzhak and Yadin [2] studied a queueing system consisting of a sequence of two service facilities with an infinite queue allowable before the first facility and no queue allowable between the facilities. Avi-Itzhak [1] derived some characteristics of a queueing system with finite queues in series and constant service times. Friedman [5] independently conducted a similar investigation. Work on applying analyses involving finite queues in series to the design of production lines has been reviewed by the authors [7] in conjunction with presenting new findings in this area.

This paper presents three types of new results. First, a procedure is described for obtaining $R$ and $L$ for either exponential or Erlang service times. This procedure, which is somewhat different than Hunt's, has been designed especially for completely automatic execution on a digital computer in as efficient a manner as possible. Second, the most unique contribution of the paper is a new procedure for obtaining an approximate value of $R$ when the service times are exponential. This procedure provides an excellent approximation for most cases and, in contrast to the exact procedure, is computationally feasible for large problems. Finally, both the exact procedure and approximate procedure are applied to obtain extensive new numerical results.
2. Procedure for Obtaining Exact Results

The procedure that was developed for obtaining an exact value of \( R \) and \( L \), given exponential or Erlang service times, may be summarized as follows. The queueing process is formulated as a continuous time parameter Markov chain. A systematic method is used to automatically generate the states of the Markov chain and the corresponding transition probability intensity matrix. Since this matrix tends to be both very large and sparse, a special method is used to store it in the computer so that most of the zero elements can be excluded. Solving for the stationary distribution of the Markov chain requires solving a system of linear equations. Rather than using the method of Gaussian elimination, a special algorithm is used to do this in order to exploit the sparseness of the matrix. Given the stationary distribution, it is then trivial to compute \( R \) and \( L \). These steps of the procedure are described in somewhat more detail below.

To indicate how to generate the states and transition matrix of the Markov chain, consider first the case of exponential service times. It is clear that the queueing process under consideration is then a continuous time parameter Markov chain whose states correspond to the distinct feasible values of the vector, \((-e_2, e_2, e_3, \ldots, e_N)\), where \( e_j \) is the sum of the number of customers in the queue being served at the \( j^{th} \) facility (either 0 or 1), the number in the queue in front of this facility, and the number being held at the \((j-1)^{st}\) facility (either 0 or 1). Thus, the feasible values of \( e_j \) are the integers between 0 and \((S_j + 2)\), although \((S_j + 1)\) and \((S_j + 2)\) are feasible only for certain combinations of the values of the other elements. To generate the states and transition
matrix \( Q = (q_{jk}) \), maintain both a list of states already identified but not analyzed (beginning initially with the state, \((-1, 0, 0, \ldots, 0)\)) and a list of states that have been both identified and analyzed. Begin each iteration by selecting a state to be analyzed from the former list. For this state, identify each of the facilities that are in the process of serving a customer. For each such facility (call it the \( j^{th} \) facility), identify the new state if a service completion were to occur (i.e., add one to \( e_{j+1} \) and subtract one from \( e_j \)). If this state has not been identified previously, add it to the list of those to be analyzed. The corresponding entry in the transition matrix for the transition probability intensity from the old state to the new state is \( \mu_j \). Continue these iterations until no states remain to be analyzed. The transition matrix now is complete except for the diagonal elements. To facilitate subsequent calculations, use \( q_{ii} = -\sum_{j \neq i} q_{ij} \) for all \( i \in \mathbb{S} \), where \( \mathbb{S} \) is the state space for the Markov chain.

Essentially the same method is used for the case of Erlang service times. The only differences arise out of the fact that the states now are also distinguished on the basis of the individual exponential service phases.

Given the states and transition matrix \( Q \), the problem of finding the stationary distribution of the Markov chain may be formulated as follows. Let \( n \) be the number of states, let \( p_j \) be the stationary probability of being in the \( j^{th} \) state, and let the vector \( P = (p_1, p_2, \ldots, p_n) \). Applying the Chapman-Kolmogorov equation yields the system of linear equations,
\[ P \cdot Q = 0, \]

where \( 0 \) is the null vector. Since any one of the equations is redundant, one of them should be replaced by the equation,

\[ \sum_{i=1}^{n} P_i = 1. \]

The problem of solving for \( P \) is thereby reduced to solving this system of \( n \) equations for the \( n \) unknown elements of \( P \).

This system of equations could be solved by the method of Gaussian elimination in a straightforward manner on a digital computer. However, the computer storage requirements would be proportional to \( n^2 \) and the computational time would be roughly proportional to \( n^3 \). Unfortunately, \( n \) is very large unless \( N \) and the \( S_j \) are very small. (For example, \( N = 4 \) and the \( S_j = 3 \) yields \( n = 204 \).) Therefore, special techniques to reduce the storage requirements and computational time would be valuable.

A special method for storing the matrix \( Q \) in the computer takes advantage of the fact that there are at most \((N + 1)\) non-zero elements in each column of \( Q \). Therefore, all information concerning \( Q \) can be stored in a \( n \times (N + 1) \times 2 \) three-dimensional array, where the array element \( A[i, j, 1] \) is the number of the row containing the \( j^{th} \) non-zero element in the \( i^{th} \) column of \( Q \), and the array element \( A[i, j, 2] \) is the value of that non-zero element.

An efficient iterative method for solving a large system of linear equations with a high proportion of zero coefficients is the Gauss-Seidel
method (see [10], p. 39), which is also called the Liebmann extrapolated method. The Aitken convergence accelerating procedure (see [10], p. 123) also can be applied at a late stage of the computations. Unfortunately, the Gauss-Seidel method is not guaranteed to converge except under fairly strong conditions (see [14]). The sufficient condition for convergence does not hold in general for the problem under consideration. Nevertheless, the method did converge in every case attempted when the following technique was used. The $n^{th}$ homogeneous equation was replaced by
\[ \sum_{i=1}^{n} P_i = 1 \] and then the coefficient of $P_n$ in this equation was made sufficiently large that its absolute value was greater than the sum of the other coefficients in this equation. A modification of the Gauss-Seidel method known as the "method of Nekrasov" (see [4]) then converged to a solution vector for this revised set of equations which was proportional to the desired solution. This solution was scaled to satisfy the condition that $\sum_{i=1}^{n} P_i = 1$, which yielded the desired result. If this method should fail to converge, one can revert to the method of Gaussian elimination.

Given the stationary distribution for the Markov chain, $R$ and $L$ would be calculated as follows. Letting $P_m$ be the sum of the probabilities of the states corresponding to $m$ customers in the system, then
\[ L = \sum_{m=0}^{\infty} m P_m, \]

(where $P_0 = 0$ and $P_m = 0$ for $m > \sum_{j=2}^{N} ((S_j + 1) + 1)$. 

8
For the case of exponential service times, if $B$ is the set of states corresponding to the last facility being busy, then

$$R = \mu_N \sum_{j \in B} P_j.$$  

For the case of Erlang service times, if $E$ is the set of states corresponding to the last facility being in its last exponential service phase, then

$$R = k N \mu_N \sum_{j \in E} P_j.$$  

3. **Procedure for Approximating $R$ Efficiently**

The procedure described above is the most efficient one known to the authors for computing $R$ (and $L$). However, as the next section indicates, even this procedure is not computationally feasible with today's digital computers except for small problems. Therefore, a new more efficient procedure was developed for obtaining an approximate value of $R$ for the case of exponential service times. The computational time required for this procedure is roughly proportional to the number of facilities. Therefore, it is computationally feasible for even very large problems. Furthermore, as the next section demonstrates, this procedure provides an excellent approximation for most cases including those for which the exact procedure is less likely to be computationally feasible.

The motivation for this procedure was provided by the result due to Burke [3] which implies that, if all of the $S_j$ were equal to infinity,
(j = 2, 3, ..., N), each facility would have a Poisson input. Therefore, since Reich [12] shows that the queue sizes would be independent, each facility could then be analyzed independently of the others by using the standard single-server queueing model with a Poisson input, exponential service times, and an infinite queue, i.e., M/M/1, (see [8]).

For the problem under consideration, all of the \( S_j \) are finite \( (j = 2, 3, ..., N) \). However, it appears that the output process for each facility (and therefore the input process to the next facility) should still be approximately Poisson (excluding when blocking occurs). Therefore, as an approximation, each facility could be analyzed individually by using the single-server queueing model with a Poisson input, exponential service times, and a finite queue (see [8]). This is a valuable simplification since \( R \) is merely the mean output rate for each of the facilities.

The one complication that arises is that it is not obvious what the effective mean arrival rate (when there isn't blocking) and the effective mean service rate are for each facility. To determine this, the remainder of the queueing system should be considered a "black box" which generates an input process into the facility and always accepts the output of the facility. This concept is illustrated in Figure 1.
Consider first the effective mean arrival rate for the $j^{th}$ facility, $\lambda_{\text{eff}}^{(j)}$, (where $j = 2, 3, \ldots, N$). It is only necessary to consider the case where the queue (including any customer being held at the preceding facility) is not full, since the rate does not apply otherwise. For this case, the mean arrival rate is $\mu_{j-1}$ over the time intervals when the $(j-1)^{st}$ facility is busy and it is zero otherwise. Assume as an approximation that the queue lengths of the facilities are independent, so that the unconditional probability that the $(j-1)^{st}$ facility is busy is equal to the conditional probability that it is busy, given that the queue for the $j^{th}$ facility is not full. It then follows that

$$\lambda_{\text{eff}}^{(j)} = \begin{cases} 
\mu_{j-1}(1 - P_0^{(j-1)}) & , \text{if } n \leq S_j + 1 \\
0 & , \text{if } n = S_j + 2 
\end{cases}$$
where \( p_{0}(j) \) is the unconditional probability that there are zero customers at the \((j-1)^{st}\) facility, (so that \( p_{0}(1) = 0 \)), \( n \) is the number of customers in the queueing system for the \( j^{th} \) facility (including any one being served there or any customer being held at the \((j-1)^{st}\) facility), and \( j = 2, 3, \ldots, N \).

The effective mean service rate for the \( j^{th} \) facility, \( \mu_{\text{eff}}^{(j)} \), is somewhat less than \( \mu_j \) (for \( j = 2, 3, \ldots, N-1 \)) because time spent in holding a customer already served must be included in the effective service time in order to yield the correct mean output rate. To determine \( \mu_{\text{eff}}^{(j)} \), recall that the mean output rate for a single-server queueing system equals the product of the mean service rate and the probability that the server is busy. Therefore,

\[
\mu_{\text{eff}}^{(j)} = \frac{R}{1 - p_{0}^{(j)}}, \quad \text{for } j = 2, 3, \ldots, N.
\]

For the last facility, customers always are released immediately when service is completed, so that

\[
\mu_{\text{eff}}^{(N)} = \mu_N.
\]

Therefore,

\[
R = \mu_N(1 - p_{0}^{(N)}),
\]

so that the problem is reduced to finding \( p_{0}^{(N)} \).
For the approximate queueing theory model being used for the $j^{th}$ facility (see Figure 1), it is known (see [6]) that

$$1 - p_0(j) = \frac{\rho_j^{j+2}}{1 - \rho_j^{j+2}} \frac{1}{S_j^{j+3}},$$

where

$$\rho_j = \frac{\lambda_{j}^{\text{eff}}}{\mu_{j}^{\text{eff}}}. $$

Unfortunately, this does not yield $p_{0}^{(N)}$ immediately since knowing $\lambda_{N}^{\text{eff}}$ requires knowing $p_{0}^{(N-1)}$. To make matters worse, for $j = 2, 3, \ldots, N-1$, $\lambda_{j}^{\text{eff}}$ and $\mu_{j}^{\text{eff}}$ are functions of $p_{0}^{(j-1)}$ and of $p_0(j)$ and $R$, respectively, where only $p_0^{(1)}$ is known at the outset.

However, by assuming a value of $R$, it is possible to find the corresponding value of $p_0^{(2)}$ by numerical methods, and then to find the corresponding value of $p_0^{(3)}$ by numerical methods, etc., until a value of $R$ can finally be calculated. By repeating this successively, the calculated values of $R$ can be made to converge by numerical methods to the desired approximate value of $R$.

To motivate the procedure summarized above, notice that
\[
\rho_j = \begin{cases} 
\frac{\mu_{j-1}(1 - P_0^{(j-1)})(1 - P_0^{(j)})}{R}, & \text{if } j = 2, 3, \ldots, N-1 \\
\frac{\mu_{N-1}}{\mu_0}(1 - P_0^{(N-1)}) & \text{if } j = N,
\end{cases}
\]

so that

\[1 - P_0^{(j)} = \frac{R}{\mu_{j-1}(1 - P_0^{(j-1)})} \rho_j, \quad \text{if } j = 2, 3, \ldots, N-1.\]

Therefore, \( \rho_j \) is the positive root of the equation,

\[\frac{y_j(1 - y_j^{j+2})}{y_j^{j+3}} = \frac{R}{\mu_{j-1}(1 - P_0^{(j-1)})} y_j = 0,\]

which can be found by numerical methods, so that \( 1 - P_0^{(j)} \) can then be calculated. Thus, given a trial value of \( R \) and \( P_0^{(1)} = 0 \), one can obtain \( P_0^{(2)}, P_0^{(3)}, \ldots, P_0^{(N-1)} \) successively. Given \( P_0^{(N-1)} \), the corresponding value of \( P_0^{(N)} \) and \( R \) are easily calculated. The difference between this calculated value of \( R \) and the trial value is a function of the trial value. Setting this function equal to zero and finding the root by numerical methods yields the desired approximation of \( R \). The details of this procedure are outlined below.
Let

\[ f_j(y_j) = \frac{y_j(1 - y_j^{S_j+2})}{1 - y_j^{S_j+3}}, \quad \text{for } j = 2, 3, \ldots, N-1, \]

where \( y_j \geq 0 \). Let

\[ g_j(x_N, y_j) = f_j(y_j) - c_j y_j, \quad \text{for } j = 2, 3, \ldots, N-1, \]

where \( c_j \) is a function of \( x_N \) (where \( x_N \) corresponds to a trial value of \( R/\mu_N \)) which will be defined shortly. (In order to simplify the notation, the argument of functions of \( x_N \) usually will be suppressed.) For a fixed value of \( x_N \), define \( y_j^* \) as the unique strictly positive real root (assuming it exists) of the equation,

\[ g_j(x_N, y_j) = 0, \quad \text{(for } j = 2, 3, \ldots, N-1), \]

and is shown in Figure 2; \((y_j^*) \) corresponds to \( \rho_j \) for the given value of \( x_N \).
Figure 2. Definition of $y_j^*$ ($j = 2, 3, \ldots, N-1$).

Let

$$x_j^* = f_j(y_j^*)$$

for $j = 2, 3, \ldots, N-1$,

(so that $x_j^*$ corresponds to $1 - P_0^{(j)}$, given $x_N$). Define $C_j$ as

$$C_j = \frac{\mu_N x_N}{\mu_{j-1} x_j^*}$$

for $j = 2, 3, \ldots, N-1$,

where $x_1^* = 1$. It can be shown that, if $0 < C_j < 1$, then $y_j^*$ exists.

This follows from the facts that (1) $g_j(x_N, 0) = 0$, (2) $f_j'(0) = 1$, so that $\frac{\partial g_j(x_N, y_j)}{\partial y_j} > 0$, (3) $\lim_{y_j \to \infty} f_j'(y_j) = 0$, so that $\lim_{y_j \to \infty} \frac{\partial g(x_N, y_j)}{\partial y_j} < 0$, and (4) $g_j(x_N, y_j)$ is strictly concave with respect to $y_j$. The first three facts are trivial to show, and the last...
fact is verified in the Appendix. It is also shown in the Appendix that $0 < C_j < 1$ for the relevant values of $x_N$.

Let

$$h(x_N) = f_N \left( \frac{\mu_{N-1}}{\mu_N} \frac{x_N^*}{x_{N-1}} \right) - x_N,$$

(so that $h(x_N)$ corresponds to the difference between the calculated value and the trial value of $R/\mu_N$). Define $x_N^*$ as the unique positive real root (assuming it exists) of the equation,

$$h(x_N) = 0,$$

as is shown in Figure 3. Thus, $\mu_N x_N^*$ is the desired approximation of $R$.

Figure 3. Definition of $x_N^* = \text{approximation of } \frac{R}{\mu_N}$.
The result that $x_N^*$ indeed does exist follows from the facts that

1. $\lim_{x_N \to 0} h(x_N) = f_N \left( \frac{\mu_{N-1}}{\mu_N} \right) > 0$ (for $0 < \mu_{N-1}, \mu_N < \infty$),
2. $h(x_N)$ is continuous,
3. $h'(x_N) \leq -1$, and
4. there exists a constant $b > x_N^* > 0$ such that $h(x_N)$ is defined (i.e., $0 < C_j < 1$ for $j = 2, 3, \ldots, N-1$) over the interval, $(0, b)$. The first two facts are obvious, whereas the last two facts will be verified in the Appendix.

Solving for the $y_j^*$ ($j = 2, 3, \ldots, N-1$) and $x_N^*$ requires the use of numerical methods. Since $\frac{\partial g_j(x_N, y_j)}{\partial y_j}$ can be expressed explicitly, $y_j^*$ can be obtained to within a specified error by the Newton-Raphson method (see [6], pp. 447 ff.). Because of the properties of $g_j(x_N, y_c)$ summarized earlier, this method is guaranteed to converge to $y_j^*$, provided only that the initial trial solution exceeds the value at which $\frac{\partial g_j(x_N, y_j)}{\partial y_j} = 0$. Solving for $x_N^*$ is a little more difficult since $h'(x_N)$ is not known. However, the method of "false position" (regula falsi) may be used (see [6], pp. 446 ff.) after finding two initial trial solutions, $x_N^{(1)}$ and $x_N^{(2)}$, such that $h(x_N^{(1)}) \geq 0$ and $h(x_N^{(2)}) \leq 0$. Finding $x_N^{(1)}$ and $x_N^{(2)}$ is relatively easy since $h'(x_N) < -1$, although care needs to be taken to select an $x_N^{(2)}$ such that $h(x_N^{(2)})$ is defined. Because of the properties of $h(x_N)$ given earlier, this method is certain to converge to $x_N^*$ to within a specified error.

The approximate procedure may now be summarized as follows. Solve for $x_N^*$, the approximation of $R/\mu_N$, by using the method of "false position" to find the positive root of $h(x_N) = 0$. To obtain $h(x_N)$ corresponding to each new trial value of $x_N^*$, apply the Newton-Raphson
method to solve for $y_j^*$ (and therefore $x_j^*$) for $j = 2, 3, \ldots, N-1$, and then use the resulting value of $x_{N-1}^*$ to calculate $h(x_N^*)$.

4. Numerical Results

The authors have applied both the exact procedure and the approximate procedure described in the preceding two sections in order to obtain comprehensive numerical results for the case where

$\mu_1 = \mu_2 = \cdots = \mu_N (= \mu), \; k_1 = k_2 = \cdots = k_N (= k)$, and

$S_2 = S_3 = \cdots = S_N (= S)$. Tables I and II both give values of the utilization, $\rho_{\text{MAX}} = \frac{R}{\mu}$, and the mean number of customers in the system, $L$, calculated by the exact procedure for the case of Erlang service times $(k > 1)$. Tables III and IV give values of $L$ and $\rho_{\text{MAX}}$, respectively, calculated by the exact procedure for the case of exponential service times $(k = 1)$. Table V gives values of $\rho_{\text{MAX}}$ calculated by the approximate procedure for the case of exponential service times. Table VI gives the correction quantities for Table V, i.e., the amounts that need to be added to the entries in Table V in order to obtain the entries in Table IV.
Table I

Calculated Values of $\rho_{\text{MAX}}$ and $L$ for $N = 2$, $k \geq 5$

<table>
<thead>
<tr>
<th>S</th>
<th>k</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\rho_{\text{MAX}}$</td>
<td>0.8025</td>
<td>0.8159</td>
<td>0.8268</td>
<td>0.8358</td>
<td>0.8436</td>
<td>0.8502</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>1.8025</td>
<td>1.8159</td>
<td>1.8268</td>
<td>1.8358</td>
<td>1.8436</td>
<td>1.8502</td>
</tr>
<tr>
<td>1</td>
<td>$\rho_{\text{MAX}}$</td>
<td>0.8985</td>
<td>0.9106</td>
<td>0.9200</td>
<td>0.9275</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>2.3985</td>
<td>2.4106</td>
<td>2.4200</td>
<td>2.4275</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$\rho_{\text{MAX}}$</td>
<td>0.9326</td>
<td>0.9418</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>2.9326</td>
<td>2.9418</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$\rho_{\text{MAX}}$</td>
<td>0.9496</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>3.4496</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table II
Calculating Values of $p_{\text{MAX}}$ and $L$ for $N \geq 2$, $k > 1$

<table>
<thead>
<tr>
<th>$N$</th>
<th>$k = 2$</th>
<th>$k = 3$</th>
<th>$k = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N = 2$</td>
<td>$N = 3$</td>
<td>$N = 2$</td>
</tr>
<tr>
<td>0</td>
<td>$p_{\text{MAX}}$</td>
<td>0.7273</td>
<td>0.6404</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>1.7273</td>
<td>2.4755</td>
</tr>
<tr>
<td>1</td>
<td>$p_{\text{MAX}}$</td>
<td>0.8214</td>
<td>0.7613</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>2.3214</td>
<td>3.6704</td>
</tr>
<tr>
<td>2</td>
<td>$p_{\text{MAX}}$</td>
<td>0.8682</td>
<td>0.8219</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>2.8682</td>
<td>4.7645</td>
</tr>
<tr>
<td>3</td>
<td>$p_{\text{MAX}}$</td>
<td>0.8957</td>
<td>0.9235</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>3.3957</td>
<td>3.4235</td>
</tr>
<tr>
<td>4</td>
<td>$p_{\text{MAX}}$</td>
<td>0.9137</td>
<td>0.9378</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>3.9137</td>
<td>3.9378</td>
</tr>
<tr>
<td>5</td>
<td>$p_{\text{MAX}}$</td>
<td>0.9264</td>
<td>0.9475</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>4.4264</td>
<td>4.4475</td>
</tr>
<tr>
<td>6</td>
<td>$p_{\text{MAX}}$</td>
<td>0.9358</td>
<td>0.9547</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>4.9358</td>
<td>4.9547</td>
</tr>
<tr>
<td>7</td>
<td>$p_{\text{MAX}}$</td>
<td>0.9431</td>
<td>0.9601</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>5.4431</td>
<td>5.4601</td>
</tr>
<tr>
<td>8</td>
<td>$p_{\text{MAX}}$</td>
<td>0.9489</td>
<td>0.9644</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>5.9489</td>
<td>5.9644</td>
</tr>
<tr>
<td>9</td>
<td>$p_{\text{MAX}}$</td>
<td>0.9537</td>
<td>0.9678</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>6.4537</td>
<td>6.4678</td>
</tr>
<tr>
<td>10</td>
<td>$p_{\text{MAX}}$</td>
<td>0.9576</td>
<td>0.9706</td>
</tr>
<tr>
<td></td>
<td>$L$</td>
<td>6.9576</td>
<td>6.9706</td>
</tr>
</tbody>
</table>
Table III
Calculated Values of $L$ for Exponential Service Times

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>2.3590</td>
<td>3.0646</td>
<td>3.7786</td>
<td>4.4983</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3.5318</td>
<td>4.8291</td>
<td>6.1354</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4.6355</td>
<td>6.4821</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5.7001</td>
<td>8.0813</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>6.7470</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>7.7817</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>8.8082</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>9.8292</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>10.8462</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>11.8602</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>12.8722</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
(\text{For } N = 2, \quad L = \frac{S + 4}{2} - \frac{1}{S + \frac{3}{2}}).
\]
Table IV

Calculated Values of $\rho_{\text{MAX}}$ for Exponential Service Times

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0.5641</td>
<td>0.5148</td>
<td>0.4858</td>
<td>0.4667</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0.6705</td>
<td>0.6312</td>
<td>0.6076</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.7340</td>
<td>0.7007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.7767</td>
<td>0.7477</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.8075</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.8308</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.8490</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>0.8637</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0.8757</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0.8858</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.8944</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[(\text{For } N = 2, \quad \rho_{\text{MAX}} = \frac{S + 2}{S + \frac{S}{3}})\]
Table V
Approximate Values of $c_{\text{MAX}}$ for Exponential Service Times

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.5698</td>
<td>0.5263</td>
<td>0.5025</td>
<td>0.4880</td>
<td>0.4783</td>
<td>0.4716</td>
<td>0.4668</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.6710</td>
<td>0.6343</td>
<td>0.6138</td>
<td>0.6011</td>
<td>0.5927</td>
<td>0.5868</td>
<td>0.5824</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.7339</td>
<td>0.7025</td>
<td>0.6849</td>
<td>0.6739</td>
<td>0.6666</td>
<td>0.6613</td>
<td>0.6576</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.7766</td>
<td>0.7494</td>
<td>0.7340</td>
<td>0.7243</td>
<td>0.7179</td>
<td>0.7133</td>
<td>0.7100</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.8076</td>
<td>0.7836</td>
<td>0.7699</td>
<td>0.7614</td>
<td>0.7556</td>
<td>0.7516</td>
<td>0.7486</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.8310</td>
<td>0.8096</td>
<td>0.7973</td>
<td>0.7896</td>
<td>0.7845</td>
<td>0.7808</td>
<td>0.7781</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.8494</td>
<td>0.8300</td>
<td>0.8189</td>
<td>0.8120</td>
<td>0.8073</td>
<td>0.8039</td>
<td>0.8015</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.8644</td>
<td>0.8465</td>
<td>0.8364</td>
<td>0.8300</td>
<td>0.8257</td>
<td>0.8227</td>
<td>0.8204</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.8762</td>
<td>0.8600</td>
<td>0.8507</td>
<td>0.8449</td>
<td>0.8409</td>
<td>0.8381</td>
<td>0.8361</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0.8864</td>
<td>0.8714</td>
<td>0.8628</td>
<td>0.8574</td>
<td>0.8537</td>
<td>0.8511</td>
<td>0.8493</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.8950</td>
<td>0.8811</td>
<td>0.8731</td>
<td>0.8680</td>
<td>0.8646</td>
<td>0.8622</td>
<td>0.8604</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0.9508</td>
<td>0.9441</td>
<td>0.9402</td>
<td>0.9377</td>
<td>0.9360</td>
<td>0.9348</td>
<td>0.9340</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0.9866</td>
<td>0.9847</td>
<td>0.9836</td>
<td>0.9829</td>
<td>0.9824</td>
<td>0.9821</td>
<td>0.9818</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.4631</td>
<td>0.4536</td>
<td>0.4499</td>
<td>0.4481</td>
<td>0.4454</td>
<td>0.4447</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.5792</td>
<td>0.5707</td>
<td>0.5674</td>
<td>0.5658</td>
<td>0.5634</td>
<td>0.5627</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.6547</td>
<td>0.6473</td>
<td>0.6443</td>
<td>0.6429</td>
<td>0.6408</td>
<td>0.6402</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.7075</td>
<td>0.7009</td>
<td>0.6983</td>
<td>0.6970</td>
<td>0.6951</td>
<td>0.6946</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.7463</td>
<td>0.7405</td>
<td>0.7381</td>
<td>0.7370</td>
<td>0.7353</td>
<td>0.7348</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.7761</td>
<td>0.7708</td>
<td>0.7687</td>
<td>0.7677</td>
<td>0.7662</td>
<td>0.7658</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.7997</td>
<td>0.7949</td>
<td>0.7929</td>
<td>0.7920</td>
<td>0.7906</td>
<td>0.7902</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.8187</td>
<td>0.8143</td>
<td>0.8126</td>
<td>0.8117</td>
<td>0.8105</td>
<td>0.8101</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.8345</td>
<td>0.8305</td>
<td>0.8288</td>
<td>0.8280</td>
<td>0.8269</td>
<td>0.8265</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0.8478</td>
<td>0.8440</td>
<td>0.8425</td>
<td>0.8417</td>
<td>0.8407</td>
<td>0.8404</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.8591</td>
<td>0.8555</td>
<td>0.8541</td>
<td>0.8534</td>
<td>0.8524</td>
<td>0.8522</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0.9333</td>
<td>0.9316</td>
<td>0.9309</td>
<td>0.9305</td>
<td>0.9300</td>
<td>0.9299</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0.9816</td>
<td>0.9812</td>
<td>0.9811</td>
<td>0.9809</td>
<td>0.9808</td>
<td>0.9807</td>
</tr>
</tbody>
</table>
Table VI

Correction Quantities for Table V

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>-0.0057</td>
<td>-0.0115</td>
<td>-0.0167</td>
<td>-0.0213</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>-0.0005</td>
<td>-0.0031</td>
<td>-0.0062</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>+0.0001</td>
<td>-0.0018</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>+0.0001</td>
<td>-0.0017</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>-0.0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>-0.0002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>-0.0004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>-0.0004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>-0.0005</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>-0.0006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>-0.0006</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table VI was constructed to aid both in evaluating the approximate procedure and in extrapolating to the correction quantities for Table V where the exact values of $\rho_{\text{MAX}}$ are not available in Table IV. As Table VI indicates, the approximate procedure is relatively imprecise for $S = 0$, especially as $N$ increases, but it becomes much more precise as $S$ increases. This is intuitively plausible since the approximation that each facility has a Poisson input and exponential effective service times is especially gross for $S = 0$, whereas it approaches exactness as $S \to \infty$. 
The error tolerances used were $10^{-6}$ for the Gauss-Seidel method, $10^{-8}$ for the Newton-Raphson method, and $5 \times 10^{-5}$ for the method of "false position." Extensive sample checks indicate that, except for the approximations incorporated into the approximate procedure and possibly computer round-off error, the numerical results given are accurate to within one unit in the fourth decimal place. The time required to compute each value of $\rho_{\text{MAX}}$ given ranged up to approximately 20 minutes (on an IBM 7090) for the exact procedure and up to 40 seconds (on a Burroughs-5500) for the approximate procedure.
APPENDIX

Properties of the Approximate Procedure

Lemma 1: \( g(y) = \frac{y(1 - y^{S+2})}{1 - y^{S+3}} - C \) is a strictly concave function for \( y > 0 \), where \( S \) and \( C \) are non-negative constants.

Proof: It is sufficient to show that \( g''(y) < 0 \) for \( y > 0 \). However,

\[
g''(y) = f_1(y)f_2(y),
\]

where

\[
f_1(y) = \frac{(S + 3)y^{S+1}}{[1 - y^{S+3}]^2}
\]

and

\[
f_2(y) = (S + 4)y - (S + 2) - y^{S+3}[S + 4 - (S + 2)y] .
\]

Since

\[
f_1(y) = \begin{cases} > 0, & \text{if } 0 < y < 1 \\ < 0, & \text{if } y > 1 \end{cases}
\]

and L'Hopital's rule shows that \( g''(1) < 0 \), it is sufficient to show that
\[
\begin{align*}
    f_2(y) &= \begin{cases} 
        < 0, & \text{if } 0 < y < 1 \\
        > 0, & \text{if } y > 1.
    \end{cases}
\end{align*}
\]

Therefore, since \( f_2(1) = 0 \), it is sufficient to show that \( f'_2(y) > 0 \) for \( y > 0 \) and \( y \neq 1 \). Note that

\[
f'_2(y) = (s + 4)[1 - y^{s+2}(s + 3 - (s + 2)y)],
\]

so that \( f'_2(1) = 0 \), so that it is sufficient to show that

\[
f''_2(y) &= \begin{cases} 
        < 0, & \text{if } 0 < y < 1 \\
        > 0, & \text{if } y > 1.
    \end{cases}
\]

However, this is observed to be true since

\[
f''_2(y) = (s + 4)(s + 3)(s + 2)[y^{s+2} - y^{s+1}],
\]

which completes the proof.

**Lemma 2**: \( h'(x_N) \leq -1 \) for \( x_N > 0 \) in the domain of \( h(\cdot) \).

**Proof**: Since

\[
h(x_N) = f_N \left( \frac{\mu_{N-1}}{\mu_N} x_{N-1}^* \right) - x_N
\]

and \( f_N(\cdot) \) is a monotone increasing function, where \( \mu_N > 0, \mu_{N-1} > 0 \), it is sufficient to show that \( x_{N-1}^* \) is a monotone decreasing function of \( x_N \). This is trivially true for \( N = 2 \) (\( x_1^* \) is a constant), so assume
\( N \geq 3 \). Consider any two distinct positive values of \( x_N \), \( x_N^{(1)} < x_N^{(2)} \). Thus, \( C_2(x_N^{(1)}) < C_2(x_N^{(2)}) \), so that it follows from the properties of \( \varepsilon_2(x_N, y_N) \) that \( y_N^*(x_N^{(1)}) > y_N^*(x_N^{(2)}) \). Since \( f_2'(y_N) > 0 \), this implies that \( x_N^*(x_N^{(1)}) > x_N^*(x_N^{(2)}) \). Therefore, \( C_3(x_N^{(1)}) < C_3(x_N^{(2)}) \), so that it follows as before that \( x_N^*(x_N^{(1)}) > x_N^*(x_N^{(2)}) \), so that \( x_N^*(x_N^{(1)}) > x_N^*(x_N^{(2)}) \), \( \ldots \), \( x_N^{N-1}(x_N^{(1)}) > x_N^{N-1}(x_N^{(2)}) \), which completes the proof.

**Lemma 3:** There exists a constant \( b > x_N^* > 0 \) such that \( h(x_N) \) is defined (i.e., \( 0 < C_j(x_N) < 1 \) for \( j = 2, 3, \ldots, N-1 \)) over the interval, \( 0 < x_N < b \).

**Proof:** The lemma is trivially true for \( N = 2 \), so assume \( N \geq 3 \).

Recall that it was shown during the proof of Lemma 2 that the \( C_j(x_N) \) are strictly monotone increasing functions and that the \( x_N^*(x_N) \) are strictly monotone decreasing functions (\( j = 2, 3, \ldots, N-1 \)). Also note that the \( x_N^*(x_N) \) are continuous functions such that \( 0 < x_N^*(x_N) < 1 \), that \( \lim_{C_j(x_N) \to 1} x_N^*(x_N) = 0 \) and \( \lim_{C_j(x_N) \to 0} x_N^*(x_N) = 1 \), and that

\[
\lim_{x_N \to 0} x_N^*(x_N) = 1, \quad \text{for } j = 2, 3, \ldots, N-1.
\]

It will now be shown by induction that the range of each of the \( x_N^*(x_N) \) functions is the interval \((0, 1)\).

Consider \( x_N^*(x_N) \). Notice that \( C_2(0) = 0 \) and \( C_2\left(\frac{\mu_1}{\mu_N}\right) = 1 \), so that

\[
\lim_{x_N \to 0} x_N^*(x_N) = 0 \quad \text{and} \quad \lim_{x_N \to \frac{\mu_1}{\mu_N}} x_N^*(x_N) = 1, \quad \text{whereas} \quad x_N^*(x_N) \text{ is undefined.}
\]
outside the interval, \( 0 < x_N \leq \frac{\mu_1}{\mu_N} \). Thus, the range of \( x_N^*(x_N) \) is the interval, \((0, 1)\). Now assume that the range of \( x_{j-1}^*(x_N) \) is the interval, \((0, 1)\). Hence, there must exist a value of \( x_N' \), call it \( x_N' \), such that \( 0 < \frac{\mu_{j-1}}{\mu_N} x_{j-1}^*(x_N') < x_N' \), so that \( C_j(x_N') > 1 \). Since 

\[ \lim_{x_N \to 0^+} C_j(x_N) = 0 \quad \text{and} \quad C_j(x_N) \text{ is continuous,} \]

it therefore follows that the range of \( C_j(x_N) \) includes the interval, \((0, 1)\). This implies that the range of \( x_N^*(x_N) \) is the entire interval, \((0, 1)\). Therefore, by the induction argument, the range of \( x_N^*(x_N) \), of \( x_N^*(x_N) \), ..., and of \( x_N^*(x_N) \) is the interval, \((0, 1)\).

Let \( b = \lim_{x_N \to 0} x_{N-1}^*(x_N) \), where the function \( x_{N-1}^*(x_N) \) is the inverse of \( x_N^*(x_N) \). Thus, the domain of \( h(x_N) \) is the interval, \( 0 < x_N < b \). Furthermore, since \( \lim_{x_N \to 0^+} h(x_N) = f_N \left( \frac{\mu_{N-1}}{\mu_N} \right) > 0 \) and 

\[ \lim_{x_N \to b^-} h(x_N) = -b < 0, \]

it follows from Lemma 2 that \( 0 < x_N^* < b \). This completes the proof.
REFERENCES


Finite Queues in Series With Exponential or Erlang Service Times

This paper considers a queueing system consisting of N service channels in series where each channel has an exponential or Erlang holding time and (except for the first channel) a finite queue, and where the input process is such that the first queue is never empty. The measures considered are the steady-state mean output rate and mean number of customers in the system (excluding the first queue). First, a procedure is described for obtaining these measures which is relatively efficient computationally. Second, an exceptionally efficient procedure is developed for approximating the mean output rate for the case of exponential holding times. It is demonstrated that this procedure provides an excellent approximation for most cases and that it is computationally feasible for large problems. Third, extensive new numerical results are obtained.
queueing theory
finite queues in series
quality control systems
sequence of inspection stations
production line systems
operations research

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (corporate author) issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Date" is included. Marking to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b & 8c. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (either by the originator or by the sponsor), also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

   (1) "Qualified requesters may obtain copies of this report from DDC."

   (2) "Foreign announcement and dissemination of this report by DDC is not authorized."

   (3) "U.S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through"

   (4) "U.S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through"

   (5) "All distribution of this report is controlled. Qualified DDC users shall request through"

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (paying for) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (T3), (S), (C), or (U).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical content. The assignment of links, roles, and weights is optional.