75-21,833

TU, Hung-Yuan, 1936-
THE DESIGN OF PARALLEL CHANNEL QUEUEING SYSTEMS.

The University of Oklahoma, Ph.D., 1975
Operations Research

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

THE DESIGN OF PARALLEL CHANNEL QUEUEING SYSTEMS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

By

HUNG-YUAN TU

Norman, Oklahoma

1975

THE DESIGN OF PARALLEL CHANNEL QUEUEING SYSTEMS

APPROVED BY

B L Foote

Marilyn Breen

Hillel Kumin

Michael D. Devine

Jerry L. Purswell

DISSERTATION COMMITTEE

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Dr. Hillel J. Kumin, for his help, guidance, and confidence in me. His patience and skillful assistance aided in the completion of this dissertation.

Thanks also go to Dr. Marilyn J. Breen for her help in resolving certain mathematical difficulties and to Dr. Michael D. Devine, Dr. Bobbie L. Foote, and Dr. Jerry L. Purswell for their comments and interest in the research.

Finally, I owe my deepest gratitude to my wife, Fu-Chiu, for her patience, understanding, and support during the entire research period.

ABSTRACT

An approach is proposed for designing a class of parallel channel Markovian queueing systems. The approach calls for estimating the expected number of customers of a particular system from its transition matrix. Two algorithms are presented to estimate the expected number of customers from transition matrices. The algorithms allow one to solve a design problem whose measures of effectiveness are the expected number of customers or the expected waiting time without needing closed formed expressions for these measures.

A two parameter design problem for a parallel channel system is then considered in which the design parameters are the service rate and the number of servers. An algorithm is developed to take advantage of the special structure of the problem. The convexity of the objective function is investigated and numerical results are presented.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED) Page

Chapter I

INTRODUCTION

In the first decade of this century, A. K. Erlang, an employee of the Copenhagen Telephone Company, devoted himself to the investigation of the effects of fluctuations in demand on the operation of telephone systems. His research resulted in the publication of "The Theory of Probabilities and Telephone Conversations," which became the first queueing model on record. Since then, studies in the field of queueing theory have been greatly accelerated. According to a study made by Morse [17], there were more than 700 papers and books published up to 1960 with applications extending from telephone traffic to areas such as machine servicing and maintenace, road traffic, railroads, air transport, inventories, production, hydro-storage, health, and physics. The development of queueing theory seems to have been dominated by studies aimed at understanding the behavior of specific systems. Unfortunately, few formal studies have been made on putting these ideas into practice. As a result, queueing theory has been attacked on two fronts. Some theoreticians say that queueing theory is closed. However, some practitioners feel that the current theory has little practical use [2]. These two conflicting views between the practitioners and the theoreticians could be eliminated if the theoreticians shift part of their attention from behavioral problems to operational problems.

There are two major types of operational problems in queueing theory: design problems and control problems. A control problem differs from a design problem in that the former is dynamic in nature whereas the latter is static. While a control problem tries to seek an optimal operating policy for a given design, a design problem attempts to make a single choice of queueing system given a set of initial conditions [9],[14]. One of the recent aspects in the development of queueing theory has been the increased amount of research directed towards the optimal control of queueing systems. Such research should prove most effective in reducing the gap between the theoreticians and the practitioners. Regretfully, a large research effort has not yet extended into the design aspects of queueing theory. Design problems have been studied formally as optimization problems by Morse (1958), Bowman and Fetter (1961), De Cani (1962), Hillier (1963), Kumin (1968), Evans (1968), Balachandran (1970), Stidham (1970), and Rolfe (1971). Most of these studies, unfortunately, emphasized setting up models for a specific application while few tried to develop a general design methodology. In a field such as queueing theory that abounds with special cases, it is of interest to ask what is the chance that a practitioner will find an existing model which is realistic enough to be used for solving the problem at hand. Thus, it seems clear that what a practitioner will appreciate most is a set of tools that can be used to set up his own problem and solve it rather than a long list of solved models.

One of the exceptions to the commonly adopted research approach of emphasizing modeling for special cases is the research done by Kumin [14] in 1968. In his work, Kumin developed an algorithm which determines the optimal mean service rate for a design problem of specified structure. His algorithm contains the following ideas:

1) Transition matrices are used to estimate the expected queue length instead of relying on closed form expressions for the steady state probabilities.
2) The design is for a class of queueing systems rather than a single one.

This thesis presents the research results pertaining to the design of parallel channel queueing systems utilizing the above two concepts of Kumin's work. It contains four major aspects: (a) the estimation of the expected number of customers directly from transition matrices without relying on any closed form expressions; (b) an investigation of the convexity of the expected number of customers as a function of mean service rate and the number of servers respectively; (c) the optimization of a two variable unconstrained nonlinear programming problem whose objective function is a convex function of a continuous variable and a discrete convex function of a discrete variable; (d) the numerical implementation of the algorithm described in (c). The research is motivated by the need for formal research in the area of applications of queueing theory - especially in regard to optimal design. It is hoped that the research will utlimately stimulate the development of a unified approach for solving design problems.

Chapter II

PREVIOUS DESIGNS OF QUEUEING SYSTEMS

The design of queueing systems has been studied formally as optimization problems by Morse [17], Bowman and Fetter [3], De Cani [6], Hillier [12], Kumin [14], Evans [8], Balachandran [1], and Rolfe [21]. Most of these studies emphasize models for specific applications. Their analyses are, in general, carried out on Poisson queueing models.

Morse considers three models. The first model is to balance service cost and customers lost. He assumes that the cost of service is directly proportional to the speed of the service and that the average sales corresponding to a single service operation yields a fixed amount of gross profit. He then sets up the net profit function for the M/M/1 case and finds the optimal mean service rate using classical calculus techniques. The second model is to balance the cost of waiting and the cost of service. Here the cost of waiting is assumed to be proportional to the mean waiting time. Again, classical calculus techniques are used to find the optimal mean service rate which minimizes the cost function for the M/M/1 case. The third model optimizes the number of servers. Here the intention is to maximize the net profit for given values of arrival rate, service rate, and average gross profit per customer served. This is carried out for the M/M/S model using a total enumeration technique.

Bowman and Fetter present a model for determining the optimal number of machines to assign to each operator based on a cost function which consists of the cost of machine waiting and the cost of operator. The machine waiting times are tabulated for the case of constant service time and the case of exponential service time respectively under the assumption that calls for service arrive at random. The optimal number of machines assigned is determined by comparing the total costs among all alternatives.

De Cani proposes a design model which is associated with a balking type queueing system. The model permits a solution in terms of expected profit maximization rather than cost minimization. The principal attribute of the model is that the arrival rate increases as the length of the waiting line decreases. Thus the expected arrival rate and, therefore, the total revenue will increase as the number of servers is increased. Hence there is a marginal revenue as well as a marginal cost associated with an increase in the number of servers. The optimal number of servers is found by marginal analysis.

Hillier presents three economic models for queueing systems with infinite calling sources and infinite waiting spaces. All of these models assume that the total cost of waiting is proportional to the total time that all arrivals spend in the system. They also assume that the cost of service at each service facility is a linear function of the number of servers at the facility. The first model presented is for the simple case where the arrival rate and service rate are fixed and the number of servers must be determined. The second model is for the case where both the arrival rate and the number of servers must be determined, i.e., where both the number of service facilities to distribute among the entire population and the number of servers to assign to each facility must be determined. The third model is for the case where both the service rate and the number of servers must be determined. A few special cases of these models are solved for Poisson queueing systems using classical calculus techniques. For other cases he suggests that a trial and error approach be used to find the optimal solution.

Kumin proposes a procedure for solving a single variable design problem without relying on the closed form expression for the expected queue length. To illustrate how this is achieved, let

A = a NxN transition matrix whose element at ith row and jth column is defined as:

$$P_{ij} \equiv P\{X_n = i | X_{n-1} = j\}$$

where X_n is the outcome of nth transition.

C_1, C_2 = cost factors.

F = (0, 1, 2,..., N-1).

L = expected number of customers in the system.

P_o = the initial probability vector.

μ = mean service rate.

λ = mean arrival rate.

Consider the design problem

$$\min\ g(\mu) = C_1\mu + C_2L$$

$$\text{s. t.}\quad \mu > \lambda$$

The above is equivalent to

$$\min\ g(\mu) = C_1\mu + C_2[\lim_{z\to\infty}(FA^zP_o)] \tag{1}$$

$$\text{s. t.}\quad \mu > \lambda$$

Problem (1) does not require any closed form expression for the expected number of customers in the system. However, it is not an easy problem to solve since the transition matrix, A, has to be raised to an infinite power. Kumin's proposal for solving problem (1) consists of a sub-algorithm and a main algorithm. The subalgorithm solves problem (1) for a fixed finite z (i.e., finds μ_z^* that minimizes $g(\mu) = C_1\mu + C_2FA^zP_o$) using an iterative approach which starts with an arbitrary initial probability vector. The main algorithm gradually increases the magnitude of z and repetatively uses the subalgorithm to generate a series of μ_z^*'s which approaches μ^*, the optimal solution of problem (1).

Evans develops two algorithms for the problem of picking a locally optimal irreducible aperiodic Markov chain from among a set of such systems. The first algorithm is for a class of continuous parameter Markov systems. It uses an iterative scheme for approximating the derivatives of the state probabilities. This leads to a stopping rule for a gradient type algorithm which permits stopping at a local optimum. The second algorithm is for the problem of selecting the optimal value of a single discrete parameter. The algorithm is essentially the same as the first one except that the first differences are used in place of the derivatives.

Balachandran analyzes priority rules that are mixtures of preemptive and postponable rules characterized by certain parameters. His work assumes an M/G/1 queueing model and linear cost function of the expected waiting time and expected number of preemptions. Optimal rule for each priority class is obtained using classical calculus techniques or, in case of discrete parameter, using difference analysis method.

Rolfe considers the problem of allocating servers to a multiple facility service system where each facility consists of a number of parallel channels and the arrival processes are Poisson. The objective is to allocate servers to facilities to minimize the expected waiting time of customers in the system subject to the overall manpower restriction. Fox's marginal allocation procedure is suggested for obtaining the optimal allocation for the constant service time case.

From the above descriptions, it can be seen that the majority of the design problems developed in the past can be characterized as follows:

1) The emphasis is on setting up models for special cases rather than trying to develop a general methodology.
2) There is a reliance on closed-form expressions for measures of effectiveness.

3) Most cases are Poisson queueing models.

4) Most cases consider only a single design parameter.

Chapter III

ESTIMATION OF THE EXPECTED NUMBER OF CUSTOMERS

The objective function associated with a queueing design problem often is a function of various measures of effectiveness such as the expected number of customers in a system or the expected waiting time a customer spent in queue, etc. Unfortunately, of the myriad of queueing systems to be designed, only a few have known closed form expressions for these measures. Therefore, any design algorithm which relies on closed form expressions will clearly have a very limited area of application. This point was realized first by Kumin and reflected in his research in 1968. However, since his interest was primarily in solving design problems, the approach that he used to obtain the expected number of customers from transition matrices cannot be separated for independent use from his optimization algorithm. Since an independent algorithm that can be used to obtain the expected number of customers directly from transition matrices should have many useful applications, this chapter will be devoted to the development of such algorithm.

3.1 ANALYTIC ASPECTS

This section is concerned with the statement and proof of the only theorem that is required to develop an algorithm for obtaining the expected number of customers of a steady state system from the transition matrix of the system without relying on the closed form expression for the expected number of customers.

Consider the following notation:

A = a NxN transition matrix whose element at ith row and jth column is defined as:

$$P_{ij} \equiv Pr(X_n=j|X_{n-1}=i)$$

where X_n is the outcome of nth transition.

$F = (0, 1, 2, \ldots, N-1)^t$.

L = expected number of customers in the steady state system.

V = steady state probability matrix whose ith element will be denoted by v_{i-1}.

$A^z F = (w_o^{(z)}, \ldots, w_{N-1}^{(z)})^t = W^{(z)}$.

$\bar{w}^{(z)} = \max_i \{w_i^{(z)}: i=0, 1, \ldots, N-1\}$.

$\underline{w}^{(z)} = \min_i \{w_i^{(z)}: i=0, 1, \ldots, N-1\}$.

$\hat{L}^{(z)} = (\bar{w}^{(z)} + \underline{w}^{(z)})/2$.

<u>THEOREM 1</u> For any positive integer z,

(a) $|L - \hat{L}^{(z)}| \leq (\bar{w}^{(z)} - \underline{w}^{(z)})/2$,

(b) $\lim_{z\to\infty} \hat{L}^{(z)} = L$.

Proof. (a) $L = VF = VA^{z}F = \Sigma v_i w_i^{(z)}$

Since

$$w_i^{(z)} \le \bar{w}^{(z)};$$

therefore,

$$\Sigma v_i w_i^{(z)} \le \Sigma v_i \bar{w}^{(z)} = \bar{w}^{(z)} \Sigma v_i = \bar{w}^{(z)}.$$

Similarly, since

$$w_i^{(z)} \ge \underline{w}^{(z)};$$

therefore,

$$\Sigma v_i w_i^{(z)} \ge \Sigma v_i \underline{w}^{(z)} = \underline{w}^{(z)} \Sigma v_i = \underline{w}^{(z)}.$$

Thus

$$L - \hat{L}^{(z)} \le \bar{w}^{(z)} - (\bar{w}^{(z)} + \underline{w}^{(z)})/2 = (\bar{w}^{(z)} - \underline{w}^{(z)})/2,$$

and

$$L - \hat{L}^{(z)} \ge \underline{w}^{(z)} - (\bar{w}^{(z)} + \underline{w}^{(z)})/2 = -(\bar{w}^{(z)} - \underline{w}^{(z)})/2.$$

It follows that

$$|L - \hat{L}^{(z)}| \le (\bar{w}^{(z)} - \underline{w}^{(z)})/2.$$

(b) Let $A^z = (p_o^{(z)},\ldots,p_{N-1}^{(z)})^t$, where $p_i^{(z)}$ is the (i+1)th row of A^z. Since we are concerned with systems whose steady state probabilities exist; therefore,

$$\lim_{z\to\infty} p_i^{(z)} = V$$

for all i. It follows

$$\lim_{z\to\infty} w_i^{(z)} = \lim_{z\to\infty} p_i^{(z)}F = VF = L$$

for all i. Let (s_n) be the sequence formed by combining the N sequences $(w_o^{(z)}),\ldots,(w_{N-1}^{(z)})$ according to the ascendant order of z. Clearly,

$$\lim_{n\to\infty} s_n = L.$$

Since $(\bar{w}^{(z)})$ and $(\underline{w}^{(z)})$ are both subsequence of (s_n), it is clear that

$$\lim_{z\to\infty} \bar{w}^{(z)} = \lim_{z\to\infty} \underline{w}^{(z)} = L$$

It follows

$$\begin{aligned}\lim_{z\to\infty} \hat{L}^{(z)} &= \lim_{z\to\infty} (\bar{w}^{(z)} + \underline{w}^{(z)})/2 \\ &= (\lim_{z\to\infty} \bar{w}^{(z)} + \lim_{z\to\infty} \underline{w}^{(z)})/2 \\ &= (L + L)/2 \\ &= L\end{aligned}$$

3.2 ALGORITHMS FOR ESTIMATING THE EXPECTED NUMBER OF CUSTOMERS

The relationship between $\hat{L}$ and L as stated in Theorem 1 can be used to develop iterative algorithms for estimating the expected number of customers from the system's transition matrix. Since such algorithms must terminate after a finite number of iterations certain amount of error will be introduced. Depending upon how the allowable errors are specified, there are two slightly different approaches.

<u>DEFINITION</u>. The absolute error of an estimation is the absolute value of the difference between the estimation and the true value.

When the true value is unknown, the largest absolute error that may occur to the estimation is called the maximal absolute error.

ALGORITHM 1 This algorithm should be used whenever the allowable error of the estimation is specified in terms of maximal absolute error and thus independent of the magnitude of the expected number of customers itself.

Step 1. Determine the allowable maximal absolute error a.

Step 2. Set $z=0$ and $W^{(0)} = (0,\ldots, N-1)^t$.

Step 3. Compute $W^{(z+1)} = AW^{(z)}$.

Step 4. If $(\bar{w}^{(z+1)} - \underline{w}^{(z+1)})/2 \leq a$, go to step 5; otherwise, increase z by 1 then go to step 3.

Step 5. The desired accuracy has been reached. Let $L = \hat{L}^{(z+1)} = (\bar{w}^{(z+1)} + \underline{w}^{(z+1)})/2$. Terminate.

DEFINITION. The relative error of an estimation is the ratio of the absolute error of the estimation to the true value. When the true value is unknown, the largest relative error that may occur to the estimation is called the maximal relative error.

THEOREM 2 Let $\hat{L}^{(n)}$ be the estimation of the expected number of customers obtained from Algorithm 1, using the allowable maximal absolute error a, then

(a) the maximal absolute error of $\hat{L}^{(n)}$ is a,

(b) the maximal relative error of $\hat{L}^{(n)}$ is $a/(\hat{L}^{(n)} - a)$.

Proof. (a) Step 4 of Algorithm 1 implies $(\bar{w}^{(n)} - \underline{w}^{(n)})/2 \leq a$. Thus, by Theorem 1, $|L - L^{(n)}| \leq (\bar{w}^{(n)} - \underline{w}^{(n)})/2 \leq a$. Hence the maximal absolute error of $\hat{L}^{(n)}$ is a.

(b) The maximal relative error occurs when $L = \hat{L}^{(n)} - a$.

$$\begin{aligned}\text{Thus maximal relative error } &= |\hat{L}^{(n)} - L|/L \\ &= |\hat{L}^{(n)} - (\hat{L}^{(n)} - a)|/(\hat{L}^{(n)} - a) \\ &= a/(\hat{L}^{(n)} - a).\end{aligned}$$

<u>ALGORITHM 2</u> This algorithm should be used whenever the allowable error of the estimation is specified in terms of the maximal relative error and thus associate the error of estimation to the magnitude of the expected number of customers.

Step 1. Determine the allowable maximal relative error r.

Step 2. Set z=0 and $W^{(0)} = (0,\ldots,N-1)^t$.

Step 3. Compute $W^{(z+1)} = AW^{(z)}$.

Step 4. If $(\bar{w}^{(z+1)} - \underline{w}^{(z+1)})/2 \leq r\hat{L}^{(z+1)}/(1+r)$, go to step 5; otherwise, increase z by 1 then go to step 3.

Step 5. The desired accuracy has been reached. Let $L = \hat{L}^{(z+1)} = (\bar{w}^{(z+1)} + \underline{w}^{(z+1)})/2$. Terminate.

<u>THEOREM 3</u> Let $\hat{L}^{(n)}$ be the estimation of the expected number of customers obtained from Algorithm 2, using the allowable maximal relative error r, then

(a) the maximal absolute error of $\hat{L}^{(n)}$ is $r\hat{L}^{(n)}/(1+r)$,

(b) the maximal relative error of $\hat{L}^{(n)}$ is r.

Proof. (a) Step 4 of Algorithm 2 implies $(\bar{w}^{(n)} - \underline{w}^{(n)})/2 \le r\hat{L}^{(n)}/(1+r)$. Thus, by Theorem 1, $|L - \hat{L}^{(n)}| \le (\bar{w}^{(n)} - \underline{w}^{(n)})/2 \le r\hat{L}^{(n)}/(1+r)$. Hence the maximal absolute error of $\hat{L}^{(n)}$ is $r\hat{L}^{(n)}/(1+r)$.

(b) The maximal relative error occurs when $L = \hat{L}^{(n)} - r\hat{L}^{(n)}/(1+r)$. Thus the maximal relative error $= |\hat{L}^{(n)} - L|/L$

$$= \frac{\hat{L}^{(n)} - (\hat{L}^{(n)} - \frac{r\hat{L}^{(n)}}{1+r})}{\hat{L}^{(n)} - \frac{r\hat{L}^{(n)}}{1+r}}$$

$$= r.$$

In both algorithms, we have chosen to calculate $A^z F$ by multiplying at each iteration the transition matrix of the system by the column matrix obtained from the previous iteration. This is represented in Step 3 of both algorithms. We have not tried to raise the transition matrix by successively multiplying the resulting matrix by itself for the following reasons:

(a) Most transition matrices of queueing systems contain a large portion of zero entries. Multiplying the original transition matrix by a column matrix allows one to utilize the special structure of the matrix.

(b) Multiplying a square matrix by a column matrix is easier than multiplying a square matrix by itself.

The alternative of successively multiplying the resulting matrix by itself should be considered if the transition matrix contains only a small portion of zero entries and z is very large.

Chapter IV

CONVEXITY OF THE EXPECTED NUMBER OF CUSTOMERS AS A FUNCTION OF THE NUMBER OF SERVERS OR THE MEAN SERVICE RATE

Just as in any mathematical programming problem, the convexity of the objective function of a queueing design problem is a valuable property in terms of optimization. While it seems unlikely that the expected number of customers in a queueing system will be convex with respect to μ and s simultaneously, there do exist classes of queueing models whose expected number of customers is a convex function of μ for fixed s and a discrete convex function of s for fixed μ. This chapter will be devoted to identifying such classes of queueing systems. Up to now, convexity proofs have usually been conducted for each individual system using closed form expressions for measures of effectiveness. Since the majority of such closed form expressions are extremely complex, or not known, few convexity results have been obtained. Such an approach will be avoided. Instead of trying to obtain results for specific systems, we will attempt to obtain results for a group of similar systems based on the common assumptions of each.

4.1 SOME PRELIMINARIES

The following definitions and theorems are essential for our later discussion.

DEFINITION. Given a convex set C in R^n, a function f: $C \to R$ is convex if $x_1, x_2 \in C$ implies $f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta)f(x_2)$ for every $0 \leq \theta \leq 1$.

DEFINITION. Given a set of consecutive integers Z, a function f: $Z \to R$ is discrete convex if $f(n+2) - 2f(n+1) + f(n) \geq 0$ for each set of n, n+1, n+2 $\in$ Z.

THEOREM 1 Let f be a twice continuously differentiable real-valued function on an open convex set C in R^n. Then f is convex on C if and only if its Hessian matrix is positive semidefinite for each $x \in C$.

See reference 20 or reference 27.

THEOREM 2 Let f_i, i = 1,...,k, be convex functions over a convex set C. If $a_i \geq 0$, i=1,...,k. Then the function $f(x) = \sum_{i=1}^{k} a_i f_i(x)$ is convex on C.

See reference 27.

DEFINITION. Let f be a function whose values are real and whose domain D_f is a subset of R^n. Then the set

$$\text{epi } f = \{(x,y): \ y \geq f(x), \ x \in D_f, \ y \in R\}$$

is called the epigraph of f.

THEOREM 3 A function $f: R^n \to R$ is convex if and only if its epigraph is convex.

See reference 20.

DEFINITION. A set $C \subset E^n$ is midpoint convex if $x^1, x^2 \in C$ implies $w = \frac{x^1}{2} + \frac{x^2}{2} \in C$.

THEOREM 4 A closed midpoint convex subset of a Euclidean space is a convex set.

See reference 7.

THEOREM 5 A function $f: R \to R$ is convex if $f(x+2\Delta x) - 2f(x+\Delta x) + f(x) \geq 0$ holds for each increment or decrement Δx of x.

Proof. Let (x_1,y_1), $(x_2,y_2) \in$ epi f. Then $f(x_1) \leq y_1$ and $f(x_2) \leq y_2$. The midpoint is $((x_1 + x_2)/2, \ (y_1 + y_2)/2)$. Since the hypothesis implies $f((x_1 + x_2)/2) \leq (f(x_1) + f(x_2))/2 \leq (y_1 + y_2)/2$, the midpoint is in the epigraph of f. Thus epi f is midpoint convex. Clearly it is closed, so by Theorem 4, epi f is convex. It follows, by Theorem 3, that f is convex.

4.2 QUEUEING MODELS WHOSE EXPECTED NUMBER OF CUSTOMERS IS A CONVEX FUNCTION OF THE NUMBER OF SERVERS OR THE MEAN SERVICE RATE

For a great number of queueing systems, the expected number of customers in the system is a convex function of the mean service rate and a discrete convex function of the number of servers. The following two theorems allow us to identify a large portion of such queueing systems.

THEOREM 6 The expected number of customers in any parallel channel queueing system is a discrete convex function of the number of servers s over the domain D_s, the set of numbers of servers under which the steady state behavior of the corresponding systems exist, if the system has the property that all of the factors other than the waiting time that can affect the expected number of customers will not be affected by the number of customers in the system at any instant.

Proof. Let L(s) be the expected number of customers in the system with s servers. Clearly, L(s) is a discrete convex function of s over D_s if and only if $\{L(s_1+2) - L(s_1+1)\} - \{L(s_1+1) - L(s_1)\} \geq 0$ holds for each s_1, s_1+1, $s_1+2 \in D_s$. In other words, L(s) is a discrete convex function of s if and only if the decrement in the expected number of customers (or equivalently, the expected waiting time per customer) resulted from adding one more server to the s_1-system is at least as much as the decrement resulting from adding the additional server to the (s_1+1)-system. Since the assumption of the theorem implies that the number of customers joining the system is the same regardless of the number of servers employed in the system, we may compare the decrements in total waiting time instead of the expected waiting time or expected number of customers. We observe that a decrement in the total waiting time occurs whenever the additional server and the original servers are all in busy status. When a customer's waiting time is cut short because of the service of the added server, all of the customers in the queue following that customer may also realize a shorter waiting time even though they may not be served directly by the added server. Since none of the relevant factors is allowed to be affected by the number of customers in system at any instant, the net effect of the added server upon the total waiting time is the decrement in total waiting time generated in the two ways mentioned above. The magnitudes of such decrements are comparable in the following ways.

1) For the same expected queue length, the less the number of servers in the system, the more we may expect that the added server and the original servers will all be in a busy status, and thus the larger the decrement will be if the other factors are fixed.
2) For the same number of servers, the longer the queue the more we may expect that the added server and the original servers will all be in a busy status, and thus the larger the decrement will be if the other factors are fixed.
3) For each unit of waiting time saved on a customer directly served by the added server, the longer the queue and the less the number of servers, the greater we may expect that the total indirect saving

of his followers will be, and thus the larger the decrement will be if the other factors are fixed.

Since for any s_1, s_1+1, $s_1+2 \varepsilon D_s$, the expected queue length of the s_1-system is always at least as long as that of (s_1+1)-system, the three possible influences mentioned above suggest consistently that the decrement in total waiting time resulted from adding one more server to the s_1-system is at least as large as the decrement resulted from adding the additional server to the s_1+1 system. We thus conclude that L(s) is convex in s over D_s.

THEOREM 7 The expected number of customers in any parallel channel queueing system is a convex function of the mean service rate μ over D_μ, the set of mean service rates under which the steady state behaviors of the corresponding systems exist, if the model has the property that all of the factors other than the waiting time that can affect the expected number of customers will not be influenced by the number of customers in the system at any instant.

Proof. Let $g(\mu)$ be the expected service time when the service rate is μ. Then $g(\mu) = 1/\mu$ and $g''(\mu) = 2\mu^{-3} > 0$. Thus by Theorem 1, the expected service time is a convex function of the mean service rate. Let $f(\mu)$ be the expected waiting time a customer spends in the queue waiting for service when the service rate is μ. If there is a $\mu_1 \varepsilon D_\mu$ such that $f(\mu_1) = 0$, then $f(\mu) = 0$ for all $\mu \geq \mu_1$; therefore, $f(\mu)$ is convex over $\{\mu: f(\mu) = 0\}$. Now consider $f(\mu)$ over $D'_\mu = \{\mu: \mu \varepsilon D_\mu \text{ and } f(\mu) \neq 0\}$. Clearly $f(\mu)$ is a monotone decreasing function of μ. Since all of the factors that can affect the expected number of customers will not be influenced by the number of customers in the system at any instant, the magnitude of the decrement (increment) in the expected waiting time in queue resulted from increasing (decreasing) service rate from μ to $\mu + \Delta\mu$ is only dependent on the magnitude of the decrement (increment) in the expected service time resulted from such a change and the magnitude of the expected waiting time in the queue when the service rate is μ. The larger the change in the expected service time and the larger the expected waiting time in queue at the service rate μ, the larger the change in the expected waiting time will be when the service rate is changed by $\Delta\mu$. Since the expected waiting time in the queue is larger for smaller μ and since $\{g(\mu+2\Delta\mu) - g(\mu+\Delta\mu)\} - \{g(\mu+\Delta\mu) - g(\mu)\} > 0$ implies the change in expected service time is also larger for smaller μ (notice that $g(\mu)$ is a monotome decreasing convex function), we may conclude that the change in the expected waiting time in the queue is larger for smaller μ. In other words, since $f(\mu)$ is also a monotone decreasing function, the inequality $\{f(\mu+2\Delta\mu) - f(\mu+\Delta\mu)\} - \{f(\mu+\Delta\mu) - f(\mu)\} > 0$ holds for all $\Delta\mu$ such that $\mu+2\Delta\mu$, $\mu+\Delta\mu$, $\mu \in D'_\mu$. Thus by Theorem 5, $f(\mu)$ is convex in μ over D'_μ and hence over D_μ. Now since the expected waiting time in system is the sum of the expected waiting time in queue and the expected service time, by Theorem 2, the expected waiting time in the system is a convex function of μ. It follows that the expected number of customers in the system is a convex function of μ over D_μ.

In Theorem 6 and 7 a sufficient condition for the expected number of customers to be convex in μ and discrete convex in s was presented. The sufficient condition is that the queueing model must possess the property that all of the factors (excluding the waiting time) that can affect the expected number of customers will not be affected by the number of customers in the system at any instant. Since the instantaneous service rate under a priority service discipline is dependent on the queue length of the system at that instant, priority queueing models are not covered by these two theorems. Neither are the finite calling source or finite waiting space type queueing models covered by these theorems since under such queueing models the effective arrival rate will be affected by the number of customers in the system. In spite of these weaknesses Theorem 6 and 7 still allow us to conclude that all of the parallel channel queueing models of the form $(GI/G/s):(GD/\infty/\infty)$ are convex in μ and discrete convex in s.

4.3 EFFECT OF FINITE WAITING SPACE UPON THE CONVEXITY OF THE EXPECTED NUMBER OF CUSTOMERS WITH RESPECT TO μ and s

The theorems stated in the last section do not apply to any queueing model which is based on the assumption of finite waiting space. One of the assumptions of Theorems 6 and 7 is that the system does not have finite waiting space. The following analysis pertains to all finite waiting space systems that satisfy all other assumptions of Theorems 6 and 7.

To investigate the effect of finite waiting space upon the discrete convexity of the expected number of customers, observe that in such a system when a server is added, the size of the decrement in the resulting expected number of customers is dependent on two factors: the expected queue length and the expected number of customers lost. Ignoring the effect of customers lost, the longer the expected queue length, the more the opportunities for the additional server to make a contribution to reducing the expected number of customers and also the larger each contribution will be. Since the expected queue length is longer for smaller number of servers, it is clear that the expected number of customers is convex in s if the effect of the customers lost is ignored. Now consider the effect of the customers lost. Whenever the number of customers in the system is reduced by one because of the contribution of the added server, the effect of such a reduction can not last beyond the arrival of a lost customer. Thus the more customers lost, the shorter the effect of a reduction will last; hence, the smaller the decrement in the expected number of customers will be when a server is added to the system and other factors remain the same. Since the number of customers lost is larger for a smaller number of servers, the effect of customers lost has a tendency to force the expected number of customers to become a discrete concave function of s. When the number of servers is very small, the number of customers lost may be very high and the difference between the number of customers lost under an s_1-system and that of under (s_1+1)-system may be substantial. Therefore, it is possible that the effect of customers lost overrides the effect of queue length and thus results in a smaller decrement in the expected number of customers when a server is added to the s_1-system than in the (s_1+1)-system. Such a situation, if it occurs, must start with s=1 until s is sufficiently large, say s'. For $s_1>s'$ the decrement in the expected number of customers resulting from adding a server to the s_1-system is never smaller than the decrement resulting from adding the server to the (s_1+1)-system. This is so since the expected number of customers lost decreases as the number of server increases and the difference between the expected number of customers lost under an s_1-system and that of the (s_1+1)-system vanishes at a faster rate than the difference between the expected queue lengths of the two systems. Thus the effect of customers lost upon the expected number of customers decreases relative to the effect of queue length as the number of servers increases. Hence once the number of servers is increased to the extent that the expected number of customers is convex in s it will never become concave again. It is thus clear that the expected number of customers for a finite waiting space queueing system will have at most one discrete concave region and that such concave region will always start with s=1.

The effect of finite waiting space upon the convexity of the expected number of customers as a function of service rate can be investigated in the same fashion. Observe that when the service rate is increased from μ to $\mu+\Delta\mu$, the size of decrement in the expected number of customers that results is dependent on the expected number of customers lost, the expected service time, and the expected queue length when the system is operated at μ rate. Ignoring the effect of the customers lost, the same reasoning used in proving Theorem 7 can be used to claim that the expected number of customers is convex in μ. The effect of customers lost, however, has a tendency to force the expected number of customers to become a concave function of μ, especially when μ is small. For the same reason as was described in the last section, such effect decreases relative to the combined effect of service time and queue length as the service rate increases. Hence the expected number of customers for a finite waiting space queueing system has only one concave segment which is located at the left end of the entire curve.

Chapter V

OPTIMIZATION OF A STRING FUNCTION

It was mentioned in Chapter IV that there exist parallel channel queuing models whose expected number of customers is convex in μ for fixed s and discrete convex in s for fixed μ. This fact prompts our special interest in the type of two variable unconstrained nonlinear programming problem whose objective function is a convex function of a continuous variable and a discrete convex function of a discrete variable. In general, this type of function does not guarantee that a local minimum will always be a global minimum. An obvious way of finding the global minimum for this type of function when the domain of the discrete variable is finite is to find the minimum of the function for each fixed value of the discrete variable using a one dimensional search algorithm such as Fibonacci Search and then from these minima select the global minimum. Such an approach, of course, fails to utilize the discrete convexity property of the function and therefore can be improved. This chapter will be devoted to exploiting properties of such two variable functions and to develop an algorithm for minimizing such functions based on these properties.

5.1 PROPERTIES OF STRING FUNCTIONS

This section is concerned with the properties of string functions which are defined below.

DEFINITION. A two variable function is called a string function if one of its variables is continuous and the other is discrete.

DEFINITION. A continuously convex string function is a string function such that for each fixed value of the discrete variable the function is convex with respect to the continuous variable.

DEFINITION. A discrete convex string function is a string function such that for each fixed value of the continuous variable the function is discrete convex with respect to the discrete variable.

DEFINITION. A function that is both a continuous convex string function and a discrete convex string function is called a frame convex string function.

Throughout this chapter the following notation will be used with the specified meanings.

f: a string function.

x: the continuous variable of a string function.

y: the discrete variable of a string function.

D_x: the set of real numbers that the continuous variable of a string function may take.

D_y: the set of consecutive integers that the discrete variable of a string function may take.

D_f: the domain of the string function f.

THEOREM 1 A sufficient condition for a function, g, of a discrete variable to be nonconvex is the existence of any three points $i < j < k$ in the domain of g such that one of the following conditions is satisfied:

(a) $g(i) < g(j)$ and $g(j) \geq g(k)$,

(b) $g(i) \leq g(j)$ and $g(j) > g(k)$.

Proof. (a) Assume g is convex under the given condition, then $\Delta^2 g(n) \geq 0$ for all $n \in D_g$. Consider the values of g at j, j+1, and k:

Case 1: If $j+1=k$ then it is obvious that $g(j) \geq g(j+1) = g(k)$.

Case 2: If $j+1 \neq k$ then $j+1<k$. Since $\Delta^2 g(j) \geq 0$ implies $g(j+2) \geq 2g(j+1) - g(j)$, if $g(j) < g(j+1)$ then $g(j) < g(j+1) < g(j+2)$. Applying the same argument on j+1 and j+2, we obtain the result that $g(j+1) < g(j+2) < g(j+3)$. Thus by repeating this process continuously it can be shown that $g(j) < g(j+1) < \ldots < g(k)$. This is a contradiction to the assumption that $g(j) \geq g(k)$. Hence $g(j) \geq g(j+1)$.

A similar approach can be used to show that $g(j-1) < g(j)$. Thus we have $g(j-1) < g(j) \geq g(j+1)$. This is a contradiction to the assumption that g is discrete convex since $\Delta^2 g(j-1) < 0$. We thus conclude that g is not a discrete convex function.

(b) The second part of the theorem can be proven in the same fashion.

<u>THEOREM 2</u> Let f be any discrete convex string function. For any $i < j < k < m < n \in D_y$, if $f(x_1,j) = f(x_1,m)$ for some $x_1 \in D_x$, then

(a) $f(x_1,i) \geq f(x_1,j) = f(x_1,m)$,

(b) $f(x_1,k) \leq f(x_1,j) = f(x_1,m)$,

(c) $f(x_1,n) \geq f(x_1,j) = f(x_1,m)$.

Proof. (a) If $f(x_1,i) \not\geq f(x_1,j)$ then $f(x_1,i) < f(x_1,j)$. Since $f(x_1,j) = f(x_1,m)$ and $i < j < m$, by theorem 1, f is not a discrete convex function of y. This is a contradiction to our assumption. Thus $f(x_1,i) \geq f(x_1,j)$.

(b) and (c) can be proved in the same way.

<u>THEOREM 3</u> For any discrete convex string function f and $i < j \in D_y$,

(a) If $f(x_1,i) < f(x_1,j)$, then $f(x_1,k) < f(x_1,k+1)$ for any $x_1 \in D_x$, $k, k+1 \in D_y$, and $k > j$.

(b) If $f(x_1,i) > f(x_1,j)$, then $f(x_1,k-1) > f(x_1,k)$ for any $x_1 \in D_x$, $k-1, k \in D_y$, and $k < i$.

Proof. (a) Since $i < j < k$ and $f(x_1,i) < f(x_1,j)$, by Theorem 1, $f(x_1,j) < f(x_1,k)$. Now since $j < k < k+1$ and $f(x_1,j) < f(x_1,k)$, by Theorem 1, $f(x_1,k) < f(x_1,k+1)$.

(b) Since $k < i < j$ and $f(x_1,i) > f(x_1,j)$, by Theorem 1, $f(x_1,k) > f(x_1,i)$. Now since $k-1 < k < i$ and $f(x_1,k) > f(x_1,i)$, by Theorem 1, $f(x_1,k-1) > f(x_1,k)$.

DEFINITION. Let f be a discrete convex string function and $k > j \in D_y$. The positive region of string k with respect to string j, denoted by P_{k-j}, is the set of $x \in D_x$ such that $f(x,k) > f(x,j)$, i.e.,

$$P_{k-j} = \{x: f(x,k) > f(x,j) \text{ and } x \in D_x\}.$$

The negative region of string k with respect to string j, denoted by N_{k-j}, is the set of $x \in D_x$ such that $f(x,k) < f(x,j)$, i.e.,

$$N_{k-j} = \{x: f(x,k) < f(x,j) \text{ and } x \in D_x\}.$$

DEFINITION. Given any i and j strings of a discrete convex string function, where $i < j$, the ignorable region of the k string of the function, denoted by I_k, is defined as:

$$\begin{aligned} I_k \; = \; & P_{j-i} \text{ if } k = j; \text{ or} \\ & N_{j-i} \text{ if } k = i; \text{ or} \\ & \{x: f(x,k) \geq f(x,k-1) \text{ and } x \in D_x\} \text{ if } k > j; \text{ or} \\ & \{x: f(x,k) \geq f(x,k+1) \text{ and } x \in D_x\} \text{ if } k < i. \end{aligned}$$

If $i=j-1$, then the search region of the k string, denoted by $S_{(k;i,j)}$ is defined as:

$$\begin{aligned} S_{(k;i,j)} \; = \; & \{(x,k): x \in D_x - I_k \text{ and, if } k+1 \in D_y,\ f(x,k+1) \geq f(x,k)\} \\ & \text{if } k \geq j; \text{ or} \\ & \{(x,k): x \in D_x - I_k \text{ and, if } k-1 \in D_y,\ f(x,k-1) \geq f(x,k)\} \\ & \text{if } k \leq i. \end{aligned}$$

<u>THEOREM 4</u> For any discrete convex string function and $i < j \in D_y$,

(a) $P_{j-i} = I_j \subseteq I_{j+1} \subseteq I_{j+2} \subseteq \ldots \subseteq I_{j+n} \ldots,$

(b) $N_{j-i} = I_i \subseteq I_{i-1} \subseteq I_{i-2} \subseteq \ldots \subseteq I_{i-n} \ldots.$

Proof. (a) $P_{j-i} = I_j$ follows directly from definition. Next, for any $x_1 \in I_j$, $f(x_1,i) < f(x_1,j)$. Thus by Theorem 1, $f(x_1,j) < f(x_1,j+1)$. Hence $x_1 \in I_{j+1}$ and $I_j \subseteq I_{j+1}$. Now assume $I_j \subseteq I_{j+1} \subseteq \ldots \subseteq I_{j+n-1}$ holds. For any $x_2 \in I_{j+n-1}$, $f(x_2,j+n-2) \leq f(x_2,j+n-1)$. Thus by Theorem 1, $f(x_2,j+n-1) \leq f(x_2,j+n)$. Hence $x_2 \in I_{j+n}$ and $I_{j+n-1} \subseteq I_{j+n}$. It follows $P_{j-i} = I_j \subseteq I_{j+1} \subseteq I_{j+2} \subseteq \ldots \subseteq I_{j+n} \ldots.$

(b) can be proved in the same way.

<u>THEOREM 5</u> Let $f(x^*,y^*)$ be the minimum of $f(x,y)$ and $i = j-1$. If $(x^*,y^*) \notin \bigcup_n S_{(n;i,j)}$, then there exists at least a point $(x'',y'') \in \bigcup_n S_{(n;i,j)}$ such that $f(x'',y'') = f(x^*,y^*)$.

Proof. If $(x^*,y^*) \notin \bigcup_k S_{(k;i,j)}$, then $x^* \in I_{y^*}$. Under such a situation, $y^* \neq j$. For if $y^* = j$ then $I_{y^*} = I_j = P_{j-i}$ and thus $f(x^*,y^*) = f(x^*,j) > f(x^*,i)$, a contradiction to the fact that $f(x^*,y^*)$ is the minimum. Similarly, $y^* \neq i$. Now assume $y^* > j$. Since $x^* \in I_{y^*}$ and $x^* \notin I_j$, there exists a k, $j \leq k < y^*$, such that $x^* \notin I_k$ and $x^* \in I_{k+1}$. Now since $x^* \in I_{k+1}$ and $I_{k+1} \subseteq I_{k+2} \subseteq \ldots \subseteq I_{y^*}$, $f(x^*,k) \leq f(x^*,y^*)$. Thus $f(x^*,k) = f(x^*,y^*)$ for $f(x^*,y^*)$ is a minimum. Let $(x'',y'') = (x^*,k)$, then $(x'',y'') \in s_{(y'';i,j)}$ and $f(x'',y'') = f(x^*,y^*)$. Thus the theorem holds for $y^* > j$. Similarly, we may prove the theorem holds for $y^* < i$. Hence the theorem holds.

Theorem 4 describes the relationship among the ignorable regions of a convex string function and thus facilitates the determination of the search regions. Theorem 5 implies that in searching the global minimum for a convex string function, one needs only search the search regions.

<u>THEOREM 6</u> Let f be a frame convex string function with domain $D_f = D_x \times D_y$. If for each pair of i, j ε D_y, the value of $f(x,i) - f(x,j)$ as x is varied does not change sign within the entire region of D_x, then any local minimum of $f(x,y)$ is equal to the global minimum of $f(x,y)$.

Proof. Let (x^*,y^*) be a global minimizing point and (x_1,y_1) be any local minimizing point. We want to prove $f(x^*,y^*) = f(x_1,y_1)$.

Case 1: If $y^* = y_1$ then $f(x,y^*)$ and $f(x,y_1)$ represent the same function which is a single variable convex function of x. Thus $f(x^*,y^*) = f(x_1,y_1)$.

Case 2: If $y^* \neq y_1$ and $f(x_1,y_1) \neq f(x^*,y^*)$, then $f(x^*,y^*) < f(x^*,y_1)$ since $f(x^*,y_1) \geq f(x_1,y_1)$. Thus by the condition of the theorem it follows $f(x_1,y^*) < f(x_1,y_1)$. This contradicts our assumption that (x_1,y_1) is a local minimizing point since f is discrete convex in y for all fixed x. Thus we conclude that $f(x^*,y^*) = f(x_1,y_1)$.

Since the above 2 cases exhaust all of the possibilities, we conclude that $f(x_1,y_1) = f(x^*,y^*)$.

The above theorem suggests that the global minimizing point (x^*,y^*) of a frame convex string function that satisfies the assumption stated in the theorem can be obtained as follows: For any $x_1 \in D_x$ find y^* that minimizes $f(x_1,y)$, then find x^* that minimizes $f(x,y^*)$. This process is relatively simple. The strict requirements on the function, however, limit this process to very few actual applications. In the next section an algorithm which has a wider application will be developed based on Theorem 4 and 5.

5.2 ALGORITHM FOR MINIMIZING A DISCRETE CONVEX STRING FUNCTION

For those discrete convex string functions whose ignorable regions are easy to determine, the following algorithm based on Theorem 4 and 5 may be used to obtain their global minima $f(x^*,y^*)$.

Step 0. Set GM (the global minimum) equal to ∞.

Step 1. Choose a number $y' \in D_y$ which is believed to be close to y^*. $y' - 1$ must be in D_y.

Step 2. Determine the ignorable regions $I_{y'}$ and $I_{y'-1}$, where

$I_{y'} = \{x: f(x,y') - f(x,y'-1) > 0\}$,

$I_{y'-1} = \{x: f(x,y') - f(x,y'-1) < 0\}$.

Step 3. Starting with $i=y'$, carry out the following iterative process:

a. If $I_i = D_x$, go to step 4.

b. If $i+1 \in D_y$, let $A = \{x: x \in D_x - I_i$ and $f(x,i+1) - f(x,i) \geq 0\}$; otherwise, let $A = \{x: x \in D_x - I_i\}$.

Find $f(x^*,i) = \min_{x \in A} f(x,i)$.

c. If $f(x^*,i) < GM$, let $GM = f(x^*,i)$.

d. If $i+1 \varepsilon D_y$, let $I_{i+1} = I_i \cup A$, increase i by 1, then go to step 3a; otherwise, go to step 4.

Step 4. Starting with $i = y'-1$ carry out the following iterative process:

a. If $I_i = D_x$, go to step 5.

b. If $i-1 \varepsilon D_y$, let $A = \{x: x \varepsilon D_x - I_i$ and $f(x,i-1) - f(x,i) \geq 0\}$; otherwise, let $A = \{x: x \varepsilon D_x - I_i\}$.

Find $f(x^*,i) = \underset{x \varepsilon A}{\mathrm{Min}}\, f(x,i)$.

c. If $f(x^*,i) < GM$, let $GM = f(x^*,i)$.

d. If $i-1 \varepsilon D_y$, let $I_{i-1} = I_i \cup A$, decrease i by 1, then go to step 4a; otherwise, go to step 5.

Step 5. Terminate the process. The global minimum equal to GM.

5.3 FINITENESS OF THE ALGORITHM

A discrete convex string function must also satisfy the following two conditions in order to assure that its global minimum can be located in a finite number of iterations using the algorithm described in the last section:

(a) Either the discrete variable is bounded above or there is a $y_1 \varepsilon D_y$ such that $f(x,y_1+1) > f(x,y_1)$ holds for all $x \varepsilon D_x$.

(b) Either the discrete variable is bounded below or there is a $y_2 \varepsilon D_y$ such that $f(x,y_2-1) > f(x,y_2)$ holds for all $x \varepsilon D_x$.

Condition (a) guarantees that the algorithm will advance from step 3 to step 4 in a finite number of iterations. This is obvious if y is bounded above. If y is not bounded above, then there exist a $y_1 \in D_y$ such that $f(x,y_1+1) > f(x,y_1)$ for all $x \in D_x$. By Theorem 3, $f(x,y+1) > f(x,y)$ holds for all $x \in D_x$ and $y \geq y_1$. Hence there exists a $y_o > y_1$ such that $y_o \geq y'$ and $f(x,y_o+1) > f(x,y_o)$ for all $x \in D_x$. If the algorithm has already advanced from step 3 to step 4 when i is still less than or equal to y_o, then clearly such advancing has been achieved in finite iterations. If such advancing has not yet achieved at the time when i has been increased to y_o, then $I_i \subset D_x$ and $A = D_x - I_i$. Thus $I_{i+1} = I_i \cup A = D_x$, implying that the algorithm will advance from step 3a to step 4 at next iteration, i.e., in finite iterations. A similar argument can be used to show that condition (b) guarantees the algorithm to advance from step 4 to step 5 in finite iterations. It is clear, therefore, that conditions (a) and (b) are sufficient for the algorithm to converge in a finite number of iterations.

5.4 ILLUSTRATION OF THE ALGORITHM

As a demonstration, we now solve the discrete convex string function shown in Figure 1 of page 38, using the algorithm developed in this chapter. In this example D_x is assumed to be the set of real numbers between 10 and 50 inclusively and D_y the set of positive integers.

Step 0. $GM = \infty$

Step 1. Choose $y' = 3$.

Step 2. $I_3 = \{x: 30 < x \leq 50\}$;

$I_2 = \{x: 10 \leq x < 30\}$.

Step 3. $i = 3$.

a. $I_3 \neq D_x$, continue.

b. $4 \in D_y$, so $A = \{x: 15 \leq x \leq 30\}$; $f(x^*,3) = 15$.

c. GM = 15.

d. $4 \in D_y$, so $I_4 = \{x: 15 \leq x \leq 50\}$; $i = 4$.

a. $I_4 \neq D_x$, continue.

b. $5 \in D_y$, so $A = \{x: 10 \leq x < 15\}$; $f(x^*,4) = 20$.

c. GM = 15.

d. $5 \in D_y$, so $I_5 = \{x: 10 \leq x \leq 50\}$; $i = 5$.

a. $I_5 = D_x$, go to step 4.

Step 4. $i = 2$.

a. $I_2 \neq D_x$, continue.

b. $1 \in D_y$, so $A = \{x: 30 \leq x \leq 50\}$; $f(x^*,1) = 10$.

c. GM = 10.

d. $1 \in D_y$, so $I_1 = \{x: 10 \leq x \leq 50\}$; $i = 0$.

Step 5. GM = 10.

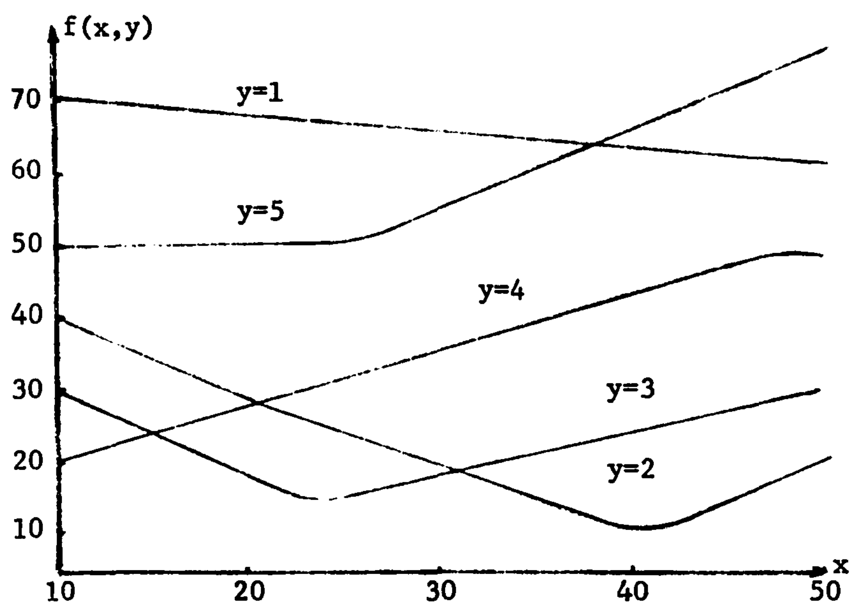

Figure 1

Chapter VI

OPTIMIZATION OF A PARALLEL CHANNEL QUEUING SYSTEM

A design problem is concerned with a single choice of queuing system given a set of initial conditions. Formally, the problem is to

$$\text{minimize } X_o = f(X) + g[P(X)]$$
$$\text{subject to } \quad X \varepsilon \psi$$

where ψ is the set of allowable vectors of values of the design parameters such that if $X\varepsilon\psi$ then $P(X)$, the steady state probability vector of the corresponding system, exists.

Consider the design problem pertaining to a Markovian type parallel channel queuing system. The design parameters can be any combination of the following three components: the arrival rate λ, the service rate μ, and the number of servers s. Here X has six possibilities, i.e., (λ), (μ), (s), (λ,μ), (λ,s), (μ,s), (λ,μ,s). Regardless which of these six possible vectors X represents, if $g[P(X)]$ is a function of the expected number of customers or the expected waiting time, then the algorithms presented in Chapter 3 can be used to estimate the value of $g[P(X)]$ and hence the value of X_o even though the closed form expression of $g[P(X)]$

is not available. For this type of design problem the solution is obtainable, at least theoretically, by total enumeration as long as the transition matrix is available. It should be noted, however, that solving design problems with total enumeration techniques often requires a considerable amount of computer time if the problem is large or if it contains a continuous variable. Part of the computer time may be saved by taking advantage of any desirable characteristic of the objective function such that methods other than total enumeration can be used for solving the problem. As an illustration of how this can be done, the remainder of this chapter will be devoted to the solution of a two parameter design problem using the knowledge acquired in previous chapters.

Consider the following design problem associated with a $(M/M/s):(FCFS/N/\infty)$ queuing model:

$$\begin{aligned} &\text{Minimize} \quad f(\mu,s) = C_1 s + C_2 \mu + C_3 L(\mu,s) \\ &\text{s.t.} \quad s_1 \le s \le s_n \\ &\qquad\quad \mu_1 \le \mu \le \mu_n \end{aligned} \tag{1}$$

where C_1, C_2, C_3 are cost factors and s, μ, $L(\mu,s)$ are a number of servers, service rate, and the expected number of customers respectively. The service rate is allowed to take any value from the real interval $[\mu_1,\mu_n]$ and the number of servers from the set of consecutive integers $\{s_1,\ldots,s_n\}$. The transition matrix for this queuing model is:

$$\begin{bmatrix}
1-\lambda & \lambda & & & & & & \\
\mu & 1-\lambda-\mu & \lambda & & & & & \\
 & 2\mu & 1-\lambda-2\mu & \lambda & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & r\mu & 1-\lambda-r\mu & \lambda & & \\
 & & & & \ddots & \ddots & \ddots & \\
 & & & & & r\mu & 1-\lambda-r\mu & \lambda \\
 & & & & & & r\mu & 1-r\mu
\end{bmatrix}$$

Since the transition matrix is known, the value of f can be calculated for each combination of μ and s.

To solve problem (1), let $\Delta\mu$ be the tolerance allowed for the service rate. Since the expected number of customers may not be a unimodal function of the service rate, μ should be sufficiently small so that $L(\mu,s)$ can be used to represent $L(\mu',s)$ for every $\mu' \varepsilon (\mu-\Delta\mu, \mu+\Delta\mu)$ in the ordinal sense. As mentioned in Section 4.3, $L(s)$ and $L(\mu)$ both have only one point of inflection and the concave portion is always at the left side of the inflection point. Therefore, the set $\{s_1,\ldots,s_n\}$ can be separated into two subsets $S_1=\{s_1,\ldots,s_i\}$ and $S_2=\{s_i,\ldots,s_n\}$ such that $L(s)$ is discrete convex over S_2 and $L(\mu)$ is convex over $[\mu_1+\Delta\mu, \mu_n]$ for every $s \varepsilon S_2$. Problem (1) is then equivalent to:

$$\text{Minimize } \{f(\mu_1^*, s_1^*);\ f(\mu_2^*, s_2^*)\}$$

where

$$f(\mu_1^*, s_1^*) = \min_{\substack{\mu_1 \le \mu \le \mu_n \\ s \in S_1}} f(\mu, s) = C_1 s + C_2 \mu + C_3 L(\mu, s) \qquad (2)$$

and

$$f(\mu_2^*, s_2^*) = \min_{\substack{\mu_1 \le \mu \le \mu_n \\ s \in S_2}} f(\mu, s) = C_1 s + C_2 \mu + C_3 L(\mu, s) \qquad (3)$$

Problem (2) can be solved by finding the minimum for each $s \in S_1$ and then the global minimum $f(\mu_1^*, s_1^*)$ from these minima. To find the minimum for any $s \in S_1$, one may start with $\mu = \mu_1$ and enumerate $f(\mu, s)$ at an increment of $\Delta\mu$ until μ reaches the region on which $f(\mu, s)$ is convex. Any existing one dimensional search algorithm can now be used to search the remaining region for the minimum after slight modification. The modification is necessary since the estimation of the expected number of customers involves a certain amount of error. Thus there is no way to claim that any two alternatives have the same total costs. In fact, one may know for certain that some alternative has a lower or higher total cost than another alternative has if and only if the absolute value of the difference of the two calculated total costs is at least twice as much as the cost factor C_3 times the maximal absolute error used in obtaining the expected number of customers. Hence additional alternatives (points) have to be evaluated each time the one dimensional search algorithm encounters the situation where it is not possible to determine which of the two alternatives under comparison has a lower or higher cost.

Problem (3) is concerned with a frame convex string function. Hence it can be solved with the algorithm stated in Section 5.2. No further explanation regarding the application of the algorithm is needed except that of concerning the determination of the positive and negative regions of any two strings. The following theorems are needed for this purpose.

THEOREM 1 Let f_1, f_2, g_1, g_2 be functions of x. If $f_1(x)$ and $f_2(x)$ are parallel to each other and $g_1(x_1) - g_2(x_1) \neq g_1(x_2) - g_2(x_2)$ for every $x_1 \neq x_2$, then the two curves defined by $f_1(x) + g_1(x)$ and $f_2(x) + g_2(x)$ intersect at no more than one point.

Proof. Let x' be an intersection point of the two curves. Then $f_1(x') + g_1(x') = f_2(x') + g_2(x')$, or equivalently, $f_1(x') - f_2(x') = g_2(x') - g_1(x')$. Similarly, suppose $x'' \neq x'$ is another intersection point of the two curves, then $f_1(x'') - f_2(x'') = g_2(x'') - g_1(x'')$. But since $f_1(x)$ is parallel to $f_2(x)$, $f_1(x') - f_2(x') = f_1(x'') - f_2(x'')$. Thus $g_1(x') - g_2(x') = g_1(x'') - g_2(x'')$. This contradicts the assumption of the theorem. Therefore, the two curves can have at most one intersection point.

THEOREM 2 Let $f(\mu,s) = C_1 s + C_2\mu + C_3 L(\mu,s)$. If for every $j \neq k$ $L(\mu,j) \neq L(\mu,k)$ holds for all $\mu \in D_u$, then $f(\mu,j)$ and $f(\mu,k)$ intersect at no more than one point.

Proof. Assume $j < k$. Since $L(\mu,j)$ is a strictly decreasing function of μ and $L(\mu,j) - L(\mu,k)$ represent the amount of decrement in the expected number of customers when the number of servers is increased from j to k, an argument similar to that of Section 4.2 allows us to conclude that $L(\mu_1,j) - L(\mu_1,k) \neq L(\mu_2,j) - L(\mu_2,k)$ for every $\mu_1 \neq \mu_2$. Let $g_1(\mu) = C_3L(\mu,j)$ and $g_2(\mu) = C_3L(\mu,k)$. It is clear that $g_2(\mu_1) - g_1(\mu_1) \neq g_2(\mu_2) - g_1(\mu_2)$ for every $\mu_1 \neq \mu_2$. Now let $f_1(\mu) = C_1j + C_2\mu$ and $f_2(\mu) = C_1k + C_2\mu$. Then $f_1(\mu) - f_2(\mu) = C_1(j - k) =$ a constant. Hence $f_1(\mu)$ is parallel to $f_2(\mu)$. But $f_1(\mu) + g_1(\mu) = C_1j + C_2\mu + C_3L(\mu,j) = f(\mu,j)$ and $f_2(\mu) + g_2(\mu) = C_1k + C_2\mu + C_3L(\mu,k) = f(\mu,k)$. Thus, by Theorem 1, $f(\mu,j)$ and $f(\mu,k)$ intersect at no more than one point.

The positive and negative regions of any two strings can now be determined as follows:

Case 1: If $[f(\mu_1,k) - f(\mu_1,j)][f(\mu_n,k) - f(\mu_n,j)] > 0$, then $f(\mu,k)$ and $f(\mu,j)$ have no intersection on $[\mu_1,\mu_n]$. Thus if $f(\mu_1,k) > f(\mu_1,j)$, then $P_{k-j} = [\mu_1,\mu_n]$. Otherwise, $N_{k-j} = [\mu_1,\mu_n]$.

Case 2: If $[f(\mu_1,k) - f(\mu_1,j)][f(\mu_n,k) - f(\mu_n,j)] \leq 0$, then there is an intersection in the interval $[\mu_1,\mu_n]$. Let $\mu_3 \varepsilon (\mu_1,\mu_n)$, then a positive region or negative region can be determined by repeating the same procedure on the subinterval $[\mu_1,\mu_3]$ or $[\mu_3,\mu_n]$.

Appendix A contains a Fortran program wirtten for the purpose of solving problem (1). This program uses the approach described above to locate the optimal solution for the design problem. At each enumeration the arrival rate and the service rate are normalized first before they are used for estimating the expected number of customers. In other words, the program finds a factor c that satisfies the inequality $1.0 > c\lambda + cs\mu > 0.1$ and uses $c\lambda$ and $c\mu$ in place of λ and μ for building the transition matrix of the system. Several examples have been solved on an IBM 370/158 computer using this program. Results are included in Appendix B and summarized in Table 1. Among the ten examples listed in Table 1, the first three differ from each other only in the orders of their transition matrices. Example 3, 4, and 5 are different from each other only in the starting points used for the optimization algorithm. So are Example 7, 8, and 9. Example 5 and 6 are different in their cost factors. The last example is deliberately constructed so that the expected number of the customers is a concave function of the number of servers over the entire allowable region, i.e., S_2 is empty. Results of these examples are consistent with our intuition that the computer time required for solving a design problem varies substantially from one problem to another depending on the number of points enumerated and the time required for each enumeration. Factors that will affect the number of points enumerated are the arrival rate, the service rates, the numbers of servers, the maximum absolute error, the tolerance, the cost factors, and the starting string used for optimization. Whereas the amount of time required for each enumeration is dependent upon the order of the transition matrix, the maximum absolute error and also interestingly upon the arrival rate and the service rate.

Table 1

Example	Arrival Rate	Order Tran. Matrx	Allowable Region: Servers From	Allowable Region: Servers To	Allowable Region: Service Rate From	Allowable Region: Service Rate To	Max Abs. Error	Tolerance	y'	Cost Factor C1	Cost Factor C2	Cost Factor C3	s*	u*	No. of pts Total	No. of pts Enumertd	Time in Sec.
1	0.030	24	1	7	0.03	0.12	0.002	0.003	4	1	120	10	2	0.056	217	47	21.6
2	0.030	16	1	7	0.03	0.12	0.002	0.003	4	1	120	10	2	0.056	217	47	10.0
3	0.030	8	1	7	0.03	0.12	0.002	0.003	4	1	120	10	2	0.056	217	47	3.0
4	0.030	8	1	7	0.03	0.12	0.002	0.003	2	1	120	10	2	0.056	217	47	3.3
5	0.030	8	1	7	0.03	0.12	0.002	0.003	7	1	120	10	2	0.056	217	53	3.3
6	0.030	8	1	7	0.03	0.12	0.002	0.003	4	15	120	300	2	0.120	217	32	1.6
7	0.020	15	1	14	0.01	0.06	0.002	0.002	3	3	100	150	2	0.060	364	80	15.0
8	0.020	15	1	14	0.01	0.06	0.002	0.002	13	3	100	150	2	0.060	364	96	20.1
9	0.020	15	1	14	0.01	0.06	0.002	0.002	7	3	100	150	2	0.060	364	84	16.5
10	0.200	15	3	10	0.01	0.05	0.004	0.002	-	3	4	6	6	0.050	168	160	31.8

Chapter VII

SUMMARY AND FURTHER RESEARCH

One of the promising approaches for designing Markovian type parallel channel queueing systems is the approach that estimates the effectiveness of the system directly from its transition matrix. For those design problems whose measures of effectiveness can be estimated from their transition matrices, the optimal system is determinable, at least theoretically, by total enumeration. One must, however, try to take advantage of every desirable characteristic of the objective function such that a more efficient method can be used to locate the optimal solution. Following this idea, three algorithms have been developed. Two of these algorithms are for the estimation of the expected number of customers of a system from its transition matrix. The third algorithm is for the optimization of a discrete convex string function. The first two algorithms allow one to set up and solve for optimal solutions in those design problems which contain only the expected number of customers or the expected waiting time as measures of effectiveness. The third algorithm and the results of the investigation on the characteristics of the expected number of customers provide us with a more realistic approach for optimizing queueing situations in terms of service rate and number of servers regardless of the inavailability of closed form expressions for appropriate measures of effectiveness.

This dissertation, however, has not exhausted every aspect of the subject. Much more work must be done. Such work includes the development of an algorithm for estimating the expected number of lost customers of a system from its transition matrix; the investigation of the character of the expected number of customers in terms of the arrival rate; the investigation of the effect of a finite calling source or priority discipline upon the convexity of the expected number of customers; and the extension of the discrete convex string function minimization algorithm to problems of more than two variables.

Appendix A

COMPUTER PROGRAM FOR SOLVING DESIGN PROBLEMS

This appendix contains a Fortran program for solving Problem (1) of Chapter VI. The method used in the program is that described in Chapter VI.

Input to this program are the parameter cards. Each of these input cards contains the following information:

Card Column	Format	Contents
1 - 5	F5.5	arrival rate
6 - 10	F5.5	the smallest service rate allowed
11 - 15	F5.5	the largest service rate allowed
16 - 20	F5.5	tolerance allowed for the service rate
21 - 25	F5.5	maximal absolute error in obtaining $L(\mu,s)$
26 - 30	F5.0	cost per unit of server
31 - 35	F5.0	cost per unit of service rate
36 - 40	F5.0	cost per unit of $L(\mu,s)$
41 - 43	I3	order of the transition matrix
44 - 46	I3	the smallest number of servers allowed
47 - 49	I3	the largest number of servers allowed
50 - 52	I3	the guessed optimal number of servers

```
C     THIS PROGRAM SOLVES PROBLEM (1) OF CHAPTER VI USING THE
C     ALGORITHMS DEVELOPED IN CHAPTER III & V.  INPUT TO THE
C     PROGRAM CONSISTS OF ARRIVAL RATE, THE LOWER AND UPPER
C     BOUNDS OF THE SERVICE RATE , TOLERANCE, MAXIMAL ABSOLUTE
C     ERROR, COST FACTORS FOR SERVERS, SERVICE RATE, AND THE
C     EXPECTED NO. OF CUSTOMERS, THE ORDER OF THE TRANSITION
C     MATRIX, THE LOWER AND UPPER BOUNDS OF THE NO. OF
C     SERVERS, AND THE STARTING POINT  FOR THE OPTMIZATION.
C     OUTPUT OF THE PROGRAM  CONTENTS  THE OPTIMAL SOLUTION
C     AND ALL OF THE POINTS WHICH ARE ENUMERATED IN ORDER TO
C     DETERMINE THE SOLUTION.
      REAL LAM
      INTEGER SL,SH,Y,YM1,SUB,SLP1,SLP2,SHM1
    1 READ(5,5010,END=100) LAM,UL,UR,DEL,ACC,C1,C2,C3,NORD,SL,
     1SH,Y
 5010 FORMAT(5F5.5,3F5.0,4I3)
      CALL SETIME
      PRINT 6001
 6001 FORMAT('1'//////)
      LN=11
      PRINT 6000,LAM,UL,UR,DEL,SL,SH,NORD,ACC,C1,C2,C3
 6000 FORMAT(' ARRIVAL RATE=',F8.5/
     1' SERVICE RATES   FROM',F6.3,' TO',F6.3,' WITH TOLERAN',
     2'CE=',F6.4/ ' NUMBER OF SERVERS    FROM',I3,' TO',I3/
     3' ORDER OF TRANSITION MATRIX =',I3/
     4' MAX ABSOLUTE ERROR OF THE ESTIMATION OF L =',F7.4/
     5' COST FACTORS   C1=',F8.2,'   C2=',F8.2,'   C3=',F8.2/
     6//' NO.  OF   SERVICE     TOTAL COST     EXPECTED        Z'
     7/' SERVERS    RATE',21X,'CUSTOMERS'/)
      IF(SH.LT.NORD.AND.Y.LE.SH) GO TO 2
      PRINT 6002
 6002 FORMAT(' INPUT ERROR')
      STOP
    2 BNDRY=2.0*ACC*C3
      VALUB=9999999.0
      SLP1=SL+1
C     IF S HAS 1 OR 2 ALTERNATIVES ONLY, DO NOT USE ALGORITHM
      IF(SH-SL-1)104,108,110
  104 CALL EXPQUE(LAM,UL,SH,NORD,ACC,F2,E2,C1,C2,C3,LN)
      GO TO 160
  108 CALL EXPQUE(LAM,UL,SH,NORD,ACC,F2,E2,C1,C2,C3,LN)
      CALL EXPQUE(LAM,UL,SL,NORD,ACC,F1,E1,C1,C2,C3,LN)
      GO TO 150
C     CHECK THE CONVEXITY OF L(S)
  110 CALL EXPQUE(LAM,UL,SL,NORD, ACC,F1,E1,C1,C2,C3,LN)
      CALL EXPQUE(LAM,UL,SLP1,NORD,ACC,F2,E2,C1,C2,C3,LN)
      ISL=SL+2
      DO 140 I=ISL,SH
      SLP2=SL+2
      CALL EXPQUE(LAM,UL,SLP2,NORD, ACC,F3,E3,C1,C2,C3,LN)
      IF(E1+E3-2.0*E2.GT.4.0*ACC) GO TO 170
C     L(S) IS NOT CONVEX ON SL,SL+1,SL+2
```

```
      CALL ENUMER(LAM,UL,UR,SL,NORD,ACC,C1,C2,C3,SUB,UUB,
     1VALUB,DEL,BNDRY,LN,F1,E1)
      E1=E2
      E2=E3
      F1=F2
      F2=F3
      SL=SL+1
  140 CONTINUE
C     ENUMERATE LAST TWO STRINGS
  150 SHM1=SH-1
      CALL ENUMER(LAM,UL,UR,SHM1,NORD,ACC,C1,C2,C3,SUB,UUB,
     1VALUB,DEL,BNDRY,LN,F1,E1)
  160 CALL ENUMER (LAM,UL,UR,SH,NORD,ACC,C1,C2,C3,SUB,UUB,
     1VALUB,DEL,BNDRY,LN,F2,E2)
      GO TO 80
C     L(S) IS CONVEX - CHECK THE CONVEXITY OF L(U)
  170 HDEL=DEL/2.0
      U2=UL+HDEL
      U3=U2+HDEL
      ISL=SL
      DO 200 I=ISL,SH
      CALL EXPQUE(LAM,U2,SL,NORD,ACC,F,E2,C1,C2,C3,LN)
      CALL EXPQUE(LAM,U3,SL,NORD,ACC,F3,E3,C1,C2,C3,LN)
      IF(E1+E3-2.0*E2.GT.4.0*ACC) GO TO 210
C     L(U) IS NOT CONVEX
  180 CALL ENUMER(LAM,U3,UR,SL,NORD,ACC,C1,C2,C3,SUB,UUB,
     1VALUB,DEL,BNDRY,LN,F3,E3)
      SL=SL+1
C     PROCESS NEXT STRING IF THERE IS ONE
      IF(SL.GT.SH) GO TO 80
      CALL EXPQUE(LAM,UL,SL,NORD,ACC,F,E1,C1,C2,C3,LN)
  200 CONTINUE
  210 IF(SH-ISL.LE.1) GO TO 180
C     INCREASE THE STARTING POINT IF NECESSARY
   10 IF(Y.NE.SL)GOTO 30
      Y=Y+1
      GO TO 10
C     DETERMINE IF THE TWO STARTING STRINGS HAVE INTERSECTIONS
   30 YM1=Y-1
      CALL EXPQUE(LAM,UL,Y,NORD,ACC,FYUL,E,C1,C2,C3,LN)
      CALL EXPQUE(LAM,UL,YM1,NORD,ACC,FYM1UL,E,C1,C2,C3,LN)
      DIFFL=FYUL-FYM1UL
      CALL EXPQUE(LAM,UR,Y,NORD,ACC,FYUR,E,C1,C2,C3,LN)
      CALL EXPQUE(LAM,UR,YM1,NORD,ACC,FYM1UR,E,C1,C2,C3,LN)
      DIFFR=FYUR-FYM1UR
      IAYM1=1
C     IF CAN NOT TELL THAT THERE IS NO INTERSECTION, SEARCH
C     BOTH SIDES OF THE S DOMAIN
      IF(ABS(DIFFL).LT.BNDRY.OR.ABS(DIFFR).LT.BNDRY) GO TO 50
      IF(DIFFL*DIFFR.LT.0.0) GO TO 50
C     NO INTERSECTIONS - SEARCH ONLY ONE SIDE
      IF(DIFFL.GT.0.0) GO TO 60
```

```
      IAYM1=0
C     FIND MINIMUM FROM STRINGS WITH S VALUE HIGHER THAN
C     THE STARTING POINT
   50 U0=UL
      U1=UR
      CALL SEARCH(LAM,U0,U1,DEL,ACC,Y,SH,NORD,FYUL,FYUR,
     1C1,C2,C3,SUB,UUB,VALUB,BNDRY,1,LN)
C     FIND MINIMUM FROM STRINGS WITH S VALUE LOWER THAN
C     THE STARTING POINT
      IF(IAYM1.EQ.0)GOTO 80
   60 CALL SEARCH(LAM,UL,UR,DEL,ACC,YM1,SL,NORD,FYM1UL,FYM1UR,
     1C1,C2,C3,SUB,UUB,VALUB,BNDRY,-1,LN)
   80 IF(LN.LE.43) GO TO 90
      PRINT 6001
C     SOLUTION OBTAINED
   90 PRINT 6010,SUB,UUB,VALUB
 6010 FORMAT(///' ** OPTIMAL  SOLUTION **'//'  NO. OF SERVER',
     1'S=',I6/'  SERVICE RATE=',F8.6/'  TOTAL COST=',E10.3)
      CALL GETIME(KMIN,KSEC,KSECC)
      PRINT 6020, KMIN,KSEC,KSECC
 6020 FORMAT('1COMPUTER TIME ',I3,' MIN',I3,'.',I3,' SEC')
      GO TO 1
  100 STOP
      END
```

```
C     THIS SUBROUTINE DETERMINES THE MINIMAL SOLUTION FOR A
C     SPECIFIED STRING.  IT STARTS THE SEARCH PROCESS WITH
C     THE LEFT MOST OF THE STRING USING  THE TOTAL ENUMERATION
C     METHOD UNTIL IT REACHES THE CONVEX PORTION OF THE
C     STRING.  THE REMAINING OF THE STRING IS THEN SEARCHED
C     WITH A MODIFIED GOLDEN SEARCH METHOD.
      SUBROUTINE ENUMER(LAM,UL,UR,S,N,ACC,C1,C2,C3,SUB,UUB,
     1VALUB,DEL,BNDRY,LN,F1,E1)
      INTEGER S,SUB
      REAL LAM
C     TOTAL ENUMERATION
      NU=(UR-UL)/DEL
      IF(F1.GE.VALUB) GO TO 10
      VALUB=F1
      SUB=S
      UUB=UL
   10 U2=UL+DEL
      CALL EXPQUE (LAM,U2,S,N,ACC,F2,E2,C1,C2,C3,LN)
      IF(F2.GE.VALUB) GO TO 20
      VALUB=F2
      SUB=S
      UUB=U2
   20 DO 100 I=2,NU
      U3=U2+DEL
      CALL EXPQUE (LAM,U3,S,N,ACC,F3,E3,C1,C2,C3,LN)
      IF(F3.GE.VALUB) GO TO 30
      VALUB=F3
      SUB=S
      UUB=U3
C     CHECK FOR CONVEXITY
   30 IF(E1+E3-2.0*E2.LE.4.0*ACC) GO TO 40
      IF(NU-I.GE.4) GO TO 110
   40 E1=E2
      E2=E3
      U2=U3
  100 CONTINUE
      RETURN
C     MODIFIED GOLDEN SEARCH FOR THE CONVEX PORTION OF
C     THE STRING
  110 CALL EXPQUE(LAM,UR,S,N,ACC,F2,E2,C1,C2,C3,LN)
      CALL GOLDEN(LAM,S,N,ACC,C1,C2,C3,SUB,UUB,VALUB,DEL,
     1BNDRY,U3,F3,UR,F2,LN)
      RETURN
      END
```

```
C       THIS SUBROUTINE PERFORMS STEP 3 OR STEP 4 OF THE
C       ALGORITHM DEVELOPED IN CHAPTER V.
        SUBROUTINE SEARCH(LAM,SUL,SUR,DEL,ACC,SBEG,SSTP,NORD,
       1FSUL,FSUR,C1,C2,C3,SUB,UUB,VALUB,BNDRY,INC,LN)
        INTEGER S,SP1,SBEG,SSTP,SUB
        REAL LAM
        S=SBEG
        SP1UL=SUL
        SP1UR=SUR
   10   UTH=(SUR-SUL)/3.0
        SP1=S+INC
        IFG=0
C       DETERMINE WHICH OF THE TWO STEPS IS TO BE DONE
        IF(INC.GT.0)GO TO 20
        IF(SP1.GE.SSTP) GO TO 30
   15   CALL GOLDEN(LAM,S,NORD,ACC,C1,C2,C3,SUB,UUB,VALUB,DEL,
       1BNDRY,SUL,FSUL,SUR,FSUR,LN)
        RETURN
C       ANY MORE STRING
   20   IF(SP1.GT.SSTP) GO TO 15
C       DETERMINE THE IGNORABLE REGION
   30   CALL EXPQUE(LAM,SP1UL,SP1,NORD,ACC,FSP1UL,E,C1,C2,C3,LN)
        CALL EXPQUE(LAM,SP1UR,SP1,NORD,ACC,FSP1UR,E,C1,C2,C3,LN)
        DFFUL=FSP1UL-FSUL
        DFFUR=FSP1UR-FSUR
        IF(UTH.LT.DEL) GO TO 300
        IF(ABS(DFFUL).LT.BNDRY) GO TO 270
        IF(ABS(DFFUR).GE.BNDRY) GO TO 230
        U=SUR-UTH
        IFG=1
C       TO LOCATE THE INTERSECTION - 1ST ATTEMPT
   40   CALL EXPQUE(LAM,U,S,NORD,ACC,FSU,E,C1,C2,C3,LN)
        CALL EXPQUE(LAM,U,SP1,NORD,ACC,FSP1U,E,C1,C2,C3,LN)
        DFFU=FSP1U-FSU
        IF(ABS(DFFU).LT.BNDRY) GO TO 70
        IF(DFFU*DFFUL.LE.0.0) GO TO 170
        IF(DFFU.GT.0.0) GO TO 50
        SUL=U
        FSUL=FSU
        GO TO 70
   50   SP1UL=U
        FSP1UL=FSP1U
   70   IF(IFG.NE.0) GO TO 150
        U=U+UTH
C       TO LOCATE THE INTERSECTION - 2ND ATTEMPT
   80   CALL EXPQUE(LAM,U,S,NORD,ACC,FSU,E,C1,C2,C3,LN)
        CALL EXPQUF(LAM,U,SP1,NORD,ACC,FSP1U,E,C1,C2,C3,LN)
        DFFU=FSP1U-FSU
        IF(ABS(DFFU).LT.BNDRY) GO TO 150
        IF(DFFU*DFFUR.GT.0.0) GO TO 100
        IF(DFFU.GT.0.0) GO TO 90
        SUL=U
```

```
      FSUL=FSU
      GO TO 150
   90 SP1UL=U
      FSP1UL=FSP1U
      GO TO 150
  100 IF(DFFU.GT.0.0) GO TO 110
      SUR=U
      FSUR=FSU
      GO TO 150
  110 SP1UR=U
      FSP1UR=FSP1U
C     PERFORM STEP 3B OR 4B
  150 CALL GOLDEN(LAM,S,NORD,ACC,C1,C2,C3,SUB,UUB,VALUB,DFL,
     1BNDRY,SUL,FSUL,SUR,FSUR,LN)
      SUL=SP1UL
      SUR=SP1UR
C     ADVANCE TO NEXT STRING
  160 S=SP1
      FSUL=FSP1UL
      FSUR=FSP1UR
      GO TO 10
  170 IF(DFFU.GT.0.0) GO TO 180
      SUR=U
      FSUR=FSU
      GO TO 150
  180 SP1UR=U
      FSP1UR=FSP1U
      GO TO 150
  230 IF(DFFUR*DFFUL.GT.0.0) GO TO 240
C     THE TWO STRINGS DO INTERSECT
      U=SUL+UTH
      GO TO 40
C     HAVE NO INTERSECTION POINTS
  240 IF(DFFUR.GT.0.0) GO TO 15
      GO TO 160
  270 IF(ABS(DFFUR).LT.0.0) GO TO 150
      U=SUL+UTH
      GO TO 80
C     WHEN THE REGION OF U IS SMALL
  300 IF((ABS(DFFUL).LT.BNDRY).OR.(ABS(DFFUR).LT.BNDRY)) GO
     1TO 150
      IF(DFFUR*DFFUL.LE.0.0) GO TO 150
      IF(DFFUR.GT.0.0) GO TO 15
      GO TO 150
      END
```

```
C     THIS SUBPROGRAM USES A MODIFIED GOLDEN SEARCH ALGORITHM
C     TO DETERMINE THE MINIMAL SOLUTION FOR A SECTION OF A
C     STRING.  THE MODIFICATION IS CONCERNED WITH THE DEALING
C     OF ERRORS INTRODUCED IN THE ESTIMATION OF THE OBJECTIVE
C     FUNCTION.
      SUBROUTINE GOLDEN(LAM,S,N,ACC,C1,C2,C3,SUB,UUB,VALUB,DEL,
     1BNDRY,ELIN,ELVAL,ERIN,ERVAL,LN)
      REAL LAM
      INTEGER S,SUB
      EL=ELIN
      ER=ERIN
C     GOLDEN SEARCH STEPS
   10 A=EL+(ER-EL)*0.382
      CALL EXPQUE(LAM,A,S,N,ACC,AVAL,E,C1,C2,C3,LN)
      B=ER-(ER-EL)*0.382
      CALL EXPQUE(LAM,B,S,N,ACC,BVAL,E,C1,C2,C3,LN)
   20 IF(ABS(AVAL-BVAL).LT.BNDRY) GO TO 100
      IF(AVAL.GT.BVAL) GO TO 50
C     DROP THE RIGHT END
      ER=B
      ERVAL=BVAL
      AOB=ER-A+EL
      CALL EXPQUE(LA[illegible]E,C1,C2,C3,LN)
      IF(AOB-A)30,1[illegible]
   30 B=A
      BVAL=AVAL
      A=AOB
      AVAL=AOBVAL
      GO TO 80
   40 B=AOB
      BVAL=AOBVAL
      GO TO 80
C     DROP THE LEFT EN[illegible]
   50 EL=A
      ELVAL=AVAL
      AOB=ER-B+A
      CALL EXPQUE(LAM,AOB,S,N,ACC,AOBVAL,E,C1,C2,C3,LN)
      IF(AOB-B) 60,10,70
   60 A=AOB
      AVAL=AOBVAL
      GO TO 80
   70 A=B
      AVAL=BVAL
      B=AOB
      BVAL=AOBVAL
   80 IF((ER-EL).GT.DEL) GO TO 20
C     THE REGION OF UNCERTAINTY IS SMALL ENOUGH
   85 IF(VALUB.LE.BVAL) GO TO 88
      SUB=S
      UUB=B
      VALUB=BVAL
   88 IF(VALUB.LE.AVAL) GO TO 90
```

```
C      THIS SUBPROGRAM USES A MODIFIED GOLDEN SEARCH ALGORITHM
C      TO DETERMINE THE MINIMAL SOLUTION FOR A SECTION OF A
C      STRING.  THE MODIFICATION IS CONCERNED WITH THE DEALING
C      OF ERRORS INTRODUCED IN THE ESTIMATION OF THE OBJECTIVE
C      FUNCTION.
       SUBROUTINE GOLDEN(LAM,S,N,ACC,C1,C2,C3,SUB,UUB,VALUB,DEL,
      1BNDRY,ELIN,ELVAL,ERIN,ERVAL,LN)
       REAL LAM
       INTEGER S,SUB
       EL=ELIN
       ER=ERIN
C      GOLDEN SEARCH STEPS
    10 A=EL+(ER-EL)*0.382
       CALL EXPQUE(LAM,A,S,N,ACC,AVAL,E,C1,C2,C3,LN)
       B=ER-(ER-EL)*0.382
       CALL EXPQUE(LAM,B,S,N,ACC,BVAL,E,C1,C2,C3,LN)
    20 IF(ABS(AVAL-BVAL).LT.BNDRY) GO TO 100
       IF(AVAL.GT.BVAL) GO TO 50
C      DROP THE RIGHT END
       ER=B
       ERVAL=BVAL
       AOB=ER-A+EL
       CALL EXPQUE(LAM,AOB,S,N,ACC,AOBVAL,E,C1,C2,C3,LN)
       IF(AOB-A)30,10,40
    30 B=A
       BVAL=AVAL
       A=AOB
       AVAL=AOBVAL
       GO TO 80
    40 B=AOB
       BVAL=AOBVAL
       GO TO 80
C      DROP THE LEFT END
    50 EL=A
       ELVAL=AVAL
       AOB=ER-B+A
       CALL EXPQUE(LAM,AOB,S,N,ACC,AOBVAL,E,C1,C2,C3,LN)
       IF(AOB-B) 60,10,70
    60 A=AOB
       AVAL=AOBVAL
       GO TO 80
    70 A=B
       AVAL=BVAL
       B=AOB
       BVAL=AOBVAL
    80 IF((ER-EL).GT.DEL) GO TO 20
C      THE REGION OF UNCERTAINTY IS SMALL ENOUGH
    85 IF(VALUB.LE.BVAL) GO TO 88
       SUB=S
       UUB=B
       VALUB=BVAL
    88 IF(VALUB.LE.AVAL) GO TO 90
```

```
      SUB=S
      UUB=A
      VALUB=AVAL
   90 IF(VALUB.LE.ELVAL) GO TO 95
      SUB=S
      UUB=EL
      VALUB=ELVAL
   95 IF(VALUB.LE.ERVAL) RETURN
      SUB=S
      UUB=ER
      VALUB=ERVAL
      RETURN
C     NOT ABLE TO TELL WHICH OF THE TWO POINTS IS LARGER -
C     INTRODUCE THE FIRST AUXILARY POINT
  100 C=(A+B)/2.0
      IF(AVAL.GE.BVAL) GO TO 110
      FIN=BVAL
      GO TO 120
  110 FIN=AVAL
  120 CALL EXPQUE(LAM,C,S,N,ACC,CVAL,E,C1,C2,C3,LN)
      IF(ABS(CVAL-AVAL).LT.BNDRY) GO TO 130
      IF(CVAL.LT.AVAL) GO TO 125
  122 PRINT 6000
 6000 FORMAT(' CONCAVE')
      STOP
  125 EL=A
      ELVAL=AVAL
  130 IF(ABS(CVAL-BVAL).LT.BNDRY) GO TO 150
      IF(CVAL.GE.BVAL) GO TO 122
      ER=B
      ERVAL=BVAL
  140 IF((ER-EL).LE.DEL) GO TO 85
      GO TO 10
  150 IF (ABS(CVAL-AVAL).GE.BNDRY) GO TO 140
      IF(CVAL.GE.VALUB) GO TO 155
      SUB=S
      UUB=C
      VALUB=CVAL
C     FIRST AUXILIARY POINT FAILS TO HELP
  155 IF ((A-EL).GT.(4.0*DEL)) GO TO 180
  160 IF(CVAL.GE.VALUB) GO TO 165
      SUB=S
      UUB=C
      VALUB=CVAL
C     TOTAL ENUMERATION
  165 NCUT=(ER-EL)/DEL
      C=EL
      DO 170 I=1,NCUT
      C=C+DEL
      CALL EXPQUE(LAM,C,S,N,ACC,CVAL,E,C1,C2,C3,LN)
      IF(VALUB.LE.CVAL) GO TO 170
      SUB=S
```

```
      UUB=C
      VALUB=CVAL
  170 CONTINUE
      GO TO 85
C     INTRODUCE THE SECOND AUXILIARY POINT
  180 C=(A+EL)/2.0
      CALL EXPQUE(LAM,C,S,N,ACC,CVAL,E,C1,C2,C3,LN)
      IF(ABS(CVAL-AVAL).LT.BNDRY) GO TO 160
      IF(CVAL.GT.AVAL) GO TO 190
      ER=A
      ERVAL=AVAL
      A=C
      AVAL=CVAL
      GO TO 140
  190 EL=C
      ELVAL=CVAL
      GO TO 140
      END
```

```
C     THIS SUBROUTINE ESTIMATES THE EXPECTED NUMBER OF
C     CUSTOMERS OF A M/M/S QUEUING MODEL USING ALGORITHM 1
C     OF CHAPTER III.  THE TOTAL COST OF THE CORRESPONDING
C     SYSTEM IS ALSO CALCULATED AND PRINTED.
      SUBROUTINE  EXPQUE(LA,MU,S,N,ACC,FO,EQ,C1,C2,C3,LN)
      REAL LAM,LA,MU
      INTEGER S
      DIMENSION WZ(100),WZP1(100)
      FS=S
      FCTNR=1.0
    2 PMAX=(FS*MU+LA)*FCTNR
      IF(PMAX.GE.0.1) GO TO 4
      FCTNR=FCTNR*10.0
      GO TO 2
    4 PMAX=(FS*MU+LA)*FCTNR
      IF(PMAX.LT.1.0) GO TO 6
      FCTNR=FCTNR*0.1
      GO TO 4
    6 LAM=LA*FCTNR
      U=MU*FCTNR
      MIN=1
      MAX=N
    8 NCNT=1
C     TO BUILD W(0)
      DO10I=1,N
      WZ(I)=I-1
   10 CONTINUE
C     TO BUILD W(Z+1) FROM W(Z)
      NM1=N-1
   20 WZP1(1)=(1.0-LAM)*WZ(1)+LAM*WZ(2)
      DO40I=2,NM1
      IF((I-1).GT.S) GO TO 30
      FIM1=I-1
      US=FIM1*U
   30 WZP1(I)=US*WZ(I-1)+(1.0-LAM-US)*WZ(I)+LAM*WZ(I+1)
   40 CONTINUE
      IF(NM1.EQ.S) GO TO 60
   42 WZP1(N)=US*WZ(NM1)+(1.0-US)*WZ(N)
C     CHECK FOR MAX ABSOLUTE ERROR
      VALMIN=WZP1(MIN)
      VALMAX=WZP1(MAX)
   45 EQ=(VALMAX+VALMIN)/2.0
      TOL=(VALMAX-VALMIN)/2.0
      IF(TOL.LT.ACC) GO TO 90
C     RESET FOR ANOTHER ITERATION
      DO 50 I =1,N
      WZ(I)=WZP1(I)
   50 CONTINUE
      NCNT=NCNT+1
      IF(NCNT.LE.20000) GO TO 20
      PRINT 6000
 6000 FORMAT(' TOO MANY ITERATIONS')
```

```
      STOP
   60 US=US+U
      GO TO 42
   90 IMIX=0
C     MAKE SURE MIN- AND MAX- COMPONENT REMAIN THE SAME
      DO100I=1,N
      IF(WZP1(I).LT.VALMIN) GO TO 95
      IF(WZP1(I).LE.VALMAX) GO TO 100
      MAX=I
      VALMAX=WZP1(I)
      IMIX=1
      GO TO 100
   95 MIN=I
      VALMIN=WZP1(I)
      IMIX=1
  100 CONTINUE
      IF(IMIX.NE.0) GO TO 45
C     EVALUATE THE OBJECTIVE FUNCTION
      FO=C1*S+C2*MU+C3*EQ
      IF(LN.GT.48) GO TO 120
  110 PRINT 6020,S,MU,FO,EQ,NCNT
 6020 FORMAT(1X,I5,F11.4,E16.4,F11.4,I9)
      LN=LN+1
      RETURN
  120 LN=0
      PRINT 6030
 6030 FORMAT('1'//////)
      GO TO 110
      END
```

Appendix B

NUMERICAL EXPERIMENTATION WITH THE DESIGN ALGORITHM

This appendix contains ten numerical problems of problem (1) of Chapter VI solved using the program listing in Appendix A. Each of these ten computer outputs contains the descriptions of the design problem, the list of points that are enumerated, and the optimal solution of the problem. These results are summarized in Table 1 on page 45. The IBM 370/158 computer times required for solving these problems are also included in the table.

Example 1

ARRIVAL RATE= 0.03000
SERVICE RATES FROM 0.030 TO 0.120 WITH TOLERANCE=0.0030
NUMBER OF SERVERS FROM 1 TO 7
ORDER OF TRANSITION MATRIX = 24
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS C1= 1.00 C2= 120.00 C3= 10.00

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0300	0.1196E 03	11.4975	1650
2	0.0300	0.1895E 02	1.3352	184
3	0.0300	0.1707E 02	1.0467	814
1	0.0315	0.9690E 02	9.2123	1558
1	0.0330	0.7783E 02	7.2871	1397
4	0.0300	0.1769E 02	1.0085	526
3	0.0300	0.1707E 02	1.0467	814
4	0.1200	0.2092E 02	0.2520	101
3	0.1200	0.1992E 02	0.2520	127
2	0.0300	0.1895E 02	1.3352	184
2	0.1200	0.1896E 02	0.2559	197
3	0.0600	0.1525E 02	0.5049	294
2	0.0600	0.1455E 02	0.5353	509
3	0.0415	0.1536E 02	0.7383	487
3	0.0485	0.1509E 02	0.6269	389
3	0.0529	0.1509E 02	0.5738	346
3	0.0507	0.1508E 02	0.5991	367
3	0.0445	0.1520E 02	0.6865	440
3	0.0475	0.1511E 02	0.6416	402
3	0.0505	0.1508E 02	0.6024	369
3	0.0535	0.1509E 02	0.5678	342
3	0.0565	0.1515E 02	0.5371	318
3	0.0595	0.1523E 02	0.5096	297
1	0.0300	0.1196E 03	11.4975	1650
1	0.1200	0.1875E 02	0.3353	510
2	0.0600	0.1455E 02	0.5353	509
1	0.0600	0.1822E 02	1.0019	186
2	0.0900	0.1625E 02	0.3448	286
1	0.0900	0.1682E 02	0.5019	819
2	0.0644	0.1467E 02	0.4947	458
2	0.0856	0.1591E 02	0.3634	306
2	0.0512	0.1457E 02	0.6423	654
2	0.0431	0.1511E 02	0.7930	879
2	0.0563	0.1451E 02	0.5758	562
2	0.0593	0.1454E 02	0.5420	518
2	0.0578	0.1452E 02	0.5584	539
2	0.0542	0.1452E 02	0.6008	596
2	0.0572	0.1452E 02	0.5647	548

2	0.0602	0.1456E 02	0.5329	506
2	0.0632	0.1464E 02	0.5046	470
1	0.1015	0.1739E 02	0.4218	667
1	0.1085	0.1786E 02	0.3839	597
1	0.0971	0.1714E 02	0.4492	718
1	0.0944	0.1700E 02	0.4679	754
1	0.0927	0.1693E 02	0.4804	778
1	0.0917	0.1688E 02	0.4883	793
1	0.0910	0.1686E 02	0.4936	803

** OPTIMAL SOLUTION **

NO. OF SERVERS= 2
SERVICE RATE=0.056280
TOTAL COST= 0.145E 02

Example 2

ARRIVAL RATE= 0.03000
SERVICE RATES FROM 0.030 TO 0.120 WITH TOLERANCE=0.0030
NUMBER OF SERVERS FROM 1 TO 7
ORDER OF TRANSITION MATRIX = 16
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS C1= 1.00 C2= 120.00 C3= 10.00

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0300	0.7959E 02	7.4991	697
2	0.0300	0.1895E 02	1.3349	132
3	0.0300	0.1707E 02	1.0469	600
1	0.0315	0.6951E 02	6.4731	670
1	0.0330	0.6045E 02	5.5492	629
4	0.0300	0.1769E 02	1.0085	400
3	0.0300	0.1707E 02	1.0469	600
4	0.1200	0.2092E 02	0.2520	81
3	0.1200	0.1992E 02	0.2520	97
2	0.0300	0.1895E 02	1.3349	132
2	0.1200	0.1896E 02	0.2558	145
3	0.0600	0.1525E 02	0.5050	220
2	0.0600	0.1455E 02	0.5353	375
3	0.0415	0.1536E 02	0.7383	362
3	0.0485	0.1509E 02	0.6269	291
3	0.0529	0.1509E 02	0.5738	259
3	0.0507	0.1508E 02	0.5991	274
3	0.0445	0.1520E 02	0.6864	328
3	0.0475	0.1511E 02	0.6416	300
3	0.0505	0.1508E 02	0.6024	276
3	0.0535	0.1509E 02	0.5678	256
3	0.0565	0.1515E 02	0.5371	238
3	0.0595	0.1523E 02	0.5096	223
1	0.0300	0.7959E 02	7.4991	697
1	0.1200	0.1875E 02	0.3353	375
2	0.0600	0.1455E 02	0.5353	375
1	0.0600	0.1822E 02	1.0017	134
2	0.0900	0.1625E 02	0.3448	210
1	0.0900	0.1682E 02	0.5020	604
2	0.0644	0.1467E 02	0.4947	337
2	0.0856	0.1591E 02	0.3634	225
2	0.0512	0.1457E 02	0.6423	481
2	0.0431	0.1511E 02	0.7930	647
2	0.0563	0.1451E 02	0.5757	414
2	0.0593	0.1454E 02	0.5420	381
2	0.0578	0.1452E 02	0.5583	397
2	0.0542	0.1452E 02	0.6008	439
2	0.0572	0.1452E 02	0.5647	403

2	0.0602	0.1456E 02	0.5328	373
2	0.0632	0.1463E 02	0.5046	346
1	0.1015	0.1739E 02	0.4218	491
1	0.1085	0.1786E 02	0.3839	440
1	0.0971	0.1714E 02	0.4492	530
1	0.0944	0.1700E 02	0.4679	556
1	0.0927	0.1693E 02	0.4804	574
1	0.0917	0.1688E 02	0.4883	585
1	0.0910	0.1686E 02	0.4936	592

** OPTIMAL SOLUTION **

NO. OF SERVERS= 2
SERVICE RATE=0.056280
TOTAL COST= 0.145E 02

Example 3

ARRIVAL RATE= 0.03000
SERVICE RATES FROM 0.030 TO 0.120 WITH TOLERANCE=0.0030
NUMBER OF SERVERS FROM 1 TO 7
ORDER OF TRANSITION MATRIX = 8
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS C1= 1.00 C2= 120.00 C3= 10.00

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0300	0.3960E 02	3.4998	158
2	0.0300	0.1855E 02	1.2947	60
3	0.0300	0.1704E 02	1.0442	342
1	0.0315	0.3722E 02	3.2445	154
1	0.0330	0.3501E 02	3.0047	148
4	0.0300	0.1768E 02	1.0077	268
3	0.0300	0.1704E 02	1.0442	342
4	0.1200	0.2092E 02	0.2517	62
3	0.1200	0.1992E 02	0.2519	67
2	0.0300	0.1855E 02	1.2947	60
2	0.1200	0.1896E 02	0.2558	88
3	0.0600	0.1525E 02	0.5048	144
2	0.0600	0.1455E 02	0.5351	218
3	0.0415	0.1535E 02	0.7379	224
3	0.0485	0.1509E 02	0.6267	184
3	0.0529	0.1509E 02	0.5737	166
3	0.0507	0.1508E 02	0.5990	175
3	0.0445	0.1520E 02	0.6862	205
3	0.0475	0.1511E 02	0.6414	189
3	0.0505	0.1508E 02	0.6023	176
3	0.0535	0.1509E 02	0.5678	164
3	0.0565	0.1515F 02	0.5370	154
3	0.0595	0.1523E 02	0.5095	145
1	0.0300	0.3960E 02	3.4998	158
1	0.1200	0.1875E 02	0.3352	216
2	0.0600	0.1455E 02	0.5351	218
1	0.0600	0.1790E 02	0.9703	63
2	0.0900	0.1625E 02	0.3448	126
1	0.0900	0.1681E 02	0.5007	337
2	0.0644	0.1467E 02	0.4946	197
2	0.0856	0.1591E 02	0.3633	135
2	0.0512	0.1456E 02	0.6416	274
2	0.0431	0.1508E 02	0.7904	355
2	0.0563	0.1451E 02	0.5754	239
2	0.0593	0.1454E 02	0.5418	221
2	0.0578	0.1452E 02	0.5580	230
2	0.0542	0.1451E 02	0.6004	252
2	0.0572	0.1451E 02	0.5644	233

2	0.0602	0.1456E 02	0.5326	217
2	0.0632	0.1463E 02	0.5045	202
1	0.1015	0.1739E 02	0.4213	279
1	0.1085	0.1786E 02	0.3836	252
1	0.0971	0.1713E 02	0.4485	299
1	0.0944	0.1700E 02	0.4671	313
1	0.0927	0.1692E 02	0.4794	322
1	0.0917	0.1687E 02	0.4872	328
1	0.0910	0.1685E 02	0.4925	331

** OPTIMAL SOLUTION **

NO. OF SERVERS= 2
SERVICE RATE=0.056280
TOTAL COST= 0.1455 02

Example 4

ARRIVAL RATE= 0.03000
SERVICE RATES FROM 0.030 TO 0.120 WITH TOLERANCE=0.0030
NUMBER OF SERVERS FROM 1 TO 7
ORDER OF TRANSITION MATRIX = 8
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS C1= 1.00 C2= 120.00 C3= 10.00

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0300	0.3960E 02	3.4998	158
2	0.0300	0.1855E 02	1.2947	60
3	0.0300	0.1704E 02	1.0442	342
1	0.0315	0.3722E 02	3.2445	154
1	0.0330	0.3501E 02	3.0047	148
2	0.0300	0.1855E 02	1.2947	60
1	0.0300	0.3960E 02	3.4998	158
2	0.1200	0.1896E 02	0.2558	88
1	0.1200	0.1875E 02	0.3352	216
3	0.0300	0.1704E 02	1.0442	342
3	0.1200	0.1992E 02	0.2519	67
2	0.0600	0.1455E 02	0.5351	218
3	0.0600	0.1525E 02	0.5048	144
2	0.0644	0.1467E 02	0.4946	197
2	0.0856	0.1591E 02	0.3633	135
2	0.0512	0.1456E 02	0.6416	274
2	0.0431	0.1508E 02	0.7904	355
2	0.0563	0.1451E 02	0.5754	239
2	0.0593	0.1454E 02	0.5418	221
2	0.0578	0.1452E 02	0.5580	230
2	0.0542	0.1451E 02	0.6004	252
2	0.0572	0.1451E 02	0.5644	233
2	0.0602	0.1456E 02	0.5326	217
2	0.0632	0.1463E 02	0.5045	202
4	0.0300	0.1768E 02	1.0077	268
4	0.0600	0.1622E 02	0.5020	128
3	0.0415	0.1535E 02	0.7379	224
3	0.0485	0.1509E 02	0.6267	184
3	0.0529	0.1509E 02	0.5737	166
3	0.0507	0.1508E 02	0.5990	175
3	0.0445	0.1520E 02	0.6862	205
3	0.0475	0.1511E 02	0.6414	189
3	0.0505	0.1508E 02	0.6023	176
3	0.0535	0.1509E 02	0.5678	164
3	0.0565	0.1515E 02	0.5370	154
3	0.0595	0.1523E 02	0.5095	145
1	0.0644	0.1729E 02	0.8566	55
1	0.0856	0.1667E 02	0.5395	365

1	0.0988	0.1723E 02	0.4377	291
1	0.0775	0.1659E 02	0.6291	430
1	0.0725	0.1671E 02	0.7011	481
1	0.0806	0.1659E 02	0.5921	403
1	0.0790	0.1659E 02	0.6101	416
1	0.0755	0.1662E 02	0.6565	449
1	0.0785	0.1659E 02	0.6170	421
1	0.0815	0.1660E 02	0.5819	396
1	0.0845	0.1664E 02	0.5505	374

** OPTIMAL SOLUTION **

NO. OF SERVERS= 2
SERVICE RATE=0.056280
TOTAL COST= 0.145E 02

Example 5

```
ARRIVAL RATE= 0.03000
SERVICE RATES   FROM 0.030 TO 0.120 WITH TOLERANCE=0.0030
NUMBER OF SERVERS    FROM  1 TO  7
ORDER OF TRANSITION MATRIX =  8
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS   C1=     1.00    C2=  120.00    C3=    10.00
```

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0300	0.3960E 02	3.4998	158
2	0.0300	0.1855E 02	1.2947	60
3	0.0300	0.1704E 02	1.0442	342
1	0.0315	0.3722E 02	3.2445	154
1	0.0330	0.3501E 02	3.0047	148
7	0.0300	0.2061E 02	1.0013	244
6	0.0300	0.1961E 02	1.0014	245
7	0.1200	0.2392E 02	0.2517	59
6	0.1200	0.2292E 02	0.2518	59
5	0.0300	0.1862E 02	1.0021	250
5	0.1200	0.2192E 02	0.2517	60
4	0.0300	0.1768E 02	1.0077	268
4	0.1200	0.2092E 02	0.2517	62
3	0.0300	0.1704E 02	1.0442	342
3	0.1200	0.1992E 02	0.2519	67
2	0.0300	0.1855E 02	1.2947	60
2	0.1200	0.1896E 02	0.2558	88
3	0.0600	0.1525E 02	0.5048	144
2	0.0600	0.1455E 02	0.5351	218
3	0.0415	0.1535E 02	0.7379	224
3	0.0485	0.1509E 02	0.6267	184
3	0.0529	0.1509E 02	0.5737	166
3	0.0507	0.1508E 02	0.5990	175
3	0.0445	0.1520E 02	0.6862	205
3	0.0475	0.1511E 02	0.6414	189
3	0.0505	0.1508E 02	0.6023	176
3	0.0535	0.1509E 02	0.5678	164
3	0.0565	0.1515E 02	0.5370	154
3	0.0595	0.1523E 02	0.5095	145
1	0.0300	0.3960E 02	3.4998	158
1	0.1200	0.1875E 02	0.3352	216
2	0.0600	0.1455E 02	0.5351	218
1	0.0600	0.1790E 02	0.9703	63
2	0.0900	0.1625E 02	0.3448	126
1	0.0900	0.1681E 02	0.5007	337
2	0.0644	0.1467E 02	0.4946	197
2	0.0856	0.1591E 02	0.3633	135
2	0.0512	0.1456E 02	0.6416	274

2	0.0431	0.1508E 02	0.7904	355
2	0.0563	0.1451E 02	0.5754	239
2	0.0593	0.1454E 02	0.5418	221
2	0.0578	0.1452E 02	0.5580	230
2	0.0542	0.1451E 02	0.6004	252
2	0.0572	0.1451E 02	0.5644	233
2	0.0602	0.1456E 02	0.5326	217
2	0.0632	0.1463E 02	0.5045	202
1	0.1015	0.1739E 02	0.4213	279
1	0.1085	0.1786E 02	0.3836	252
1	0.0971	0.1713E 02	0.4485	299
1	0.0944	0.1700E 02	0.4671	313
1	0.0927	0.1692E 02	0.4794	322
1	0.0917	0.1687E 02	0.4872	328
1	0.0910	0.1685E 02	0.4925	331

** OPTIMAL SOLUTION **

NO. OF SERVERS= 2
SERVICE RATE=0.056280
TOTAL COST= 0.145E 02

Example 6

```
ARRIVAL RATE= 0.03000
SERVICE RATES   FROM 0.030 TO 0.120 WITH TOLERANCE=0.0030
NUMBER OF SERVERS    FROM  1 TO  7
ORDER OF TRANSITION MATRIX =  8
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS   C1=   15.00   C2=  120.00   C3=  300.00
```

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0300	0.1069E 04	3.4998	158
2	0.0300	0.4220E 03	1.2947	60
3	0.0300	0.3618E 03	1.0442	342
1	0.0315	0.9921E 03	3.2445	154
1	0.0330	0.9204E 03	3.0047	148
4	0.0300	0.3659E 03	1.0077	268
3	0.0300	0.3618E 03	1.0442	342
4	0.1200	0.1499E 03	0.2517	62
3	0.1200	0.1350E 03	0.2519	67
2	0.0300	0.4220E 03	1.2947	60
2	0.1200	0.1211E 03	0.2558	88
3	0.0600	0.2036E 03	0.5048	144
2	0.0600	0.1977E 03	0.5351	218
3	0.0415	0.2713E 03	0.7379	224
3	0.0485	0.2388E 03	0.6267	184
3	0.0529	0.2234E 03	0.5737	166
3	0.0556	0.2152E 03	0.5452	157
3	0.0573	0.2106E 03	0.5290	152
3	0.0583	0.2079E 03	0.5196	149
3	0.0590	0.2062E 03	0.5137	147
1	0.0300	0.1069E 04	3.4998	158
1	0.1200	0.1300E 03	0.3352	216
2	0.0644	0.1861E 03	0.4946	197
2	0.0856	0.1493E 03	0.3633	135
2	0.0988	0.1357E 03	0.3128	112
2	0.1069	0.1293E 03	0.2884	101
2	0.1119	0.1259E 03	0.2749	96
2	0.1150	0.1240E 03	0.2673	93
2	0.1169	0.1228E 03	0.2627	91
2	0.1159	0.1234E 03	0.2650	92
2	0.1149	0.1240E 03	0.2675	93
2	0.1179	0.1223E 03	0.2605	90

```
** OPTIMAL  SOLUTION **

 NO. OF SERVERS=      2
 SERVICE RATE=0.120000
 TOTAL COST= 0.121E 03
```

Example 7

ARRIVAL RATE= 0.02000
SERVICE RATES FROM 0.010 TO 0.060 WITH TOLERANCE=0.0020
NUMBER OF SERVERS FROM 1 TO 14
ORDER OF TRANSITION MATRIX = 15
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS C1= 3.00 C2= 100.00 C3= 150.00

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0100	0.1954E 04	12.9980	389
2	0.0100	0.1093E 04	7.2402	853
3	0.0100	0.4362E 03	2.8416	371
1	0.0110	0.1921E 04	12.7772	445
1	0.0120	0.1880E 04	12.5045	508
1	0.0140	0.1765E 04	11.7357	656
1	0.0160	0.1586E 04	10.5446	807
1	0.0180	0.1338E 04	8.8871	908
1	0.0200	0.1055E 04	6.9988	912
1	0.0220	0.7970E 03	5.2785	830
1	0.0600	0.8429E 02	0.5019	82
1	0.0365	0.1883E 03	1.2109	240
1	0.0455	0.1255E 03	0.7866	144
1	0.0510	0.1051E 03	0.6464	113
1	0.0545	0.9581E 02	0.5824	99
1	0.0566	0.9095E 02	0.5486	92
1	0.0579	0.8828E 02	0.5300	88
1	0.0587	0.8664E 02	0.5185	86
1	0.0592	0.8581E 02	0.5126	84
1	0.0595	0.8511E 02	0.5077	83
2	0.0100	0.1093E 04	7.2402	853
2	0.0110	0.8492E 03	5.6140	778
2	0.0120	0.6592E 03	4.3464	668
3	0.0100	0.4362E 03	2.8416	371
2	0.0100	0.1093E 04	7.2402	853
3	0.0600	0.6538E 02	0.3359	197
2	0.0600	0.6372E 02	0.3448	304
4	0.0100	0.3392E 03	2.1743	182
4	0.0600	0.6829E 02	0.3353	164
3	0.0267	0.1267E 03	0.7666	549
4	0.0267	0.1277E 03	0.7537	403
3	0.0291	0.1169E 03	0.6997	485
3	0.0409	0.8714E 02	0.4937	311
3	0.0482	0.7658E 02	0.4184	254
3	0.0527	0.7164E 02	0.3825	229
3	0.0555	0.6901E 02	0.3631	216
3	0.0572	0.6757E 02	0.3523	208
3	0.0583	0.6667E 02	0.3456	204

3	0.0589	0.6622E 02	0.3422	201
3	0.0586	0.6645E 02	0.3440	202
3	0.0592	0.6598E 02	0.3404	200
5	0.0100	0.3222E 03	2.0415	117
5	0.0267	0.1305E 03	0.7521	353
4	0.0156	0.2100E 03	1.3095	78
5	0.0156	0.2102E 03	1.2911	61
4	0.0211	0.1573E 03	0.9545	538
5	0.0211	0.1596E 03	0.9499	455
4	0.0164	0.1998E 03	1.2411	72
4	0.0203	0.1630E 03	0.9934	566
4	0.0227	0.1471E 03	0.8854	489
4	0.0242	0.1389E 03	0.8300	452
4	0.0252	0.1344E 03	0.7989	432
4	0.0257	0.1318E 03	0.7812	420
4	0.0261	0.1301E 03	0.7702	413
6	0.0100	0.3206E 03	2.0106	94
6	0.0211	0.1625E 03	0.9492	422
5	0.0137	0.2367E 03	1.4687	72
6	0.0137	0.2386E 03	1.4618	670
5	0.0142	0.2282E 03	1.4119	68
5	0.0169	0.1951E 03	1.1894	588
5	0.0185	0.1795E 03	1.0845	529
5	0.0195	0.1713E 03	1.0288	498
5	0.0201	0.1666E 03	0.9972	480
5	0.0205	0.1639E 03	0.9788	470
7	0.0100	0.3225E 03	2.0035	85
7	0.0137	0.2415E 03	1.4606	633
6	0.0114	0.2828E 03	1.7578	79
6	0.0123	0.2640E 03	1.6317	72
6	0.0128	0.2537E 03	1.5626	68
6	0.0132	0.2477E 03	1.5229	66
2	0.0291	0.1261E 03	0.7813	86
2	0.0409	0.8840E 02	0.5220	520
2	0.0482	0.7615E 02	0.4355	410
2	0.0527	0.7061E 02	0.3956	362
2	0.0555	0.6771E 02	0.3744	337
2	0.0572	0.6611E 02	0.3626	324
2	0.0583	0.6514E 02	0.3554	315
2	0.0589	0.6463E 02	0.3516	311
2	0.0586	0.6488E 02	0.3535	313
2	0.0592	0.6438E 02	0.3497	309

** OPTIMAL SOLUTION **

NO. OF SERVERS= 2
SERVICE RATE=0.060000
TOTAL COST= 0.637E 02

Example 8

ARRIVAL RATE= 0.02000
SERVICE RATES FROM 0.010 TO 0.060 WITH TOLERANCE=0.0020
NUMBER OF SERVERS FROM 1 TO 14
ORDER OF TRANSITION MATRIX = 15
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS C1= 3.00 C2= 100.00 C3= 150.00

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0100	0.1954E 04	12.9980	389
2	0.0100	0.1093E 04	7.2402	853
3	0.0100	0.4362E 03	2.8416	371
1	0.0110	0.1921E 04	12.7772	445
1	0.0120	0.1880E 04	12.5045	508
1	0.0140	0.1765E 04	11.7357	656
1	0.0160	0.1586E 04	10.5446	807
1	0.0180	0.1338E 04	8.8871	908
1	0.0200	0.1055E 04	6.9988	912
1	0.0220	0.7970E 03	5.2785	830
1	0.0600	0.8429E 02	0.5019	82
1	0.0365	0.1883E 03	1.2109	240
1	0.0455	0.1255E 03	0.7866	144
1	0.0510	0.1051E 03	0.6464	113
1	0.0545	0.9581E 02	0.5824	99
1	0.0566	0.9095E 02	0.5486	92
1	0.0579	0.8828E 02	0.5300	88
1	0.0587	0.8664E 02	0.5185	86
1	0.0592	0.8581E 02	0.5126	84
1	0.0595	0.8511E 02	0.5077	83
2	0.0100	0.1093E 04	7.2402	853
2	0.0110	0.8492E 03	5.[illegible]140	778
2	0.0120	0.6592E 03	4.3464	668
13	0.0100	0.3400E 03	2.0003	811
12	0.0100	0.3370E 03	2.0003	813
13	0.0600	0.9528E 02	0.3352	132
12	0.0600	0.9227E 02	0.3351	133
11	0.0100	0.3340E 03	2.0003	816
11	0.0600	0.8928E 02	0.3352	133
10	0.0100	0.3311E 03	2.0004	822
10	0.0600	0.8627E 02	0.3352	134
9	0.0100	0.3281E 03	2.0005	831
9	0.0600	0.8327E 02	0.3352	135
8	0.0100	0.3251E 03	2.0008	850
8	0.0600	0.8029E 02	0.3352	136
7	0.0100	0.3225E 03	2.0035	85
7	0.0600	0.7728E 02	0.3352	139
6	0.0100	0.3206E 03	2.0106	94

6	0.0600	0.7429E 02	0.3352	143
5	0.0100	0.3222F 03	2.0415	117
5	0.0600	0.7129E 02	0.3353	150
6	0.0267	0.1334E 03	0.7519	331
5	0.0267	0.1305E 03	0.7521	353
6	0.0164	0.2032E 03	1.2236	552
6	0.0203	0.1681E 03	0.9871	440
6	0.0227	0.1525E 03	0.8816	391
6	0.0242	0.1445E 03	0.8272	365
6	0.0252	0.1400E 03	0.7966	351
6	0.0257	0.1374E 03	0.7791	343
6	0.0261	0.1358E 03	0.7682	338
4	0.0100	0.3392E 03	2.1743	182
4	0.0600	0.6829E 02	0.3353	164
5	0.0267	0.1305E 03	0.7521	353
4	0.0267	0.1277E 03	0.7537	403
5	0.0164	0.2005E 03	1.2260	609
5	0.0203	0.1652E 03	0.9879	475
5	0.0227	0.1496E 03	0.8821	419
5	0.0242	0.1416E 03	0.8275	391
5	0.0252	0.1370E 03	0.7969	375
5	0.0257	0.1345E 03	0.7793	366
5	0.0261	0.1329E 03	0.7684	361
3	0.0100	0.4362E 03	2.8416	371
3	0.0600	0.6538E 02	0.3359	197
4	0.0267	0.1277E 03	0.7537	403
3	0.0267	0.1267E 03	0.7666	549
4	0.0164	0.1998E 03	1.2411	72
4	0.0203	0.1630E 03	0.9934	566
4	0.0227	0.1471E 03	0.8854	489
4	0.0242	0.1389E 03	0.8300	452
4	0.0252	0.1344E 03	0.7989	432
4	0.0257	0.1318E 03	0.7812	420
4	0.0261	0.1301E 03	0.7702	413
2	0.0100	0.1093E 04	7.2402	853
2	0.0600	0.6372E 02	0.3448	304
3	0.0267	0.1267E 03	0.7666	549
2	0.0267	0.1399E 03	0.8746	102
3	0.0433	0.8319E 02	0.4657	289
2	0.0433	0.8375E 02	0.4895	477
3	0.0291	0.1169E 03	0.6997	485
3	0.0409	0.8714E 02	0.4937	311
3	0.0482	0.7658E 02	0.4184	254
3	0.0527	0.7164E 02	0.3825	229
3	0.0555	0.6901E 02	0.3631	216
3	0.0572	0.6757E 02	0.3523	208
3	0.0583	0.6667E 02	0.3456	204
3	0.0589	0.6622E 02	0.3422	201
3	0.0586	0.6645E 02	0.3440	202

1	0.0592	0.6598E 02	0.3404	200
2	0.0394	0.9161E 02	0.5445	50
2	0.0473	0.7746E 02	0.4449	421
2	0.0521	0.7125E 02	0.4002	367
2	0.0551	0.6807E 02	0.3770	340
2	0.0570	0.6630E 02	0.3640	325
2	0.0581	0.6528E 02	0.3564	317
2	0.0589	0.6466E 02	0.3518	311
2	0.0593	0.6432E 02	0.3493	309

** OPTIMAL SOLUTION **

NO. OF SERVERS= 2
SERVICE RATE=0.060000
TOTAL COST= 0.637E 02

Example 9

ARRIVAL RATE= 0.02000
SERVICE RATES FROM 0.010 TO 0.060 WITH TOLERANCE=0.0020
NUMBER OF SERVERS FROM 1 TO 14
ORDER OF TRANSITION MATRIX = 15
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0020
COST FACTORS C1= 3.00 C2= 100.00 C3= 150.00

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
1	0.0100	0.1954E 04	12.9980	389
2	0.0100	0.1093E 04	7.2402	853
3	0.0100	0.4362E 03	2.8416	371
1	0.0110	0.1921E 04	12.7772	445
1	0.0120	0.1880E 04	12.5045	508
1	0.0140	0.1765E 04	11.7357	656
1	0.0160	0.1586E 04	10.5446	807
1	0.0180	0.1338E 04	8.8871	908
1	0.0200	0.1055E 04	6.9988	912
1	0.0220	0.7970E 03	5.2785	830
1	0.0600	0.8429E 02	0.5019	82
1	0.0365	0.1883E 03	1.2109	240
1	0.0455	0.1255E 03	0.7866	144
1	0.0510	0.1051E 03	0.6464	113
1	0.0545	0.9581E 02	0.5824	99
1	0.0566	0.9095E 02	0.5486	92
1	0.0579	0.8828E 02	0.5300	88
1	0.0587	0.8664E 02	0.5185	86
1	0.0592	0.8581E 02	0.5126	84
1	0.0595	0.8511E 02	0.5077	83
2	0.0100	0.1093E 04	7.2402	853
2	0.0110	0.8492E 03	5.6140	778
2	0.0120	0.6592E 03	4.3464	668
7	0.0100	0.3225E 03	2.0035	85
6	0.0100	0.3206E 03	2.0106	94
7	0.0600	0.7728E 02	0.3352	139
6	0.0600	0.7429E 02	0.3352	143
5	0.0100	0.3222E 03	2.0415	117
5	0.0600	0.7129E 02	0.3353	150
6	0.0267	0.1334E 03	0.7519	331
5	0.0267	0.1305E 03	0.7521	353
6	0.0164	0.2032E 03	1.2236	552
6	0.0203	0.1681E 03	0.9871	440
6	0.0227	0.1525E 03	0.8816	391
6	0.0242	0.1445E 03	0.8272	365
6	0.0252	0.1400E 03	0.7966	351
6	0.0257	0.1374E 03	0.7791	343
6	0.0261	0.1358E 03	0.7682	338

4	0.0100	0.3392E 03	2.1743	182
4	0.0600	0.6829E 02	0.3353	164
5	0.0267	0.1305E 03	0.7521	353
4	0.0267	0.1277E 03	0.7537	403
5	0.0164	0.2005E 03	1.2260	609
5	0.0203	0.1652E 03	0.9879	475
5	0.0227	0.1496E 03	0.8821	419
5	0.0242	0.1416E 03	0.8275	391
5	0.0252	0.1370E 03	0.7969	375
5	0.0257	0.1345E 03	0.7793	366
5	0.0261	0.1329E 03	0.7684	361
3	0.0100	0.4362E 03	2.8416	371
3	0.0600	0.6538E 02	0.3359	197
4	0.0267	0.1277E 03	0.7537	403
3	0.0267	0.1267E 03	0.7666	549
4	0.0164	0.1998E 03	1.2411	72
4	0.0203	0.1630E 03	0.9934	566
4	0.0227	0.1471E 03	0.8854	489
4	0.0242	0.1389E 03	0.8300	452
4	0.0252	0.1344E 03	0.7989	432
4	0.0257	0.1318E 03	0.7812	420
4	0.0261	0.1301E 03	0.7702	413
2	0.0100	0.1093E 04	7.2402	853
2	0.0600	0.6372E 02	0.3448	304
3	0.0267	0.1267E 03	0.7666	549
2	0.0267	0.1399E 03	0.8746	102
3	0.0433	0.8319E 02	0.4657	289
2	0.0433	0.8375E 02	0.4895	477
3	0.0291	0.1169E 03	0.6997	485
3	0.0409	0.8714E 02	0.4937	311
3	0.0482	0.7658E 02	0.4184	254
3	0.0527	0.7164E 02	0.3825	229
3	0.0555	0.6901E 02	0.3631	216
3	0.0572	0.6757E 02	0.3523	208
3	0.0583	0.6667E 02	0.3456	204
3	0.0589	0.6622E 02	0.3422	201
3	0.0586	0.6645E 02	0.3440	202
3	0.0592	0.6598E 02	0.3404	200
2	0.0394	0.9161E 02	0.5445	50
2	0.0473	0.7746E 02	0.4449	421
2	0.0521	0.7125E 02	0.4002	367
2	0.0551	0.6807E 02	0.3770	340
2	0.0570	0.6630E 02	0.3640	325
2	0.0581	0.6528E 02	0.3564	317
2	0.0589	0.6466E 02	0.3518	311
2	0.0593	0.6432E 02	0.3493	309

```
** OPTIMAL  SOLUTION **

 NO. OF SERVERS=      2
 SERVICE RATE=0.060000
 TOTAL COST= 0.637E 02
```

Example 10

ARRIVAL RATE= 0.20000
SERVICE RATES FROM 0.010 TO 0.050 WITH TOLERANCE=0.0020
NUMBER OF SERVERS FROM 3 TO 10
ORDER OF TRANSITION MATRIX = 15
MAX ABSOLUTE ERROR OF THE ESTIMATION OF L = 0.0040
COST FACTORS C1= 3.00 C2= 4.00 C3= 6.00

NO. OF SERVERS	SERVICE RATE	TOTAL COST	EXPECTED CUSTOMERS	Z
3	0.0100	0.9196E 02	13.8196	160
4	0.0100	0.9452E 02	13.7459	173
5	0.0100	0.9702E 02	13.6627	188
3	0.0120	0.9171E 02	13.7765	169
3	0.0140	0.9144E 02	13.7302	179
3	0.0160	0.9115E 02	13.6802	190
3	0.0180	0.9083E 02	13.6261	202
3	0.0200	0.9048E 02	13.5673	215
3	0.0220	0.9011E 02	13.5032	229
3	0.0240	0.8970E 02	13.4333	245
3	0.0260	0.8924E 02	13.3564	262
3	0.0280	0.8874E 02	13.2717	281
3	0.0300	0.8819E 02	13.1778	302
3	0.0320	0.8757E 02	13.0732	324
3	0.0340	0.8687E 02	12.9562	349
3	0.0360	0.8609E 02	12.8247	375
3	0.0380	0.8521E 02	12.6761	403
3	0.0400	0.8421E 02	12.5079	434
3	0.0420	0.8307E 02	12.3166	465
3	0.0440	0.8177E 02	12.0994	498
3	0.0460	0.8030E 02	11.8526	532
3	0.0480	0.7863E 02	11.5735	566
3	0.0500	0.7676E 02	11.2595	599
6	0.0100	0.9944E 02	13.5674	202
4	0.0120	0.9413E 02	13.6801	187
4	0.0140	0.9370E 02	13.6071	203
4	0.0160	0.9322E 02	13.5253	220
4	0.0180	0.9267E 02	13.4333	239
4	0.0200	0.9206E 02	13.3292	261
4	0.0220	0.9135E 02	13.2104	286
4	0.0240	0.9054E 02	13.0736	313
4	0.0260	0.8959E 02	12.9151	344
4	0.0280	0.8849E 02	12.7299	377
4	0.0300	0.8719E 02	12.5124	413
4	0.0320	0.8567E 02	12.2564	451
4	0.0340	0.8387E 02	11.9553	490
4	0.0360	0.8176E 02	11.6032	528
4	0.0380	0.7933E 02	11.1960	566

4	0.0400	0.7656E 02	10.7327	598
4	0.0420	0.7347E 02	10.2168	626
4	0.0440	0.7012E 02	9.6567	645
4	0.0460	0.6658E 02	9.0656	656
4	0.0480	0.6295E 02	8.4599	658
4	0.0500	0.5934E 02	7.8570	650
5	0.0100	0.1018E 03	13.4576	216
5	0.0120	0.9645E 02	13.5674	207
5	0.0140	0.9580E 02	13.4575	229
5	0.0160	0.9504E 02	13.3293	254
5	0.0180	0.9414E 02	13.1784	283
5	0.0200	0.9307E 02	12.9984	315
5	0.0220	0.9178E 02	12.7817	351
5	0.0240	0.9021E 02	12.5191	391
5	0.0260	0.8830E 02	12.2001	432
5	0.0280	0.8600E 02	11.8149	474
5	0.0300	0.8326E 02	11.3561	513
5	0.0320	0.8006E 02	10.8222	547
5	0.0340	0.7646E 02	10.2199	573
5	0.0360	0.7253E 02	9.5651	589
5	0.0380	0.6844E 02	8.8815	594
5	0.0400	0.6434E 02	8.1961	587
5	0.0420	0.6037E 02	7.5343	570
5	0.0500	0.4765E 02	5.4091	451
5	0.0451	0.5485E 02	6.6115	530
5	0.0469	0.5184E 02	6.1086	501
5	0.0481	0.5014E 02	5.8248	482
5	0.0488	0.4916E 02	5.6601	470
5	0.0493	0.4857E 02	5.5615	463
6	0.0100	0.1040E 03	13.3307	228
6	0.0120	0.9865E 02	13.4335	227
6	0.0140	0.9769E 02	13.2723	256
6	0.0160	0.9652E 02	13.0754	289
6	0.0180	0.9506E 02	12.8320	327
6	0.0200	0.9325E 02	12.5284	367
6	0.0220	0.9099E 02	12.1507	410
6	0.0240	0.8821E 02	11.6863	451
6	0.0260	0.8488E 02	11.1299	487
6	0.0280	0.8104E 02	10.4879	514
6	0.0300	0.7680E 02	9.7806	530
6	0.0320	0.7237E 02	9.0402	532
6	0.0340	0.6796E 02	8.3034	522
6	0.0360	0.6376E 02	7.6026	502
6	0.0500	0.4507E 02	4.4791	295
6	0.0413	0.5442E 02	6.0432	421
6	0.0447	0.5014E 02	5.3265	369
6	0.0467	0.4797E 02	4.9633	338
6	0.0480	0.4678E 02	4.7651	321
6	0.0487	0.4610E 02	4.6505	311

6	0.0492	0.4570E 02	4.5840	305
6	0.0495	0.4545E 02	4.5421	301
9	0.0100	0.1062E 03	13.1858	237
7	0.0120	0.1007E 03	13.2729	246
7	0.0140	0.9930E 02	13.0408	281
7	0.0160	0.9754E 02	12.7456	320
7	0.0180	0.9529E 02	12.3696	361
7	0.0200	0.9246E 02	11.8967	401
7	0.0220	0.8900E 02	11.3191	436
7	0.0240	0.8496E 02	10.6446	462
7	0.0260	0.8050E 02	9.8996	475
7	0.0280	0.7586E 02	9.1249	474
7	0.0300	0.7131E 02	8.3643	461
7	0.0320	0.6705E 02	7.6534	438
7	0.0500	0.4618E 02	4.1641	212
7	0.0389	0.5581E 02	5.7761	336
7	0.0431	0.5124E 02	5.0108	280
7	0.0458	0.4904E 02	4.6421	250
7	0.0474	0.4786E 02	4.4449	235
7	0.0484	0.4718E 02	4.3319	226
7	0.0490	0.4680E 02	4.2666	221
7	0.0494	0.4655E 02	4.2257	217
10	0.0100	0.1082E 03	13.0277	241
8	0.0120	0.1025E 03	13.0808	262
8	0.0140	0.1006E 03	12.7568	300
8	0.0160	0.9809E 02	12.3384	339
8	0.0180	0.9493E 02	11.8102	375
8	0.0200	0.9111E 02	11.1711	404
8	0.0220	0.8673E 02	10.4412	421
8	0.0240	0.8206E 02	9.6608	425
8	0.0260	0.7738E 02	8.8788	417
8	0.0280	0.7294E 02	8.1383	398
8	0.0500	0.4854E 02	4.0560	173
8	0.0364	0.5918E 02	5.8388	287
8	0.0416	0.5402E 02	4.9760	231
8	0.0448	0.5161E 02	4.5722	205
8	0.0468	0.5034E 02	4.3585	192
8	0.0480	0.4961E 02	4.2366	184
8	0.0488	0.4919E 02	4.1662	180
8	0.0493	0.4893E 02	4.1222	177
9	0.0120	0.1042E 03	12.8589	272
9	0.0140	0.1017E 03	12.4324	308
9	0.0160	0.9841E 02	11.8915	340
9	0.0180	0.9450E 02	11.2387	365
9	0.0200	0.9008E 02	10.4996	379
9	0.0220	0.8540E 02	9.7192	381
9	0.0240	0.8078E 02	8.9474	371
9	0.0260	0.7645E 02	8.2244	353
9	0.0500	0.5131E 02	4.0188	157

9	0.0352	0.6221E 02	5.8442	250
9	0.0408	0.5695E 02	4.9650	203
9	0.0443	0.5448E 02	4.5513	182
9	0.0465	0.5317E 02	4.3313	171
9	0.0478	0.5242E 02	4.2056	165
9	0.0487	0.5199E 02	4.1329	162
9	0.0492	0.5172E 02	4.0873	160
9	0.0495	0.5157E 02	4.0628	159
10	0.0120	0.1058E 03	12.6227	273
10	0.0140	0.1027E 03	12.1055	304
10	0.0160	0.9892E 02	11.4765	328
10	0.0180	0.9463E 02	10.7596	342
10	0.0200	0.9006E 02	9.9974	346
10	0.0220	0.8552E 02	9.2380	339
10	0.0240	0.8122E 02	8.5207	325
10	0.0500	0.5424E 02	4.0065	150
10	0.0339	0.6589E 02	5.9583	233
10	0.0401	0.6026E 02	5.0162	192
10	0.0439	0.5762E 02	4.5737	173
10	0.0462	0.5622E 02	4.3389	163
10	0.0477	0.5542E 02	4.2049	158
10	0.0485	0.5496E 02	4.1277	155
10	0.0491	0.5467E 02	4.0792	153
10	0.0494	0.5452E 02	4.0532	152

** OPTIMAL SOLUTION **

NO. OF SERVERS= 6
SERVICE RATE=0.050000
TOTAL COST= 0.451E 02

REFERENCES

1. Balachandran, K. R., "Parametric Priority Rules: An Approach to Optimization in Priority Queues," Operations Research 18, 1970, 526-539.

2. Bhat, U. N., "Sixty Years of Queueing Theory," Management Science 15B, 1969, 280-294.

3. Bowman, E. H. and Fetter, R. B., Analysis for Production Management, Richard D. Irwin, Inc., Homewood, Illinois, 1961, 289-303.

4. Brigham, G., "On a Congestion Problem in an Aircraft Factory," Operations Research 3, 1955, 412-428.

5. Clarke, A. B. and Disney, R. L., Probability and Random Processes for Engineers and Scientists, John Wiley & Sons, Inc., New York, 1970.

6. De Cani, J. S., "A Queueing Model for Business Students," Management Science 9, 1962, 278-286.

7. Durst, L. K., Proceedings CUPM Geometry Conference, Part 1, August 1967, Mathematical Association of America, page 57.

8. Evans, R. V., "Programming in Markov Processes," Technical Memorandum No. 123, October 1968, Operations Research Department, Case Western Reserve University.

9. ___________, "Design of Discrete Parameter Markov Systems," Technical Memorandum No. 127, October 1968, Operations Research Department, Case Western Reserve University.

10. ___________, "Programming Problems and Changes in the Stable Behavior of a Class of Markov Chains," Technical Memorandum No. 126, 1968, Operations Research Department, Case Western Reserve University.

11. Fox, Bennett, "Discrete Optimization Via Marginal Analysis," Management Science 13, 1966, 210-215.

12. Hillier, F. S., "Economic Models for Industrial Waiting Line Problems," Management Science 10, 1963, 119-130.

13. ___________, "Cost Models for the Application of Priority Waiting Line Theory to Industrial Problems," Journal of Industrial Engineering 16, 1965, 178-185.

14. Kumin, H. J., "The Design of Markovian Congestion Systems," Technical Memorandum No. 115, Operations Research Department, Case Western Reserve University, 1968.

15. Little, J. D. C., "A Proof of the Queueing Formula $L = \lambda W$," Operations Research 9, 1961, 383-387.

16. Mangelsdorf, T. M., "Waiting Line Theory Applied to Manufacturing Problems," S. M. Thesis, MIT, 1955. Reprinted in Analyses of Industrial Operations, edited by Bowman, E. H. and Fetter, R. B., Richard D. Irwin, Inc., Homewood, Illinois, 1959.

17. Morse, P., Queues, Inventories and Maintenance, John Wiley & Sons, Inc., New York, 1958.

18. ___________, "The Application of Queueing Theory in Operations Research," printed in Queueing Theory, edited by R. Cruon, American Elsevier Publishing Company, Inc., New York, 1967. Peck, L. G. and Hazelwood, R. N., Finite Queueing Tables, John Wiley & Sons, Inc., New York, 1958.

19. Riordan, J., Stochastic Service Systems, John Wiley & Sons, Inc., New York, 1962.

20. Rockafellar, R. T., Convex Analysis, Princeton University Press, Princeton, New Jersey, 1970, 10-25.

21. Rolfe, A. J., "A Note on Marginal Allocation in Multiple-Server Service Systems," Management Science 17, 1971, 656-658.

22. Saaty, T. L., Elements of Queueing Theory, McGraw Hill Book Company, New York, 1961.

23. Sasieni, M. W., private communication, 1967.

24. Stidham, S., "On the Optimality of Single-Server Queueing Systems," Operations Research 18, 1970, 708-731.

25. ___________, "A Last Word on $L = \lambda W$," Operations Research 22, 1974, 417-421.

26. Taha, H. A., Operations Research, An Introduction, the Macmillan Company, New York, 1971.

27. Zangwill, W. I., Nonlinear Programming: A Unified Approach, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1969.