FOR REFERENCE

NOT TO BE TAKEN FROM THIS ROOM

OPTIMAL CONTROL
OF
M/M/S QUEUEING SYSTEMS

by

NURİZER ÖZEKİCİ (MÜCELDİLİ)

Bogazici University Library

39001100315715 14

Submitted to the Faculty of the School of Engineering

in Partial Fulfillment of

the Requirements for the Degree of

Master of Science

in

Industrial Engineering

BOĞAZİÇİ UNIVERSITY

1982

We hereby recommend that the thesis entitled "*Optimal Control of M/M/S Queueing Systems*" submitted by Nurizer Özekici (Müceldili) be accepted in partial fulfillment of the requirements for the Degree of Master of Science in Industrial Engineering, School of Engineering, Boğaziçi University.

<u>Examining Committee</u>

Dr. Akif EYLER

Doç. Dr. Yorgo ISTEFANOPULOS

Dr. Ali Rıza KAYLAN
(Thesis Advisor)

Date: 12.5.1982

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS

This study was conducted under the supervision of Dr. Ali Rıza Kaylan to whom I wish to express my sincere gratitude for his guidance and helpful suggestions throughout the course of my research.

I am also greatful to Doç. Dr. Yorgo Istefanopulos and Dr. Akif Eyler for serving on my thesis committee and their comments which led to refinements in the earlier version of this document.

I wish to thank to my mother for her constant moral and encouragement throughout my education.

My hearthfelt thanks go to my husband who has helped me in innumerable ways. His knowledge have been a valuable contribution to this thesis.

ABSTRACT

This thesis characterizes the optimal operating policy of a multi-server queueing system subject to Poisson arrival process and exponentially distributed service times (M/M/S queue). Optimal policy minimizes the long-run total expected discounted cost to the system. The cost components of the system are taken as the server cost and the holding cost which is considered as the lost profit from the business or the lost production with respect to the type of the system.

Markov Decision Theory is used in the characterization of the controlled process. Generator is the basic tool of the formulation. Application of some solution procedures is very easy for this type of formulation. Two different algorithms are presented to obtain the optimal policy: Successive approximation algorithm and policy improvement algorithm.

Optimal policy for a simple maintenance problem is found using these two methods. Computational experiments on the computer indicate that the policy improvement method converges to the optimal policy more quickly.

The theoretic results are extended to tandem queueing systems at the end.

ÖZET

Bu çalışma, Poisson geliş süreçli, üssel dağılımlı hizmet süreli ve çok işgörenli (M/M/S) kuyruk sistemlerinde eniyi işletme politikalarını belirlemeyi amaçlamaktadır. Eniyi politika, sistemdeki toplam maliyet beklentisini enazlayan olarak tanımlanmaktadır. Toplam maliyet işlevi uzun süreli ele alınmakta ve paranın zaman içerisinde değer yitirmesi özelliğinide içermektedir.

Denetim altındaki sistemin özelliklerini tanımlamakta kullanılan yaklaşım Markov Karar Kuramına dayanmaktadır. Üretmen (generator), model gösteriminde yararlanılan temel araçtır. Bu tür model gösterimlerinde çeşitli çözüm yöntemleri kolaylıkla uygulanabilmektedir. Bu çalışmada eniyi politikanın bulunabilmesi için ardarda yaklaşıklama ve kural iyileştirme yöntemleri kullanılmıştır.

Küçük boyutlu bir bakım sisteminde eniyi politikayı belirleme problemi bu iki yöntemle ayrı ayrı çözülmüştür. Sistem kısaca N makinalı bir üretim ünitesinde makinaların zaman zaman bozulmalarını ve bakım ünitesinde onarılmalarını içermektedir. Gözönüne alınan maliyetler, hizmet görenlerin sayısına göre birim zaman onarım maliyeti ve makinaların çalışmamasından doğan üretim kaybı maliyetleridir. Her durum için işgörenlerin sayısını belirleyen politikalar içinde en-

iyisi olarak sistemin maliyetlerini enazlayanı seçilmektedir. Bu örnek üzerinde iki algoritmanın çözümsel verimliliği kıyaslanmış ve kural iyileştirme yönteminin yakınsama hızının daha çabuk olduğu gözlenmiştir.

Kuramsal sonuçlar seri bağlı kuyruk sistemleri için de ayrıca genişletilmiştir.

I. INTRODUCTION

The purpose of this thesis is to characterize the optimal operating policy of the multi-server queueing systems with a Poisson arrival process and exponentially distributed service times and with finite or infinite capacity. The shorthand notation M/M/S/K refers to such a queue. After the characterization, two algorithms will be presented to obtain the optimal policy. Then the theoretic results will be extended to some other queueing systems.

The arrival process is assumed to be a Poisson process with a content dependent arrival rate and the service time of each server is exponential. At any time, the decision maker has to decide on the total number of servers to be employed by observing the total number of customers in the queue. To be more precise, if the content of the queue is x and policy $\pi(x)$ is used, then there is a Poisson stream of arrivals with rate $\lambda(x)$ and the number of servers working is $\pi(x)$. If the service rate of each server is μ, then the queue size either increases by one with rate $\lambda(x)$ by an arrival or decreases by a service completion with rate $\mu\pi(x)$ whichever comes first.

For each policy $\pi(x)$, we define the economic effectiveness of the service station by the long-run total expected discounted return. Shortage, holding and server costs effect this return by discounting continuously at rate $\alpha > 0$. A policy minimizing this total expected discounted return is catled optimal. We seek to find the conditions under which an optimal policy exists. We also seek to characterize an optimal policy, if one exists.

Based on the Markovian property of the process, Markov Decision Theory is applied. Generator is an important tool in the optimal control of Markov processes. So, using the generator and its characteristics the dynamic functional equations are written. Then an algorithm is developed to solve these equations.

Before we discuss the contents of this thesis, we present a brief summary of the literature on the optimum control of queueing systems.

A great deal of emphasis in queueing theory recently has been in the area of design and control. The early works on the control of queues were essentially in the form of descriptive analysis of a set of plausible control policies from which 'optimal policies' were selected by mathematical optimization techniques. More recently researchers have begun to employ Markovian Decision model to solve queueing control problems.

Considerably more effort has been put forth on the rate-control models which deal with when and how arrival or service rates should be changed to optimize some objective function.

The work to date on these models can be classified as:

a) Control of server
b) Control of service rate
c) Control of arrivals.

Here we only deal with the first two cases. In the first case, the control action is to turn the server on or off at the service completions or at the customer arrivals.

Miller (1969) considers a c-server no-waiting-space queueing system with m customer classes, each class yielding a different reward. The queueing model is assumed to be M/M/C/C, and the problem is formulated as an infinite horizon, continuous time Markov decision problem. The objective function here is the expected reward rate over an infinite planning horizon and it is desired to find the policy which maximizes this. Qualitative results which characterize the form of the optimal policy are given, as well as a comparison, via simulation, of some approximate policies deduced from the analysis, for arbitrary service-time distributions.

Heyman(1968) considers an M/G/1 state dependent model. He considers a server start-up cost, a server shut-down cost and a cost per unit time when the server is running and a customer waiting cost. He proves that the form of the optimal policy is "turn the server on when there are n customers in the system and turn the server off when the system is empty." Heyman considers various combinations of cases involving discounting or not discounting costs over time and a finite or infinite planning horizon.

Sobel(1969) considers the same problem as did Heyman, namely, starting and stoping service but generalizes it to G/G/1, as well as assuming more general cost structure. Considering the average cost rate over an infinite horizon, he shows that the policy form is "provide no service if the system size is m or less, when system size increases to M (M > m), turn the server on and continue serving until the system size again drops to m". He refers to these as (M,m) policies.

Blackburn(1972) also treats an extension of Heyman's model in that he incorporates balking and reneging into the M/G/1 queue. Now, the longer the server is in the off position, the more chance there is of a balk or renege. He shows that the stationary optimal policy which maximizes discounted reward over an infinite horizon can also be characterized by a simple pair of critical values (M,m). Blackburn analyzes the problem as a Markov renewal decision process.

Magazine(1971) shows that a policy of (M,m) form is also optimal for the M/M/1 system under periodic review and extends his work and shows the existence of an analogous rule for the multi-server systems. Formulation as a dynamic programming problem is given and proofs for existence are represented for finite horizon, infinite horizon and average cost criteria.

In the case of "b", a service rate can be chosen from a set of allowable service rates at customer arrivals or at service completions.

Brosh(1970) considers a two service-rate model, with a limit on queue size. The system is observed at short, equally spaced intervals of time, and at each interval, after observing the system size, a decision on which service rate to use is made for the next interval. A Markovian analysis is used, since the assumption is made that in any interval at most one arrival occurs with the probability of an arrival being λ. The probability of a service completion is μ_i depending on which service rate was chosen. Thus Brosh is essentially dealing with a state-dependent M/M/1/K model. Considering different costs for each service rate and cost for lost customers, the optimal policy which minimizes the long-run average cost rate is desired. Brosh categorizes the structure of the policy space, eliminating from consideration those policies which are dominated by better ones, and from this structuring of the space develops an algorithm which searches over the admissible policies.

Crabill(1972) has employed a continuous time. Markovian decision model to investigate the M/M/1 queue with k possible service rates. Including holding and service costs he finds the optimal policy which minimizes the long-run average cost rate. He proves that the optimal policy is characterized completely by k-1 numbers and that the optimal service rate is nondecreasing in the state of the system.

Lipman(1975) generalizes Crabill's results by implementing a different cost structure. Also Lipman establishes the existence of monotone optimal discounted and average cost policies.

Mitchell(1973) considers a single server, Poisson arrival general service queuing system in which the service rate may be varied continuously between fixed limits. The problem is to find a policy for selecting the service rate which minimizes the expected average service plus holding cost per unit time. Considering it as a Markov decision process, the model is approximated in that the service rate can be changed only at equally spaced points in time. He proves that if (i) the service cost rate is a convex function of the service rate and (ii) the holding cost rate is a polynomial approximation to a convex function of the work remaining in the system, then there exists a stationary deterministic optimal policy in which the service rate is a nondecreasing function of the work remaining in the system.

There does not seem to be any study of finding optimal policy by using the generator of the Markov process, in the queuing models. But there are some studies on Markov Decision processes which use generator in general without making any assumption about the specific nature of the controlled process.

Miller(1968) considered a Markov decision process with continuous time parameters by restricting his attention to a finite state space case. Later, Kakumanu(1972) extended his results to the case of a countable state space. Markov decision processes with continuous time parameter and fairly general state space case is studied by Doshi(1976).

This thesis combines the studies in the aforementioned areas. As it is pointed out in the literature survey, there is no work except Magazine(1971) on the selection of the optimum number of servers for a given cost structure. The studies which are done in the control of server are generally based on the decision of shutting down the available single server or starting it up.

The closest work related to our thesis is Magazine's article as it is seen in the literature survey. Even though this thesis is primarily focusses on the optimal control of M/M/S/K system, it clearly differs from Magazine's work in several respects. Magazine takes the arrival rate as constant, but in our study arrival rate depends on the queue content. He assumes that the decision points are at equally spaced time intervals. But we are observing the system continuously and we could make a decision any time we want by noting the number of customers at that time. Also the cost structures are different. He considers a constant shutting-down cost, starting-up cost and unit operating cost for an open server and a convex holding cost function. We have no switching cost and our server cost is not constant. The requirement for the server and holding costs in our thesis is to be real-valued, nonnegative, bounded functions.

After defining the structure of the system Magazine gives the dynamic programming formulation for the infinite and finite horizon cases. At this stage we completely follow a different approach from Magazine. We formalize the control problem as a Markov Decision problem and then give two different solution procedure to obtain the optimal policy.

In the remainder of this chapter, scope and organization of this thesis will be briefly mentioned.

Chapter II deals with the analysis of M/M/S/K queue problem. In Section 1, we will describe the control problem, define the admissible policy set and then we will prove the Markovian property of the system. Section 2 defines the methods to find the generator of the process. Since generator has an important role in our study we give two method. The first one is from Breiman(1968) which can be used for every general process and the second method is from Çınlar(1975) which is very simple for the single station birth and death processes. Section 3 characterize the expected discounted cost function for any arbitrary policy and proves its uniqueness.

In the first section of Chapter III, the characterization of the minimum expected discounted return and that of the optimal policy is given. Section 2 describes a version of successive approximation algorithm and proves that it generates a sequence of iterating cost functions which finally converge to the minimum expected discounted cost. Section 3 defines the policy-improvement algorithm which is originally suggested by Doshi(1976).

Chapter IV extends the original single station model to the series queue model and also to the optimum service rate selection models. Section 3 compares the algorithms by solving a maintenance system as an example on the computer.

Chapter V summarizes the conclusions of the thesis.

II. ANALYSIS OF THE QUEUE PROBLEM

This chapter analyzes all characteristics of our M/M/S/K service system. After the analysis of the system's Markovian property, the basic Markov decision process is presented. Since the generator is a basic tool in the control of Markov process, all characteristics of it will be given.

Using generator, the dynamic functional equations is found for the uncontrolled queue process and the existence of the unique solution to this functional equations is verified.

II.1 Description of The Control Problem

Let X_t denote the number of customers present in the queue at time $t \geq 0$. The state space of X_t is $E = \{0,1,2,...,K\}$, where K denotes the queue capacity which is either infinite or a given finite positive integer. In some queueing processes there is a physical limitation to the amount of waiting room, so that when the line reaches a certain length, no further customers are allowed to enter until space becomes available by a service completion. These are referred to as finite queuing systems. In our model

K is finite, in other words when the number of customers present in the queue is equal to K, the new arrivals cannot enter the queue until a departure occurs. The results obtained in this thesis are stated for finite K, but similar results can be obtained for the infinite queue capacity case. The similarity will be pointed out by remarks throughout the thesis.

At any time t, the planner observes X_t customers present in the queue, and based on that information he determines the number of servers to be employed. In other words, if S_t is the number of servers employed at time t, then S_t is a function of X_t only, i.e. $S_t = \pi(X_t)$ for some nonnegative integer valued function π defined on E. Clearly $\pi(.)$ is the control function in our problem which gives the number of servers when the queue content is (.) and it is only reasonable to assume that $\pi(x) \in M_x \equiv \{0,1,2,...,x \wedge m\}$ for all $x \in E$ where m is some positive integer denoting the maximum number of servers that can be employed and $x \wedge m \equiv \min(x,m)$. This assumption implies that the number of servers employed cannot exceed the number of customers in the system or the number of available servers.

The customer arrivals are modeled as a Poisson process whose parameter (mean arrival rate) varies with the total number of customers present in the system. Thus, this is a state-dependent arrival process. Service times of servers are assumed to be independent and identically distributed with an exponential distribution. If the content of the queue is x, then there is a Poisson arrival with rate $\lambda(x)$ and the number of servers working is $\pi(x)$. If the common service rate is μ, then the queue size either increases by an arrival with rate (x) or decreases by a departure with rate $\mu\pi(x)$ whichever comes first. In real practice, it is often likely that arrivals become discouraged when the queue is long and may not wish to wait. If people see K ahead of them in the system, they do not join and $\lambda(K) = 0$. Figure 1 shows the multiserver system which is described above.

(1.1) DEFINITION: An admissible policy is a measurable function mapping E into M. Let M be the set of all admissible controls, then it is reasonable to define

$$M = (\pi:\{0,1,2,\ldots,K\} \to \{0,1,\ldots,m\})$$

So, M is the set of all bounded, positive and integer valued functions defined on E and bounded by an integer m.

(1.2) REMARK: Thoughout this thesis, we require that $S_t = 0$ is the only admissible decision whenever $X_t = 0$. That means, for all $\pi \in M$, $\pi(0) = 0$.

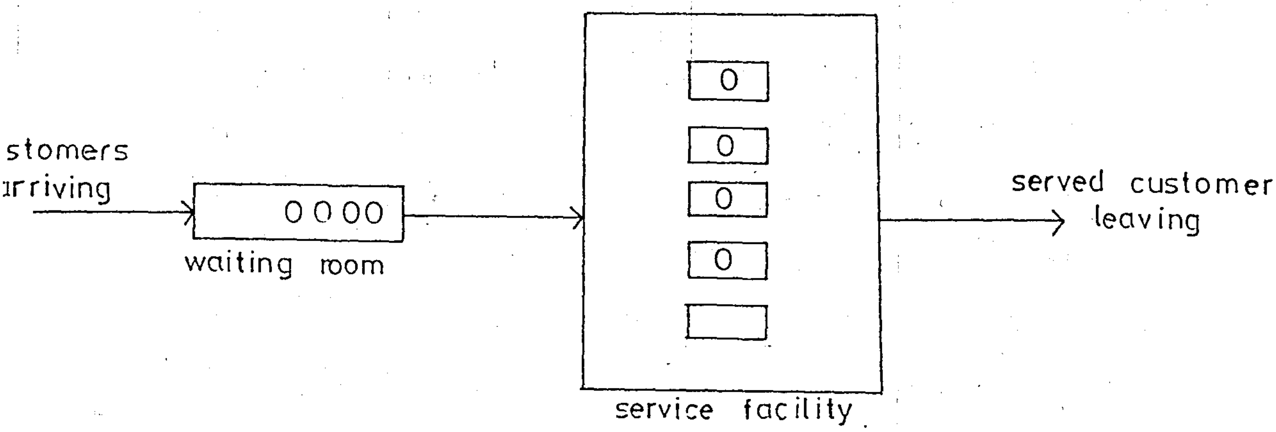

Fig. 1 - Multiserver queueing system

Let $A = \{A_t;\ t \geq 0\}$ and $D = \{D_t;\ t \geq 0\}$ be the customer arrival and departure processes respectively. In other words, A_t and D_t are the total number of arrivals and departures until time t respectively. Then it is clear that the queue content process $X = \{X_t;\ t \geq 0\}$ is given by,

$$X_t = A_t - D_t$$

which implies that,

$$X_{t+s} = X_t + A_{t+s} - A_t - (D_{t+s} - D_t)$$

This shows that the number of customers at time 't+s' is equal to the sum of X_t and the number of arrivals during the interval [t,t+s), less the number of services completed during [t,t+s). In generalized Poisson processes, the numbers of arrivals in nonoverlapping intervals are statistically independent; that is the process has independent increments. Therefore, the number of arrivals during [t,t+s) is independent of everything else that went on before time t. Also from the memoryless property of the exponential distribution the remaining service times are completely independent of the past. Memoryless property states that the remaining service time of a customer currently in service is independent of how long he has already been in service. Hence the number of services completed during $[t,t+s)$ can depend only on X_t and the arrivals during this interval.

This analysis shows the Markovian property of the queue process $X = \{X_t;\ t \geq 0\}$ which states that future behaviour is independent of the past given the present. That is,

$$P\{X_{t+s} = i \mid X_u; u \leq t\} = P\{X_{t+s} = i \mid X_t\}$$

The size of the queue at time t increases by one when an arrival occurs or decreases by one when a service is completed. This is a pure jump process which is also referred to as a "Birth and Death Process".

Figure 2 shows a typical realization of the queuing process X. T_1 is the time of the first arrival and at that time X_t increases by one, then $X_{T_1} = 1$. This first customer requires service up to time D_1. But before time D_1, two more customers arrive at time T_2 and T_3 which increase the number of customers to 2 and 3 respectively. When the first customer leaves the system at time D_1, queue content decreases by one then $X_{D_1} = 2$, and so on.

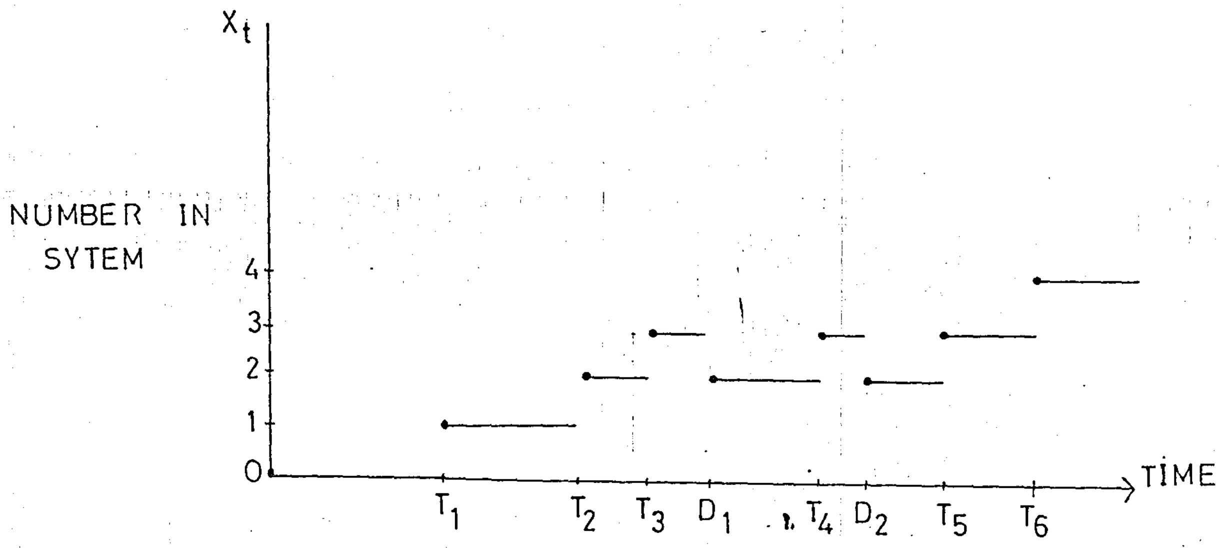

Fig. 2 - A realization of Markov Process

In the following theorem we prove that X is indeed a Markov process.

(1.3) THEOREM: The process $X = \{X_t;\ t \geq 0\}$ is a Markov process with state space .

Proof: Let V_t and W_t be respectively the lengths of times from t until the instants of the next arrival and next departure. That is,

$$V_t = \inf\{s \geq t;\ X_s = X_{s-} + 1\} - t$$

$$W_t = \inf\{s \geq t;\ X_s = X_{s-} - 1\} - t$$

Throughout the following we will let

$$P_x\{\cdot\} \equiv P\{\cdot | X_o = x\}, \qquad x \in E$$

for notational convenience.

We shall first find the probability that an arrival occurs before a departure after time t given the past information and that $X_t = x$

$$
\begin{aligned}
P\{V_t \le W_t | X_u; u \le t,\ X_t = x\} &= P\{V_t \le W_t | X_t = x\} \\
&= P_x\{V_t \le W_t\} \\
&= E_x[P_x\{V_t \le W_t | V_t\}] \\
&= E_x[e^{-\pi(x)\mu V_t}] \\
&= \int_0^\infty \lambda(x) e^{-\lambda(x)v}\, e^{-\pi(x)\mu v}\, dv
\end{aligned}
$$

$$
(1.4) \qquad = \frac{\lambda(x)}{\lambda(x) + \pi(x)\mu}
$$

The first equality follows from the fact that once X_t is given both V_t and W_t are independent of the past since interarrival and service times have the memoryless property. The second equality is simplified notation and the third one follows from basic probability theory. The fourth equality simply states that if $\pi(x)$ servers are employed in the system and there are no arrivals then W_t or the time of first service completion is exponentially distributed with parameter $\mu\pi(x)$. Similarly if there are no departures the fact that V_t or the time of the first arrival after time t is exponentially distributed with parameter $\lambda(x)$ justifies the fifth equality.

A similar argument given above shows that,

$$P_x\{V_t > u, W_t > u\} = P_x\{A_{t+u} - A_t = 0, W_t > u\}$$

$$= P_x\{W_t > u | A_{t+u} - A_t = 0\} P_x\{A_{t+u} - A_t = 0$$

$$= e^{-\pi(x)\mu u} \cdot e^{-\lambda(x) u}$$

(1.5) $$= e^{-(\lambda(x) + \pi(x)\mu)u}$$

From the definiton in Çınlar(1975,p.271) equation (1.4) and (1.5) imply that X is a Markov process with state space E. Furthermore, since X can either increase or decrease by one, it is a 'birth and death process'.

(1.6) REMARK: It is clear from the proof given above that limiting the queue capacity does not effect the proof and Markovian property remains unchanged.

Before we formally state the total expected cost, we shall attempt to give a physical interpretation of individual cost components in our system.

Decisions regarding the amount of service capacity to provide usually are based primarily on two considerations: (i) the cost incurred by providing the service, (ii) the cost of waiting for that service. It is apparent that these two cost components create conflicting pressures on the decision maker. The objective of reducing service costs recommends a minimal level of service. On the other hand, long waiting times are undesirable, which recommends a high level of service. Therefore, for the comparison of service costs and waiting times, it is necessary to adopt a common measure of their impact. The natural choice for this measure is monetary so that it becomes necessary to estimate the cost of waiting which is also referred as holding cost.

A common viewpoint in practice is that the cost of waiting is often too intangible to be amenable to estimation. For different types of situations, the subjective waiting cost can be viewed as follows:

For the profit-making organizations where the customers are external to the organization providing the service, the cost of holding probably would consist primarily of the lost profit from lost business. This lost business may occur immediately (because the customer grew impatient and left) or in the future (because the customer was considerably irritated that he did not come again.)

For the social service systems, the cost of waiting usually is a social cost of some kind. It is necessary to evaluate the consequences of waiting for the indivuals involved or for society as a whole and to try to impute a monetary value to avoid these consequences.

A situation that may be more amenable to estimating waiting costs is one in which the customers are internal to the organization providing the service. Business-industrial service systems are good examples for this case. For example, the customers may be the machines or employees of a firm. The primary cost of waiting for this case may be the lost profit from lost productivity.

Throughout this thesis holding (waiting) cost is symbolized by $h(X_t)$. It is assumed that $h(X_t)$ is quantified according to the aforementioned guidelines and made available to the investigator. Service cost is denoted by $c(S_t) = c(\pi(X_t))$ and is basically considered as the cost which is paid to the servers. The only requirements for the functions h and c are to be nonnegative, real-valued, bounded and monotonically increasing functions.

In addition to the above costs, we have a shortage cost for the finite systems. It incurres only when the system size achieves the full capacity level of the waiting space. When the system is completely full the new arrivals cannot enter ($\lambda(K) = 0$) and the system will have a lost profit with rate ℓ per lost customers. If we take the avarge arrival rate as γ then the shortage cost rate is $\gamma\ell$.

Economic effectiveness of the system is described by the expected total discounted cost composed of the three cost components described above. In this thesis, we try to find the policies which minimize the expected discounted cost. Since the process has the Markovian property this is a Markov Decision Problem.

For a formal statement of Markov Decision Problem, let $D_s(x,t)$ denote the expected cost incurred up to time t by following the policy s and with x customers in the system initially. And let $\alpha > 0$ be the interest rate used for discounting future cost, i.e.the present value of cost c incurred at time t is $ce^{-\alpha t}$.

Let the expected continuously discuounted cost of a policy π over an infinite time horizon be denoted by,

$$V_\pi(x) = E_\pi[\int_0 e^{-\alpha t} D_s(x,t)] \quad ,$$

Let V(x) be the minimum expected discounted cost function,

$$(1.7) \quad V(x) = \inf_{\pi \varepsilon M} V_\pi(x) \quad , \quad x \geq 0$$

Define $\pi^* \varepsilon M$ as an optimal policy if,

$$V_{\pi^*}(x) = V(x) \quad , \quad x \geq 0$$

II.2 The Generator of The Queue Process

The generator plays an important role in the optimal control of Markov processes. The objective of this section is to find an expression for the generator of the queue process X.

There are some relationships between the transition function, transition matrix and the generator of a process.

If the transition function P_t is known, then its derivative at $t = 0$ gives the generator A. The generator can determine the transition matrix Q which shows the probability of going from one state to another. Considering the converse problem, knowing the transition matrix Q and $\gamma(i)$ (the parameter of exponential distribution of a sojourn time in state i) generator A can be found. Generator shows the rate of change of state. After calculating the generator, transition function can be computed. One method is to solve thi infinite system of differential equations.

$$\frac{d}{dt} P_t = AP_t$$

$$\frac{d}{dt} P_t = P_t A$$

These equations are called, respectively, Kolmogorov's backward and forward equations. If the state space of Markov process is discrete, then the generator could be written in matrix form. And the computation of P_t from A is done by using certain matrix theoretic methods.

(2.1) DEFINITION: For every $\pi \in M$, the generator A_π of the process X, the range $R(A_\pi)$ of A_π and the domain $D(A_\pi)$ of A_π are defined as follows:

i) $R(A_\pi)$ is the set of all bounded measurable functions f on E such that,

$$E_x[f(X_t)] \to f(x) \quad \text{as } t \downarrow 0, \quad \text{for all } x \in E$$

ii) $D(A_\pi)$ is the set of all $f \in R(A_\pi)$ such that,

$$[E_x\, f(X_t) - f(x)]/t$$

converges boundedly pointwise on E as $t \downarrow 0$ to a function in $R(A_\pi)$

iii) For any function $f \in D(A_\pi)$, $A_\pi f$ is defined to be the limiting function in (ii). Such that

$$\lim_{t \downarrow 0} \frac{E_x[f(X_t)] - f(x)}{t} = A_\pi f$$

Note that this definiton of the generator is equivalent to the weak infinitesimal generator given in Breiman(1968, pp. 341).

(2.2) PROPOSITION: For each $\pi \in M$, $R(A_\pi)$ consists of the set of all bounded functions on E.

Proof: Let f be any bounded function on E and T be the time of the first jump, such that

$$T = \inf\{t \geq 0, \quad X_t \neq X_o\}$$

For notational convenience, define

$$P_t f(x) = E_x[f(X_t)] = E[f(X_t) \mid X_o = x]$$

For every $x \in E$ and $t \geq 0$,

$$(2.3) \quad P_t f(x) = E_x[f(X_t); T > t] + E_x[f(X_t); T \leq t]$$

$$E_x[f(X_t).1_{\{T>t\}}] = E_x[f(x).1_{\{T>t\}}]$$

$$= f(x).P_x\{T > t\}$$

$$(2.4) \quad = f(x).e^{-(\lambda(x) + \pi(x)\mu)t}$$

$$(2.5) \quad E_x[f(X_t).1_{\{T\leq t\}}] = E_x[f(X_t).1_{\{T\leq t\}}.1_{\{T=A\}}]$$

$$+ E_x[f(X_t).1_{\{T\leq t\}}.1_{\{T=D\}}]$$

$$= a_1(x,t) + a_2(x,t)$$

A and D represent the times of the first arrival and first departure respectively. Using the strong Markov property at T we obtain,

$$a_1(x,t) = E_x[E_x[f(X_t).1_{\{A\leq t\}}.1_{\{T=A\}} \mid T]]$$

$$= E_x[1_{\{A=T\}}.1_{\{A\leq t\}}.E_{X_T}[f(X_{t-T})]]$$

$$= E_x[1_{\{A=T\}}.1_{\{A\leq t\}}.E_{x+1}[f(X_{t-T})]]$$

$$= E_x[1_{\{A=T\}}.1_{\{A\leq t\}}.P_{t-T}f(x + 1)]$$

$$= E_x[E_x[1_{\{A\le t\}}\cdot 1_{\{A<D\}}\cdot P_{t-A}f(x+1)\,|A]]$$

$$= E_x[1_{\{A\le t\}}\cdot P_{t-A}f(x+1)\,.e^{-\pi(x)\mu A}]$$

$$= \int_0^\infty \lambda(x)e^{-\lambda(x)u}\; 1_{\{u\le t\}}\cdot P_{t-u}f(x+1)e^{-\pi(x)\mu u}\,du.$$

(2.6) $$= \frac{\lambda(x)}{\lambda(x)+\pi(x)\mu}\int_0^t (\pi(x)\mu+\lambda(x))e^{-(\lambda(x)+\pi(x)\mu)u}P_{t-u}f(x+1$$

Similarly, using the strong Markov property at T we obtain

$$a_2(x,t) = E[f(X_t)\,.1_{\{T\le t\}}\cdot 1_{\{T=D\}}]$$

(2.7) $$= \frac{\pi(x)\mu}{\lambda(x)+\pi(x)\mu}\int_0^t (\lambda(x)+\pi(x)\mu)e^{-(\lambda(x)+\pi(x)\mu)u}P_{t-u}f(x-1)d$$

By a change of variables setting w = t - u we obtain

(2.8) $$a_1(x,t) = \frac{\lambda(x)}{\lambda(x)+\pi(x)\mu}\int_0^t (\pi(x)\mu+\lambda(x))e^{-(\lambda(x)+\pi(x)\mu)(t-w)}P_w f(x+1)$$

(2.9) $$a_2(x,t) = \frac{\pi(x)\mu}{\lambda(x)+\pi(x)\mu}\int_0^t (\lambda(x)+\pi(x)\mu)e^{-(\lambda(x)+\pi(x)\mu)(t-w)}P_w f(x-1$$

Putting (2.8), (2.9) and (2.4) together

(2.10) $$P_t f(x) = f(x)e^{-(\lambda(x)+\pi(x)\mu)t} + a_1(x,t) + a_2(x,t)$$

It follows from (2.8) and (2.9) that for all $x \in E$,

$$\lim_{t\downarrow 0} a_1(x,t) = \lim_{t\downarrow 0} a_2(x,t) = 0$$

Therefore, it is clear from (2.10) that

$$\lim_{t \downarrow 0} P_t f(x) = f(x)$$

So the range of the generator is the set of all bounded functions on E.

We will now try to characterize $D(A_\pi)$. To be able to do that first we will state a definiton.

(2.11) DEFINITION: A sequence of functions $\{f_t\}_{-}$ b(E) converges boundedly pointwise to a function $f \in b(E)$ as $t \downarrow 0$ if;

i) $\lim_{t \downarrow 0} f_t(X) = f(x)$ for every $x \in E$

ii) there exists some constant $M < \infty$ such that

$$||f_t|| = \sup_{x \in E} |f_t(x)| \leq M$$

for all t sufficiently small.

(2.12) PROPOSITION: For each $\pi \in M$, $D(A_\pi)$ consists of all $f \in R(A_\pi)$

Proof: By definition

$$A_\pi f(x) = \lim_{t \downarrow 0} \frac{1}{t}[P_t f(x) - f(x)] \quad , \quad f \in R(A_\pi)$$

where the domain $D(A_\pi)$ is the set of all $f \in R(A_\pi)$ for which this limit exists boundedly pointwise and belongs to $R(A_\pi)$.

BOĞAZİÇİ ÜNİVERSİTESİ KÜTÜPHANESİ

It follows from (2.10) that

$$\text{(2.13)}\quad \frac{1}{t}[P_t f(x) - f(x)] = \frac{1}{t} a_1(x,t) + \frac{1}{t} a_2(x,t) + \frac{1}{t} a_3(x$$

where $a_3(x,t) = e^{-(\lambda(x) + \pi(x)\mu)t} . f(x) - f(x)$

It is clear that for $f \varepsilon R(A_\pi)$

$$\text{(2.14)}\quad \lim_{t\downarrow 0} \frac{1}{t} a_1(x,t) = \lambda(x) f(x + 1)$$

$$\lim_{t\downarrow 0} \frac{1}{t} a_2(x,t) = \mu\pi(x) f(x - 1)$$

Also it follows from (2.8), (2.9) and definition (2.11) that, for all $f \varepsilon R(A_\pi)$

$$\frac{1}{t}|a_1(x,t)| \leq \lambda(x) ||f|| \frac{1}{t} \int_0^t e^{-(\lambda(x)+\pi(x)\mu)z} dz \leq \overline{\lambda}||f|$$

$$\frac{1}{t}|a_2(x,t)| \leq \pi(x)\mu||f|| \frac{1}{t} \int_0^t e^{-(\lambda(x)+\pi(x)\mu)z} dz \leq m\mu||f$$

where $\overline{\lambda} = \max_x \lambda(x)$ and m is the maximum number of servers. Since they are finite, it shows that these limits exists boundedly pointwise. Therefore $f \varepsilon R(A_\pi)$ is in $D(A_\pi)$ if and only if $(1/t)a_3(x,t)$ converges boundedly pointwise as $t\downarrow 0$.

To proceed, note that

$$e^{-(\lambda(x) + \pi(x)\mu)t} = 1 - (\lambda(x) + \pi(x)\mu)t + O(t)$$

where $O(t)/t \to 0$ as $t\downarrow 0$. Therefore,

$$\frac{1}{t}\, a_3(x,t) = \frac{1}{t}[f(x)(1 - (\lambda(x) + \pi(x)\mu)t + O(t)) - f(x)$$

Now,

$$(2.15) \quad \lim_{t\downarrow 0} \frac{1}{t}[-(\lambda(x) + \pi(x)\mu)t.f(x) + O(t)] = -(\lambda(x) + \pi(x)\mu)f($$

boundedly pointwise for all $f \in R(A_\pi)$

These results imply that the domain of A is also the set of bounded functions defined on E.

Çınlar(1975) gives a simple way of calculating the generator for the birth and death processes. Since we proved in Section 1, that the process $X = \{X_t, t \geq 0\}$ is a birth and death process we can use his definition.

Looking to the definition in Çınlar(1975, p. 271), the time rates of arrivals and departures, which are symbolized as a_i and b_i respectively when the population size is i, can be found by using the equation (1.4) and (1.5). Therefore,

$$(2.16) \quad a_i = \lambda(i)$$
$$b_i = \pi(i)\mu$$

Depending on the time rates the generator of a birth and death process is given like,

$$(2.17) \quad A = \begin{bmatrix} -a_o & a_o & & & 0 \\ b_1 & -a_1-b_1 & a_1 & & \\ & b_2 & -a_2-b_2 & a_2 & \\ & & . & . & . \\ & & & . & . & . \\ 0 & & & & \end{bmatrix}$$

Adopting this form to our process using equation (2.16), the generator of our finite capacity queue system for any $\pi \varepsilon M$ can be written

(2.18)
$$A_\pi = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ K \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & K-1 & K \end{array} \\ \left[\begin{array}{cccccc} -\lambda(0) & \lambda(0) & & & 0 & \\ \mu\pi(1) & -\lambda(1)-\pi(1)\mu & \lambda(1) & & & \\ & \ddots & \ddots & \ddots & & \\ & & & & & \\ 0 & & & & \mu\pi(K) & -\mu\pi(K) \end{array}\right] \end{array}$$

The matrix A_π shows that; by using policy $\pi \varepsilon M$, if there is one customer in the queue with rate $\mu\pi(1)$ there will be nobody in the queue $(A_\pi(1,0))$; and with rate $\lambda(1)$ another customer arrives and the queue content becomes two $(A_\pi(1,2))$. As you notice, $b_o = a_K = 0$. Because, if there is nobody in the queue the only possible action could be an arrival. Also if the queue is completely full, i.e. the process is in state K, an arrival could not be occur then $a_K = 0$. Also no pair of $(a_i, b_i) = (0,0)$. If $a_i = b_i = 0$, then there could not be any state called i.

(2.19) REMARK: For the case of infinite capacity system, the generator matrix will have infinite states and all of them will be written in the form of (2.17) after finding the time rates.

We could end up with the same result as in (2.18) by using the definition of A_π given in the definition (2.1)

$$\lim_{t \downarrow 0} \frac{1}{t}[P_t f(x) - f(x)] = A_\pi f(x)$$

Combining the equations (2.13), (2.14) and (2.15) we obtain for $0 < x < K$

$$\mu\pi(x)f(x-1) - (\lambda(x) + \pi(x)\mu)f(x) + \lambda(x)f(x) = A_\pi f(x)$$

II.3 The Uncontrolled Queue Process

In this thesis, our first objective is the characterization of the V(x) function which is defined by (1.7). Following the standard Markov Decision Theory, this is accomplished in three steps. First write down the dynamic functional equations. Secondly verify that there exists a solution to these functional equations and then investigate the solution properties. The final step is to confirm that this solution is indeed equal to the minimum expected discounted cost function.

The objective of this section is to do the first step of the Markov decision theory.

Now, we shall state a theorem without proof, which is the basic tool in our thesis. For a proof see Çınlar (1975, p. 257).

For a regular Markov process with state space E and generator A, Çınlar states;

(3.1) THEOREM: Let $\alpha > 0$. For any $g \in R(A)$ there exists a unique function $f \in D(A)$ which satisfies

$$(\alpha I - A) f = g$$

where

$$f(x) = E_x\left[\int_0^\infty e^{-\alpha t}\, g(X_t)\,dt\right], \qquad x \in E$$

In the above theorem $g(X_t)$ is the rate of reward at time t and f(x) represents the expected value of the total discounted reward given the initial state is x.

In our problem, we have two types of costs which are defined in Section 1. At time t, when the state is $X_t = x$, the rate of cost for any policy $\pi \in M$,

(3.2) $$g_\pi(x) = h(x) + c(\pi(x))$$

For the state K, $X_t = K$, we have a shortage cost in addition to the others

(3.3) $$g_\pi(K) = h(K) + c(\pi(K)) + \gamma \ell$$

Since h and c functions are given as bounded, g_π function will also be bounded. In proposition (2.2) we proved that the range of A is the set of all bounded functions on E, therefore $g_\pi \in R(A)$.

(3.4) THEOREM: Let $\alpha > 0$. For any policy $\pi \varepsilon M$, expected total discounted cost V_π is the unique solution of the system of linear equations,

(3.5) $$V_\pi(0) = \frac{1}{\alpha + \lambda(0)} [h(0) + c(0) + \lambda(0)V_\pi(1)] , \quad x = 0$$

$$V_\pi(x) = \frac{1}{\alpha + \lambda(x) + \pi(x)\mu}[h(x) + c(\pi(x)) + \lambda(x)V_\pi(x+1) + \mu\pi(x)V_\pi(x-1)] , \quad 0 < x < K$$

$$V_\pi(K) = \frac{1}{\alpha + \mu\pi(K)} [h(K) + c(\pi(K)) + \gamma\ell + \mu\pi(K)V_\pi(K-1)] , \quad x = K$$

Proof: Since V_π is defined for any $\pi \varepsilon M$ as,

$$V_\pi(x) = E_x[\int_0^\infty e^{-\alpha t} g_\pi(X_t)] , \quad x \varepsilon E$$

where $g_\pi(x)$ is defined in (3.2) and (3.3), according to the theorem (3.1) V_π is in the domain of A_π and is the unique solution of

$$(\alpha I - A\ )V_\pi = g_\pi$$

Then

(3.6) $$\alpha V_\pi = g_\pi + A_\pi V_\pi$$

Putting the generator A_π found in (2.18), we obtain

(3.7) $$\alpha V_\pi(0) = h(0) + c(0) - \lambda(0)V_\pi(0) + \lambda(0)V_\pi(1) , \quad x = 0$$

$$\alpha V_\pi(x) = h(x) + c(\pi(x)) - [\lambda(x) + \mu\pi(x)\ V_\pi(x)$$

$$+ \lambda(x)V_\pi(x+1) + \mu\pi(x)V_\pi(x-1)], \quad 0 < x < K$$

$$\alpha V_\pi(K) = h(K) + c(\pi(K)) + \gamma\ell + \mu\pi(K)V_\pi(K-1)$$

$$- \mu\pi(K)V_\pi(K), \quad x = K$$

Rearranging the above equations we end up with the equations (3.5).

(3.8) REMARK: In the case of infinite state space, the final equation which is for the state K will be omitted.

This result gives the characterization of the expected infinite time horizon discounted cost for any policy π when the rate of cost is given by g. Theorem (3.4) will be our basic tool in the optimal control problem as we shall see in the next chapter.

III. OPTIMAL CONTROL OF THE QUEUE PROCESS

We are first interested in characterizing the expected return function V_π for any policy $\pi \in M$. This is achieved by expressing V_π as the unique solution of the functional equations in theorem (3.4) in the previous chapter. Now, our objective is to obtain a similar functional equation characterization for the minimum expected discounted return.

After obtaining the sufficient condition of optimality; two algorithms will be presented to find the optimal policy.

III.1 A Sufficient Condition of Optimality

In this section, following the standard Markov Decision Theory, we characterize the minimum expected discounted return function V and obtain a sufficient condition of optimality.

We shall first state a definition and a theorem which will aid us in providing our main theorem.

(1.1) DEFINITION: Let B be the Banach space with the usual supremum norm $||\cdot||$. A mapping $T:B \to B$ is said to be a contracting mapping if

$$||Tu - Tv|| \leq \beta||u - v||$$

for some $\beta < 1$, for every $u \in B$, $v \in B$.

We shall state the following theorem without proof. For a proof see Ross(1970, App.1).

(1.2) THEOREM: (Contraction Mapping Fixed Point Theorem) If $T:B \to B$ is a contraction mapping, then there exists a unique function $w \in B$, such that

$$Tw = w$$

Furthermore, for all $u \in B$

$$\lim_{n\to\infty} T^n u = w$$

We are now ready to prove the following theorem.

(1.3) THEOREM: Suppose $V = \min_{\pi \in M} V_\pi$. Then V is the unique solution of

(1.4) $$\alpha V(x) = \min_{\pi \in M} \{g_\pi(x) + A_\pi V(x)\}, \qquad x \in E$$

Proof: To be able to make the proof clearly, we write the equation (1.4) in an open form by using equation (3.5) from Chapter II and defining

$$y_s(x) = h(x) + c(s) \quad , \quad 0 \leq x < K$$
$$y_s(K) = h(K) + c(s) + \gamma\ell, \quad x = K$$

then it becomes

$$(1.5) \qquad V(x) = \underset{s \in M_x}{\text{Min}} \left\{\frac{1}{\alpha + \lambda(x) + s\mu}[y_s(x) + \lambda(x)V(x+1) + s\mu V(x-1)]\right\}$$

As you notice we combine the three equation in (3.5). We can do this, because we know that $\lambda(K) = 0$ then the term '$\lambda(x)V(x+1)$' drops in state K and satisfies the third equation in (3.5). Also for the first equation, we know that for $x = 0$, $M_o = \{0\}$. Then s can take only zero value and all the terms which consist s drop and it satisfies the first equation in (3.5).

So, let B be the Banach space of all bounded real valued functions on the state space E with the usual supremum norm. In other words, for any $f \in B$

$$||f|| = \sup_{i \in E}|f(i)|$$

Define an operator Γ mapping B into itself,

$$(1.6a) \qquad \Gamma f(x) = \underset{s \in M_x}{\text{Min}} \left\{\frac{1}{\alpha + \lambda(x) + s\mu}[y_s(x) + \lambda(x)f(x+1) + s\mu f(x-1)]\right\}, \quad 0 \leq x \leq K$$

Now, if we can show that Γ is a contraction mapping then it will follow that V is the unique solution to (1.4).

Let f and $g \in B$,

$$(1.6b)\qquad (\Gamma(f) - \Gamma(g))(x) = \inf_{s\varepsilon M_x} \{\frac{1}{\alpha + s\mu + \lambda(x)}[y_s(x) + \lambda(x)f(x+1) + s\mu f(x-1)\}]$$

$$- \inf_{s\varepsilon M_x} \frac{1}{\alpha + s\mu + \lambda(x)}[y_s(x) + \lambda(x)g(x+1) + s\mu g(x-1)]$$

Let s^* minimizes the second part of the right hand side of the equation (1.6b). It follows that,

$$(\Gamma(f) - \Gamma(g))(x) \leq \frac{1}{\alpha + s^*\mu + \lambda(x)}[\lambda(x)(f(x+1) - g(x+1)) + s^*\mu(f(x-1) - g(x-1))]$$

$$\leq \frac{1}{\alpha + s^*\mu + \lambda(x)}[\lambda(x)||f - g|| + s^*\mu||f - g||]$$

$$\leq \frac{\lambda(x) + s^*\mu}{\alpha + \lambda(x) + s^*\mu}||f - g||$$

Or we can write,

$$(\Gamma(f) - \Gamma(g))(x) \leq \underset{s\varepsilon\{0,..m\wedge x\}}{\text{Max}} \{\frac{\lambda(x) + s\mu}{\alpha + \lambda(x) + s\mu}\}\ ||f - g||$$

$$(1.7)\qquad (\Gamma(f) - \Gamma(g))(x) \leq \frac{\bar{\lambda} + m\mu}{\alpha + \bar{\lambda} + m\mu}\ ||f - g||$$

where $\bar{\lambda} = \max_x \lambda(x)$

Reversing f and g, and repeating the same procedure, we will get the same result.

(1.8) $$(\Gamma(g) - \Gamma(f))(x) \leq \frac{\bar{\lambda} + m\mu}{\alpha + \bar{\lambda} + m\mu} ||f - g||$$

Let $k = \frac{\bar{\lambda} + m\mu}{\alpha + \bar{\lambda} + m\mu} < 1$

Then combining (1.7) and (1.8), we obtain,

(1.9) $$||\Gamma f - \Gamma g|| \leq k||f - g||$$

Therefore from the definition (1.1), we can say that Γ is a contraction mapping. Then with respect to the theorem (1.2), there exists a unique function $V \in B$ such that

$$V = \Gamma V$$

which is the equation (1.5) itself.

By an optimal policy we mean the specification as to when the server should be opened or closed as a function of the number of customers in the system so as to minimize the cost function. The following theorem gives the characterization of the optimal policy.

(1.9) THEOREM: Let a policy $\pi^* \in M$ and corresponding V_{π^*} satisfies

$$V_{\pi^*}(x) = \underset{\pi \in M}{\text{Min}} \{g_\pi(x) + A_\pi V_{\pi^*}(x)\}$$

Then for all $x \in E$

$$V_{\pi^*}(x) = V(x) \leq V_\pi(x) \quad , \text{ for all } \pi \in M$$

and π^* is called as the optimal policy.

Proof: Let $\pi \in M$ be an arbitrary policy and assume V_{π^*} is given as above. Then $V_\pi \in D(A)$ and from equation (3.6). For $0 \leq x \leq K$,

$$\alpha[V_{\pi^*}(x) - V_\pi(x)] = \underset{\pi \in M}{\text{Min}}\{g_\pi(x) + A_\pi V_{\pi^*}(x)\} - \{g_\pi(x) + A_\pi V_\pi(x)\}$$

Using equation (3.5), we write the above equation in an open form. For $x = 0$

(1.10) $$\alpha[V_{\pi^*}(0) - V_\pi(0)] = \lambda(0)[V_{\pi^*}(1) - V_\pi(1)] - \lambda(0)[V_{\pi^*}(0) - V_\pi(0)]$$

For $0 < x < K$

(1.11) $$\alpha[(V_{\pi^*}(x) - V_\pi(x)] = \underset{s \in M_x}{\text{Min}}\{h(x) + c(s) + \lambda(x)V_{\pi^*}(x+1) \\ s\mu V_{\pi^*}(x-1) - (\lambda(x) + s\mu)V_{\pi^*}(x)\} \\ - \{h(x) + c(\pi(x)) + \lambda(x)V_\pi(x+1) \\ + \pi(x)\mu V_\pi(x-1) - (\lambda(x) + \pi(x)\mu)V_\pi(x)\}$$

Add the right hand side of the equation (1.11) the quantity

$$\pi(x)\mu(V_{\pi^*}(x-1) - V_{\pi^*}(x-1)) + (\lambda(x) + \pi(x)\mu)(V_{\pi^*}(x) - V_{\pi^*}(x$$

and rearranging it we obtain

$$(1.12)\quad \alpha[V_{\pi^*}(x) - V_{\pi}(x)] = \underset{s \varepsilon M_x}{\text{Min}} \{c(s) + s\mu V_{\pi^*}(x-1) - (\lambda(x) + s\mu) V_{\pi^*}(x)\}$$

$$-\{c(\pi(x)) + \pi(x)\mu V_{\pi^*}(x-1) - (\lambda(x) + \pi(x)\mu V_{\pi+}(x)\}$$

$$+ \lambda(x)(V_{\pi^*}(x+1) - V_{\pi}(x+1)) + \pi(x)\mu(V_{\pi^*}(x-1) - V_{\pi^*}(x-1))$$

$$- (\lambda(x) + \pi(x)\mu)(V_{\pi^*}(x) - V_{\pi}(x))$$

For $x = K$

$$(1.13)\quad \alpha[V_{\pi^*}(x) - V_{\pi}(x)] = \underset{s \varepsilon M_K}{\text{Min}} \{h(K) + c(s) + \gamma\ell + s\mu V_{\pi^*}(K-1) - s\mu V_{\pi^*}(K)\} - \{h(K) + c(\pi(K)) + \gamma\ell + \pi(K)\mu V_{\pi}(K-1) - \pi(K)\mu V_{\pi}(K)\}$$

Adding the quantity

$$\pi(K)\mu(V_{\pi^*}(K-1) - V_{\pi^*}(K-1)) + \pi(K)\mu(V_{\pi^*}(K) - V_{\pi^*}(K))$$

to the RHS of (1.13) and rearranging it, we obtain

$$(1.14)\quad \alpha[V_{\pi^*}(K) - V_{\pi}(K)] = \underset{s \varepsilon M_K}{\text{Min}} \{c(s) + s\mu V_{\pi^*}(K-1) - s\mu V_{\pi^*}(K)\}$$

$$-\{c(\pi(K)) + \pi(K)\mu V_{\pi^*}(K-1) - \pi(K)\mu V_{\pi^*}(K)\}$$

$$+ \pi(K)\mu[V_{\pi^*}(K-1) - V_{\pi}(K-1)]$$

$$- \pi(K)\mu[V_{\pi^*}(K) - V_{\pi}(K)]$$

Define,

$$u(x) = V_{\pi^*}(x) - V_{\pi}(x)$$

and

$$w(x) = \begin{cases} 0 & , \quad x = 0 \\ \underset{s \varepsilon M_x}{\text{Min}} \{c(s) + s\mu V_{\pi*}(x-1) - (\lambda(x) + s\mu)V_{\pi*}(x)\} \\ \quad - \{c(\pi(x)) + \pi(x)\mu V_{\pi*}(x-1) - (\lambda(x) + \pi(x)\mu)V_{\pi*}(x)\} & , \quad 0 < x < K \\ \underset{s \varepsilon M_K}{\text{Min}} \{c(s) + s\mu V_{\pi*}(K-1) - s\mu V_{\pi*}(K)\} \\ \quad - \{c(\pi(x) + \pi(x)\mu V_{\pi*}(K-1) - \pi(x)\mu V_{\pi*}(K)\} & , \quad x = K \end{cases}$$

It is now clear that $w \leq 0$, $w \varepsilon R(A)$ and $u \varepsilon D(A)$ and in particular combination of (1.10), (1.12), (1.14) gives

$$\alpha u = w + A_{\pi} u$$

It follows from the theorem (II.3.1)

$$u(x) = E_x[\int_0^{\infty} e^{-\alpha t} w(X_t) dt]$$

where X is the queue process obtained by using $\pi \varepsilon M$ as a control. Therefore $w \leq 0$ implies that $u \leq 0$, then

$$V_{\pi*}(x) \leq V_{\pi}(x) \qquad \text{for all} \qquad \pi \varepsilon M$$

From our definition of the optimal control problem in equation (1.4)

$$V_{\pi*}(x) = V(x) \ , \quad \text{for all} \qquad x \varepsilon E$$

Since π^* selects the action (server number) minimizing the right hand side of (1.4) in each state, then π^* is the optimal policy.

As a result of theorem (1.3) we can show that the unique optimal solution V is bounded in the following Lemmas.

(1.15) LEMMA: Optimal solution is nonnegative for all possible states. That is,

$$V(x) \geq 0 \quad , \quad \text{for all} \quad x \in E$$

Proof: To prove this lemma it is sufficient to show that for any function $f \in B$; if $f \geq 0$ then $\Gamma f \geq 0$ also holds. But this is trivially true from the definition of Γf in (1.6a). Function $y_s(x) \geq 0$, since h and c functions are assumed to be nonnegative. So, given $f \geq 0$ implies $\Gamma f \geq 0$. Therefore, from the properties of contraction mapping,

$$V(x) \geq 0 \quad , \quad \text{for all} \quad x \in E$$

(1.16) LEMMA: There is an upper bound for the optimal solution, such that

$$V(x) \leq \frac{\bar{h} + c(0) + \gamma \ell}{\alpha} \quad , \quad x \in E$$

where $\bar{h} = \max_x h(x)$.

Proof: If we can show that for any $f \in B$ and $f \leq (\bar{h} + c(0) + \gamma\ell)/\alpha$, Γf is also less than this quantity, then the proof is completed. Assume $f \leq (\bar{h} + c(0) + \gamma\ell)/\alpha$

and put this value to the definition of Γf in (1.6a) and take s = 0

$$\Gamma f(x) \le \frac{1}{\alpha + \lambda(x)}[h(x) + c(0) + \lambda(x)\,\frac{(\bar{h} + c(0) + \gamma\ell)}{\alpha}]$$

$$\Gamma f(x) \le \frac{1}{\alpha + \lambda(x)}[\bar{h} + c(0) + \lambda(x)\,\frac{(\bar{h} + c(0) + \gamma\ell)}{\alpha}]$$

$$\Gamma f(x) \le \frac{\bar{h} + c(0) + \gamma\ell}{\alpha}$$

Lemma (1.15) and (1.16) show that V has an upper and lower bound, such that

$$0 \le V(x) \le \frac{\bar{h} + \gamma\ell + c(0)}{\alpha}$$

The upper bound can be considered as the worst case in the system; it denotes no server, so the lost profit and shortage cost incurres all the time.

III.2 Successive Approximation Algorithm

In this section we present an algorithm by which the optimal policy actually can be obtained. Although it was shown that optimal policy exists in M when it satisfies the equation (1.9), the problem of finding optimal policies is nontrivial. The number of policies in M may be astronomically large. For example if E contains N states and if the possible actions for each state is 2, then M contains 2^N different policies. For very small values of N and for small action space the method of simple enumeration is feasible; however for large N, complete enumeration is virtually impossible.

Here we present a version of the successive approximation method originally suggested by Derman(1970). This is one of the classical methods used in solving differential and integral equations. In itself it does not provide a method for obtaining a solution in a finite number of iterations; however slightly modified it can.

Now, we describe this algorithm and prove that it successively iterates to the minimum expected discounted cost. We also seek to establish the conditions under which this procedure converges to the optimal policy.

Before presenting the algorithm, we shall state the transformation which is the core of the procedure. We define a sequence $\{V_n;\ n \geq 0\}$ by

(2.1) $$V_{n+1} = \Gamma V_n$$

That is, for $0 \leq x \leq K$

(2.2) $$V_{n+1}(x) = \underset{s \in M_x}{\text{Min}} \left\{ \frac{1}{\alpha + \lambda(x) + s\mu} \left[y_s(x) + \lambda(x) V_n(x+1) + s\mu V_n(x-1) \right] \right\}$$

And also corresponding policies generate a sequence $\{\pi_n;\ n \geq 0\}$. In sequence n, for the state x, the s value which minimizes the right hand side of the (2.2) is put in to the $\pi_n(x)$.

ALGORITHM.

STEP 1: INITIALIZATION. Set $n = 0$. Take an arbitrary policy $\pi_o \varepsilon M$ and find the corresponding values V_o from the equation (3.5) in Chapter II. (For the simplicity of computation the initial policy π_o can be taken as $\pi_o(x) = 0$, for all $x \varepsilon E$; i.e. use zero server in each state. At this time some of the terms in equation (3.5) drop and the solution can easily be obtained starting from state K. Since this policy is in M, there is no problem of choosing it. If we choose an arbitrary policy other than this one; then we have to solve a system of linear equations.)

STEP 2: TRANSFORMATION. Calculate V_{n+1} by using the transformation

$$V_{n+1} = \Gamma V_n$$

and find the corresponding policy π_{n+1}.

STEP 3: TERMINATION. If $|V - V_n| < \varepsilon$ terminate and π_{n+1} is the ε-optimal policy. Otherwise go to step 2.

The following theorem shows that in each iteration the transformed values are decreasing in each sequence. And then we show that in the limit this sequence reaches the optimal value.

(2.3) THEOREM: $V_{n+1}(x) \leq V_n(x)$ for all n, $x \varepsilon E$.

Proof: Here we will use the induction method. Taking n = 1, we shall see first if $V_1 \leq V_o$.

For x = 0, utilizing equation (2.2)

$$V_1(0) = \frac{1}{\alpha + \lambda(0)} [h(0) + c(0) + \lambda(0)V_o(1)]$$

which turns out to be equal to $V_o(0)$ where $V_o(x)$ is defined as the unique solution to equation (3.5) when using the initial policy $\pi_o(x) = 0$.

For $0 < x < K$, taking s = 0 in equation (2.2) we obtain

$$V_1(x) \leq \frac{1}{\alpha + \lambda(x)} [h(x) + c(0) + \lambda(x)V_o(x+1)]$$

For x = K,

$$V_1(K) \leq \frac{1}{\alpha}[h(K) + c(0) + \gamma\ell]$$

The right hand side of the above inequalities is equal to the $V_o(x)$ and $V_o(K)$ respectively. So, combining these three results we end up with

$$V_1(x) \leq V_o(x) \quad , \quad \text{for all} \quad x \varepsilon E.$$

Now assuming $V_{n+1}(x) \leq V_n(x)$, for all $x \varepsilon E$, we can show that

$$V_{n+2}(x) \leq V_{n+1}(x)$$

Using transformation (2.2), for $0 \leq x \leq K$,

$$V_{n+2}(x) = \Gamma V_{n+1}(x) = \underset{s \in M_x}{\text{Min}} \{\frac{1}{\alpha + \lambda(x) + s\mu}[y_s(x) + \lambda(x)V_{n+1}(x+1) + s\mu V_{n+1}(x-1)]\}$$

$$V_{n+1}(x) = \Gamma V_n(x) = \underset{s \in M_x}{\text{Min}} \{\frac{1}{\alpha + \lambda(x) + s\mu}[y_s(x) + \lambda(x)V_n(x+1) + s\mu V_n(x-1)]\}$$

Since we assume that $V_{n+1}(x) \leq V_n(x)$, comparison of the right hand side of the two equations above justifies that $V_{n+2}(x) \leq V_{n+1}(x)$. Therefore, by mathematical induction, it is true for all n.

(2.4) LEMMA. $\lim_{n \to \infty} V_n(x) = V(x)$, for all $x \in E$.

Proof: Since the sequence V_n is defined as in (2.1) and since we proved that Γ is a contraction mapping then it is obvious from the theorem (1.2) that for all $V_n \in B$,

$$\lim_{n \to \infty} \Gamma V_n = V$$

Since V_n is the solution to (2.2) then $V_n \in D(A)$ where $D(A) \equiv B$.

Up to here, we proved the convergence of the V_n values to the minimum cost. But this does not imply the convergence of the policies to the optimum policy. To prove the convergence of the policies, we define another sequence $\{U_n; n \geq 0\}$ by

(2.5) $\alpha U_n(x) = g_{\pi_n}(x) + A_{\pi_n} U_n(x)$, for all $x \varepsilon E$

where $g_{\pi_n}(x)$ is defined in (II.3.2) and (II.3.3) and π_n is the obtained policy in sequence n. In the sense that

$$U_n(x) = V_{\pi_n}(x)$$

where V_{π_n} is expected discounted cost for policy π_n.

(2.6) LEMMA: $U_n(x) \leq V_n(x)$; for all n, $x \varepsilon E$.

Proof: Take π_n as the obtained policy for sequence n and subtract equation (2.5) from (2.2) and write in an open form.

$$V_n(0) - U_n(0) = \frac{1}{\alpha + \lambda(0)}\{\lambda(0)[V_{n-1}(1) - U_n(1)]\}$$

For $0 < x < K$,

$$V_n(x) - U_n(x) = \frac{1}{\alpha + \lambda(x) + \pi_n(x)\mu}\{\lambda(x)[V_{n-1}(x+1) - U_n(x+1)] + \pi_n(x)\mu[V_{n-1}(x-1) - U_n(x-1)]\}$$

$$V_n(K) - U_n(K) = \frac{1}{\alpha + \pi_n(K)}\{\pi_n(K)[V_{n-1}(K-1) - U_n(K-1)]\}$$

Add the following quantities to the above equations,

First one: $\lambda(0)[V_n(1) - V_n(1)]$

Second one: $\lambda(x)[V_n(x+1) - V_n(x+1)] + \pi_n(x)\mu[V_n(x-1) - V_n(x-1)]$

Third one: $\pi_n(K)\mu[V_n(K-1) - V_n(K-1)]$

Rearranging the terms, we obtain

$$(2.7)\qquad V_n(0) - U_n(0) = \frac{1}{\alpha + \lambda(0)}\{\lambda(0)[V_{n-1}(1) - V_n(1)] + \lambda(0)[V_n(1) - U_n(1)]\}$$

$$(2.8)\qquad V_n(x) - U_n(x) = \frac{1}{\alpha + \lambda(x) + \pi_n(x)\mu}\{\pi_n(x)\mu[V_n(x-1) - U_n(x-1)] + \pi_n(x)\mu[V_{n-1}(x-1) - V_n(x-1)] + \lambda(x)[V_n(x+1) - U_n(x+1)] + \lambda(x)[V_{n-1}(x+1) - V_n(x+1)]\}$$

$$(2.9)\qquad V_n(K) - U_n(K) = \frac{1}{\alpha + \pi_n(K)\mu}\{\pi_n(K)\mu[V_{n-1}(K-1) - V_n(K-1)] + \pi_n(K)\mu[V_n(K-1) - U_n(K-1)]\}$$

Define,

$$y(x) = V_n(x) - U_n(x) \quad , \quad \text{for all } x$$

and

$$w(0) = \lambda(0)[V_{n-1}(1) - V_n(1)]$$

$$w(x) = \lambda(x)[V_{n-1}(x+1) - V_n(x+1)] + \pi_n(x)\mu[V_{n-1}(x-1) - V_n(x-1)]$$

$$w(K) = \pi_n(x)\mu[V_{n-1}(K-1) - V_n(K-1)]$$

After these definitions; equation (2.7), (2.8) and (2.9) become

$$y(0) = \frac{1}{\alpha + \lambda(0)}[w(0) + \lambda(0)y(1)]$$

For $0 < x < K$

$$y(x) = \frac{1}{\alpha + \lambda(x) + \pi_n(x)\mu}[w(x) + \lambda(x)y(x+1) + \pi_{n+1}(x)\mu y(x-1)]$$

$$y(K) = \frac{1}{\alpha + \pi_n(K)\mu}[w(K) + \pi_n(K)\mu y(K-1)]$$

By previous result (theorem (3.1) in Chapter II) we have,

$$y(x) = E_x[\int_0^\infty e^{-\alpha t} w(X_t)dt]$$

where X_t is the queue process obtained when policy π_n is used. Since we showed in theorem (2.3) that $V_n \leq V_{n-1}$, which implies $w(x) \geq 0$, for all $x \varepsilon E$. Then this justifies that $y(x) \geq 0$, i.e.,

$$V_n(x) \geq U_n(x) \qquad , \qquad x \varepsilon E$$

(2.10) LEMMA: $V(x) \leq U_n(x) \qquad , \qquad x \varepsilon E$

<u>Proof:</u> It is obvious that $V(x) \leq U_n(x)$. Because, U_n is obtained by using a policy π_n, but only the optimal policy π^* gives V.

Lemma (2.6) and (2.10) together show that the policies in each iteration converges to the optimal policy at the end.

Theorem (2.3) shows that in each sequence the value function iterates. But for the sequence $\{U_n; n \geq 0\}$ this is not true. For the same policies π_n and π_{n+1} we obtain $U_n(x) = U_{n+1}(x)$ for all $x \in E$. But this equality does not imply that these policies are optimal, because from the characteristic of successive approximation algorithm we can obtain the same policy in the preceeding iterations and some steps later it could change.

In the successive approximation method the limiting function will satisfy equation (1.4) and the optimal policy will be obtained. In practice, the limiting function will only be approximated. However, in order to have an approximation close enough to V, a large number of iterations are required. This method does not have a specific stopping criterion. So based on your problem you can modify a stopping rule and you end up with ε-optimal policy. In the algorithm we defined the stopping rule as $|V - V_n| < \varepsilon$ for any epsilon which will be specified according to the nature of the problem.

(2.11) LEMMA: For any $\varepsilon > 0$, $n \geq n(\varepsilon) = \ln(\frac{\varepsilon(1-K)}{C})/\ln k$ implies

$$||V - V_n|| \leq \varepsilon$$

then $\pi_{n(\varepsilon)}$ is ε-optimal policy.

<u>Proof:</u> From the definition of contraction mapping in (1.1), we know

$$||\Gamma V_n - \Gamma V_{n-1}|| \leq k||V_n - V_{n-1}||$$

where k is defined in (1.8) as

$$k = \frac{\bar{\lambda} + m\bar{\mu}}{\alpha + \bar{\lambda} + m\mu}$$

Using the transformation in (2.1), the above equation turns out to be,

$$||V_{n+1} - V_{n-1}|| \leq k||V_n - V_{n-1}||$$

Then

$$||V_{n+1} - V_n|| \leq k^n||V_1 - V_o||$$

From Lemma (1.16) and (1.15)

$$||V_{n+1} - V_n|| \leq k^n \frac{\bar{h} + \gamma\ell + c(0)}{\alpha}$$

Let $$C = \frac{\bar{h} + \gamma\ell + c(0)}{\alpha}$$

So

$$||V_{n+1} - V_n|| \leq k^n.C$$

Also

$$||V_{n+m} - V_n|| \leq ||V_{n+m} - V_{n+m-1}|| + \ldots\ldots$$

$$+ ||V_{n+2} - V_{n+1}|| + ||V_{n+1} - V_n||$$

$$||V_{n+m} - V_n|| \leq \sum_{i=n}^{n+m} k^i.C$$

$$||V_{n+m} - V_n|| \leq k^n \sum_{j=o}^{m} k^j.C$$

Then we could write

$$||V - V_n|| = \lim_{n\to\infty} ||V_{n+m} - V_n|| \leq k^n \sum_{j=o}^{\infty} k^j.C$$

Then

$$||V - V_n|| \leq \frac{k^n}{1 - k} C$$

If an ε-optimal policy is desired

$$||V - V_n|| \leq \varepsilon$$

Therefore

$$\frac{k^n}{1 - k} C \leq \varepsilon$$

(2.12) $n \geq \ln(\frac{(1 - k)\varepsilon}{C})/\ln k$

So

$$n(\varepsilon) = \inf\{n \geq 0 : n \geq \ln(\frac{(1 - k)\varepsilon}{C})/\ln k\}$$

From Lemma (2.6) we could write

$$||U_{n(\varepsilon)} - V|| \le ||V_{n(\varepsilon)} - V|| \le \varepsilon$$

Since $\pi_{n(\varepsilon)}$ is obtained from the transformations of $V_{n(\varepsilon)}$ and since $U_{n(\varepsilon)}$ is obtained by using $\pi_{n(\varepsilon)}$ policy, we could say that for all $n(\varepsilon)$ satisfying equation (2.12) $\pi_{n(\varepsilon)}$ is ε-optimal policy.

III.3 Policy Improvement Algorithm

In this section we shall give another algorithm which is given by Doshi (1976). This is a version of policy improvement algorithm. It finds the optimum policy for the equation (1.4) and proves the existence of the solution.

Doshi describes his algorithm and proves that it generates an improving sequence of stationary policies. He assumes finite action space and states that the existence of a solution to the functional equations and of an optimal policy cannot be established directly using this algorithm when the action space is not finite.

In the following we give his algorithm adopting to our case.

ALGORITHM: Given a policy $\pi^{o} \varepsilon M$ we generate a sequence $\{\pi^{n}; n \ge 0\}$ of policies in M by the policy improvement algorithm.

STEP 1: After finding $\pi^n \varepsilon M$ we obtain the expected discounted return function for π^n from the value determination equations

$$V_{\pi^n}(x) = g_{\pi^n}(x) + A_{\pi^n} V_{\pi^n}(x) \quad , \quad x \varepsilon E$$

If $n \geq 2$ and $V_{\pi^n}(x) = V_{\pi^{n-1}}(x)$ for all $x \varepsilon E$, then we terminate the algorithm and conclude that π^n and π^{n-1} are optimal in M. Otherwise proceed to step 2.

STEP 2: A policy π^{n+1} is defined as

$$\pi^{n+1}(x) = S_x \quad , \quad x \varepsilon E$$

where for each $x \varepsilon E$

$$\frac{1}{\alpha + \lambda(x) + S_x\mu}[y_{S_x}(x) + \lambda(x)V_{\pi^n}(x+1) + S_x\mu V_{\pi^n}(x-1)] =$$

$$\min_{S \varepsilon M_x} \left\{ \frac{1}{\alpha + \lambda(x) + s\mu} [y_S(x) + \lambda(x)V_{\pi^n}(x+1) + S\mu V_{\pi^n}(x-1)] \right\}$$

Go back to step 1.

Policy improvement algorithm is an iterative procedure that improves on each iteration and terminates after a finite number of iterations with an optimal policy. As you notice in the iteration cycle it has two steps: (i) Value determination operation, (ii) Policy-improvement routine. The first step yields values as a function of policy whereas the second one yields the policy as a function of the values.

The properties of the policy-improvement algorithm can be described as follows,

i) The solution of the decision process is just solving sets of simultaneous linear equations,

ii) Each succeeding policy found in the iteration cycle has a cost smaller than the previous one, i.e., $V_{\pi^{n+1}} < V_{\pi^n}$

iii) The iteration cycle has a specific stopping rule such that the optimal policy is reached when the policies on two successive iterations are identical; it will usually find this policy in a small number of iterations.

Since M contains only a finite number of policies and since each iteration is accompanied by a strict improvement, no repetitions will occur. This method finds the policy that has a smaller average return per transition than any other policy under consideration. Thus at some point no improvements will be possible, then the procedure terminates after a finite number of iterations.

In the next chapter, we shall solve a simple example with these two algorithms and make a comparison between them.

IV. APPLICATIONS AND EXTENSIONS

This chapter extends our original single station system to some other queue models. First the series queues are considered and the corresponding generator and its range and domain is found. Secondly we take our original system to find the optimum service rate from the given set. In the third model, not allowing the queue formation in front of the stations we formalize the problem of optimum service rate selection in series queues.

Then in Section 3 we give an example problem and solve it by computer with the given algorithms. Examining the results we make an analysis of the algorithms.

IV.1 Examples of Actual Queueing Systems

There are many well-known, common areas of application for the queueing theory. Let us just briefly mention some examples of real queueing systems. Generally, real-life systems do not lend themselves for operating characteristics found in a standard textbook on queues. However, one sometimes is lucky enough to find a system which behaves like the textbook models. Such is the case in drive-in banking. For example: A local bank wanted to expand their drive in facilities. Two options were basically available for expansion: Teller stations or Robo-window stations. Since the space is limited up to a teller stations could be used where as the number of robo-stations was limited to b. Upon discussion, the prime consideration was lost customer due to poor service and the cost of new servers.

Another example of queueing network theory being applied is the transportation system. Consider an air terminal design problem. It is carried out by analysing the flow of passengers through terminal and finding the number of necessary person in the necessary areas to cope with the range of flows encountered in all the component areas. The flow of passengers can be described as a series of linked queueing model.

Examples for the applications of queueing theory to the health care systems can be categorized as: (i) appointment systems, (ii) determination of the optimum staffing level, (iii) determination of the number of patients in a hospital. The second case is relevant to our thesis. Consider the messanger unit at a hospital. The function of it is to transport patients, specimens, reports and miscellaneous objects in response to request from any section of the hospital whenever a call is received, the dispatcher sends a messenger, if one is available, to provide a service. If no messenger is free calls must wait and may thus accumulate, forming a queue. The question to be answered here is: how large a staff is required to give adequate service from a hospital messenger unit? Due to the difficulty of estimating the cost of a waiting call the variation of the ratio of the cost of waiting per call per hour to the cost of service per messenger per hour.

Most of the studies in the optimum control of queueing processes are applied to the maintenance problems. The basic question in this area is to determine the optimum number of a repair or servicing crew for a given number of machines. Machines break down from time to time needing the service of repair crew to put it back in running order. Machines are assigned to operators with the objective of minimizing an expected cost model of the queueing system.

IV.2 Extensions

This section extends the original problem to some other queueing systems. First we consider the optimization properties in the series queues. Then we show the similarity of the problem which finds the optimal service rate to minimize the infinite horizon expected discounted cost, with the original problem that finds the optimum server number. Thirdly we will analyze the case which does not allow queue formation in the series queues.

IV.2.1 Optimization in The Tandem Queues

In this section we consider models in which there exits a series of stations which each calling unit must visit successively prior to leaving the system. Some examples of such tandem queueing situations (sometimes referred to as series queues) are manufacturing or assembly line processes in which units must proceed through a series of work stations, each station preforming a given task, or a clinic physical examination procedure where the patient goes through a series of stages.

The tandem model to be considered here is composed of two service station with limited waiting room capacities K_1 and K_2 respectively. Such a situation is pictured in Figure 3. We further assume that the customers arrive according to a Poisson process with mean λ, and the service time of each server at station i (i = 1,2) is exponential with mean $1/\mu_i$. Maximum number of servers that can work in each station is m_i.

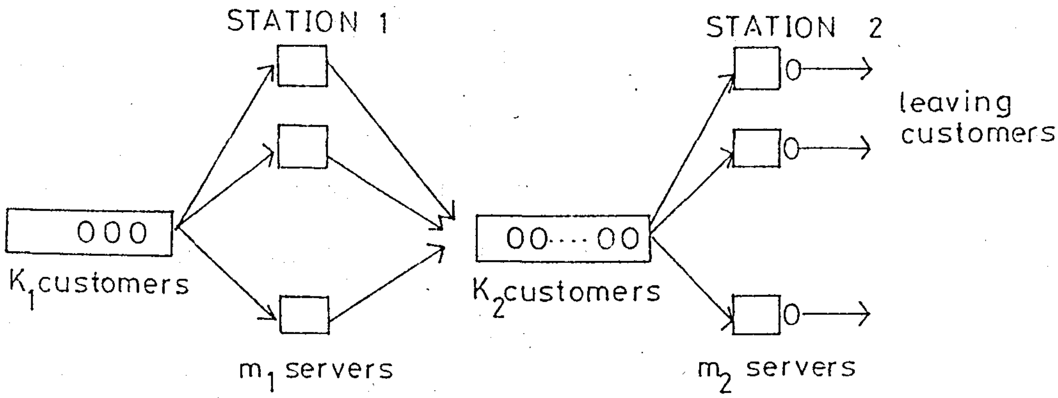

Fig. 3 - Series queue, finite waiting room

Walrand(1980) claims that he could not find any study of optimal policies when there are two or more connected service stations. In our literature survey, we have come across some studies about series queues; but not in their optimization generally in the distribution of their output processes. Walrand analyzes the case where customers in a Poisson stream enter a network consisting of two exponential servers in tandem. The service rate $u \in [0,a]$ at station 1 is to be selected as a function of the state (x_1,x_2) and the service rate at station 2 is constant with μ. He wants to minimize the expected total discounted cost corresponding to the instantaneous cost $c_1x_1 + c_2x_2$. He first formalizes the problem as a dynamic programming and then constructs an equivalent linear programming problem to prove the convexity. He finds that the optimal policy is of the form $u = a$ or $u = 0$ according as $x_1 < S(x_2)$ and $x_1 \geq S(x_2)$ and S is a switching function.

Our case is very different from Walrand. The service rate in a station is changing when we increase or decrease the server number. As you notice Walrand takes constant service rate in the second station, and his cost structure is a simple linear cost. In our system server number can change in each station and our cost structure is more general.

The state of the system at time t is defined as $X_t = (X_t^1, X_t^2)$ where X_t^i is the number of customers at time t and the state space is $E = E_1 x E_2$ where $E_1 = \{0,1,\ldots,K_1\}$ and $E_2 = \{0,1,2,\ldots,K_2\}$. At any time t, the designer determines the number of servers to be employed (S_t) in each station by knowing the state of the system. Then S_t is a function of X_t, i.e. $S_t = (S_t^1, S_t^2) = \pi(X_t^1, X_t^2)$ where $\pi(.,.)$ is the control function which gives the number of servers in each station and $\pi(x_1, x_2) \in M = M_1 x M_2$ where $M_1 = \{0,1,\ldots,x_1 \wedge m_1\}$, $M_2 = \{0,1,\ldots,x_2 \wedge m_2\}$. In this case admissible policy set can be defined as,

$$M = (\pi : \{0,1,\ldots,K_1\} x \{0,1,\ldots,K_2\} \rightarrow \{0,1,\ldots,m_1\} x \{0,1,\ldots,m$$

or simply,

$$M = (\pi : E \rightarrow M)$$

The first station is an $M/M/S_1/K_1$ model. It is necessary to know the output distribution of the first station in order to find the input distribution to the second station. Early studies show that the departure time distribution from an $M/M/S_1/K_1$ queue is identical to the interarrival time distribution. So the second station is an $M/M/S_2/K_2$ model.

Now, our aim is to find an optimum policy which minimizes infinite horizon expected discounted cost. The cost structure of the system is described as following:

i) $h(x^1,x^2)$ - cost of holding customers at the stations.

ii) $C(\pi(x^1,x^2))$ - server cost in the stations.

iii) $\lambda\ell$ - shortage cost at station 1,

where λ is the mean arrival rate and ℓ is the lost profit from lost customers. When the queue capacity of the first station is completely full in the first station, then new arrivals will not enter and the system will have lost profit from these lost customers. The cost appears only at the states (K_1,x^2) where $0 \leq x^2 \leq K_2$. We have not this type of cost in the second station because when a customer comes to the system, after his service completion in station 1, he has to take service from station 2 necessarily.

Define for any policy $\pi \in M$, the rate of cost at time t as,

(2.1) $g_\pi(X_t) = h(x^1,x^2) + C(\pi(x^1,x^2))$

For the states of (K_1,x_2), $0 \leq x_2 \leq K_2$

$$(2.2) \qquad g_\pi(K_1,x_2) = h(K_1,x^2) + c(K_1,x^2)) + \gamma\ell$$

Then expected total discounted cost for any policy $\pi \varepsilon M$, given the initial state is $x = (x^1,x^2)$

$$V_\pi(x) = E_x[\int_0^\infty e^{-\alpha t} g_\pi(X_t)dt] \quad , \quad x \varepsilon E$$

Define the minimum expected discounted cost function as,

$$V(x) = \min_{\pi \varepsilon M} V_\pi(x) \quad , \quad x \varepsilon E.$$

Now, this problem turns out to be the same one of the original problem. So, if we find the generator of the process and write the functional equations, then we can use the algorithms defined in Chapter III.

This process is also a 'Birth and Death Process'. Because, we are continuously observing the system, there could only be one change at one time. One can arrive to the station 1 or the service of a customer can be completed in station one and he joins the second queue or a service completion occurs at station 2. Let us use the definition (II.2.1), to find the generator of the process and also the range and the domain of it.

Define for $x = (x_1,x_2)$ where $0 < x_1 < K_1$, $0 < x_2 < K_2$

$$P_t f(x) = P_t f(x_1,x_2) = E_x[f(X_t^1,X_t^2)]$$

$$(2.3) \qquad P_t f(x) = E_x[f(X_t^1, X_t^2) 1_{\{T \le t\}}] + E_x[f(X_t^1, X_t^2) 1_{\{T > t\}}]$$

where T is defined as the time of the first change, i.e.,

$$T = \inf\{t \ge 0 : X_t \ne X_o\} = \inf\{A, D, C\}$$

where A is the time of the first arrival to station 1, D is the time of the first departure from station 1 and C is the time of the first departure from station 2. Then define for arbitrary policy $\pi \in M$

$$\pi(x_1, x_2) \doteq \pi(\pi_1(x_1), \pi_2(x_2))$$

Let us find the probability of no change in the state

$$(2.4) \qquad E_x[f(X_t^1, X_t^2) \cdot 1_{\{T>t\}}] = E_x[f(x_1, x_2) \cdot 1_{\{T>t\}}]$$

$$= f(x_1, x_2) P_x\{T > t\}$$

$$= f(x_1, x_2) e^{-(\lambda + \pi_1(x_1)\mu_1 + \pi_2(x_2)\mu_2)t}$$

$$(2.5) \qquad E_x[f(X_t^1, X_t^2) \cdot 1_{\{T \le t\}}] = E_x[f(X_t^1, X_t^2) \cdot 1_{\{T=A\}} \cdot 1_{\{T \le t\}}]$$

$$+ E_x[f(X_t^1, X_t^2) \cdot 1_{\{T=D\}} \cdot 1_{\{T \le t\}}]$$

$$+ E_x[f(X_t^1, X_t^2) \cdot 1_{\{T=C\}} \cdot 1_{\{T \le t\}}]$$

$$= a_1(x,t) + a_2(x,t) + a_3(x,t)$$

$$(2.6) \qquad a_1(x,t) = E_x[f(X_t^1, X_t^2) \cdot 1_{\{T \le t\}} \cdot 1_{\{T=A\}} | T]$$

$$= E_x[1_{\{A \le t\}} \cdot 1_{\{T=A\}} E_{(x_1+1, x_2)}[f(X_{t-T}^1, X_{t-T}^2)]]$$

$$= E_x[1_{\{A\leq t\}}\cdot 1_{\{T=A\}}\cdot P_{t-T}f(x_1+1,x_2)]$$

$$= E_x[1_{\{A\leq t\}}\cdot 1_{\{A<D\}}\cdot 1_{\{A<C\}}\cdot P_{t-T}f(x_1+1,x_2)]$$

If we put a condition on A to the above equation, we obtain

$$a_1(x,t) = E_x[1_{\{A\leq t\}}\cdot P_{t-A}f(x_1+1,x_2)\,.e^{-(\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)A}]$$

$$= \int_0^t \lambda\, e^{-\lambda u}\, P_{t-u}f(x_1+1,x_2)\,.e^{-(\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)u}\,du$$

Make a change of variables by setting $w = t-u$,

(2.7) $$a_1(x,t) = \int_0^t \lambda e^{-\lambda(t-w)}\,.e^{-(\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)(t-w)}\,.P_w f(x_1+1,x_2)dw$$

Following the same procedure,

(2.8) $$a_2(x,t) = E_x[1_{\{D\leq t\}}\cdot P_{t-D}f(x_1-1,x_2+1)\,.e^{-(\lambda+\pi_2(x_2)\mu_2)D}]$$

$$= \int_0^t \pi_1(x_1)\mu_1.e^{-(\pi_1(x_1)\mu_1(t-w)}\,.e^{-(\lambda+\pi_2(x_2)\mu_2)(t-w)}$$

$$.\ P_w f(x_1-1,x_2\ 1)dw$$

(2.9) $$a_3(x,t) = E_x[1_{\{C\leq t\}}\cdot P_{t-C}f(x_1,x_2-1)\,.e^{-(\lambda+\pi_1(x_1)\mu_1)C}]$$

$$= \int_0^t \pi_2(x_2)\mu_2.e^{-(\pi_2(x_2)\mu_2)(t-w)}\,.e^{-(\lambda+\pi_1(x_1)\mu_1)(t-w)}$$

$$.\ P_w f(x_1,x_2-1)dw$$

It is clear from (2.7), (2.8) and (2.9) that for all $x \in E$

$$\lim_{t\downarrow 0} a_1(x,t) = \lim_{t\downarrow 0} a_2(x,t) = \lim_{t\downarrow 0} a_3(x,t) = 0$$

The only left term in $P_t f(x_1,x_2)$ is $f(x_1,x_2)e^{-(\lambda+\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)t}$ from equation (2.4). It is obvious that this term goes to $f(x_1,x_2)$ when $t\downarrow 0$. Therefore from the definition the range of the generator consists the set of all bounded measurable functions on E.

From the definition,

$$(2.10) \quad A_\pi f(x_1,x_2) = \lim_{t\downarrow 0} \frac{1}{t}[P_t f(x_1,x_2) - f(x_1,x_2)], \quad f \in R(A_\pi)$$

where the domain $D(A_\pi)$ is the set of all $f \in R(A_\pi)$ for which this limit exits boundedly pointwise and belongs to $R(A_\pi)$. It follows from (1.10) that

$$(2.11) \quad \lim_{t\downarrow 0} \frac{1}{t}[P_t f(x_1,x_2) - f(x_1,x_2)] = \lim_{t\downarrow 0}\{\frac{1}{t}[f(x_1,x_2) \cdot e^{-(\lambda+\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)t} - f(x_1,x_2)] + \frac{1}{t} a_1(x,t) + \frac{1}{t} a_2(x,t) + \frac{1}{t} a_3(x,t)\}$$

It is obvious that for $f \in R(A)$

$$(2.12) \quad \lim_{t\downarrow 0} a_1(x,t) = f(x_1+1,x_2)$$

$$(2.13) \quad \lim_{t\downarrow 0} a_2(x,t) = \pi_1(x_1)\mu_1 f(x_1-1,x_2+1)$$

$$(2.14) \quad \lim_{t\downarrow 0} a_3(x,t) = \pi_2(x_2)\mu_2 f(x_1,x_2-1)$$

Also from the definition (2.11) in Chapter II

$$\frac{1}{t}|a_1(x,t)| \leq \lambda||f|| \frac{1}{t}\int_0^t e^{-(\lambda+\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)z}\,dz$$

$$\leq \lambda||f|| \frac{1}{t}$$

$$\frac{1}{t}|a_2(x,t)| \leq \pi_1(x_1)\mu_1||f|| \frac{1}{t}\int_0^t e^{-(\lambda+\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)z}\,dz$$

$$\leq m_1\mu_1||f|| \frac{1}{t}$$

$$\frac{1}{t}|a_3(x,t)| \leq \pi_2(x_2)\mu_2||f|| \frac{1}{t}\int_0^t e^{-(\lambda+\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)z}\,dz$$

$$\leq m_2\mu_2||f|| \frac{1}{t}$$

Since $m_1\mu_1$, $m_2\mu_2$ and λ are finite, then it shows that $a_1(x,t)$, $a_2(x,t)$ and $a_3(x,t)$ converges boundedly pointwise. Therefore $f \in R(A)$ is in $D(A)$ if and only if the left term in equation (2.11);

$$\frac{1}{t}[f(x_1,x_2)e^{-(\lambda+\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2)t} - f(x_1,x_2)]$$

converges boundedly pointwise as $t\downarrow 0$. We counld write it as,

$$\lim_{t\downarrow 0} \frac{1}{t}\{f(x_1,x_2)[1 - (\lambda+\pi_1(x_1)\mu_1 + \pi_2(x_2)\mu_2)t + 0(t)] - f(x_1,x_2)\}$$

where $\lim_{t\downarrow 0} (0(t))/t = 0$.

Now,

$$\lim_{t \downarrow 0} \frac{1}{t}\{[-\lambda - \pi_1(x_1)\mu_1 - \pi_2(x_2)\mu_2]t + 0(t)\}f(x_1,x_2)$$

$$(2.15) \quad = -(\lambda + \pi_1(x_1)\mu_1 + \pi_2(x_2)\mu_2)f(x_1,x_2)$$

This result with the others imply that the domain of A_π is also the set of bounded functions defined on E.

Therefore putting equations (2.12), (2.13), (2.14) and (2.15) into the equation (2.10) we end up with the generator. We shall give the generator of a two-series-connected-station system in general. In the following think the state of generator as $A((i,j);\ (k,\ell))$ for any policy $\pi(x_1,x_2) = (\pi_1(x_1),\pi_2(x_2))$

a) If $i = 0,\ j = 0$

$$A_\pi((i,j);\ (k,\ell)) = \begin{cases} \lambda, & k = i+1, \quad \ell = j \\ -\lambda, & k = i \quad , \quad \ell = j \end{cases}$$

b) If $i = 0,\ j \neq 0$

$$A_\pi((i,j);\ (k,\ell)) = \begin{cases} \lambda & , k = i+1 \quad \ell = j \\ \pi_2(x_2)\mu_2 & , k = i \quad \ell = j-1 \\ -(\lambda+\pi_2(x_2)\mu_2), & k = i \quad \ell = j \end{cases}$$

c) If $i \neq 0,\ j = 0$

$$A_\pi((i,j);(k,\ell)) = \begin{cases} \lambda & , \ k = i+1, \ \ell = j \\ \pi_1(x_1)\mu_1 & , \ k = i-1, \ \ell = j+1 \\ -(\lambda+\pi_1(x_1)\mu_1), & k = i \ , \ \ell = j \end{cases}$$

d) If $i \neq 0,\ j \neq 0$

$$A_\pi((i,j);(k,\ell)) = \begin{cases} \lambda & , \ k=i+1, \ \ell=j \\ \pi_1(x_1)\mu_1 & , \ k=i-1, \ \ell=j+1 \\ \pi_2(x_2)\mu_2 & , \ k=i \ , \ \ell=j-1 \\ -(\lambda+\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2), & k=i \ , \ \ell=j \end{cases}$$

e) If $i = K_1,\ j \neq 0$

$$A_\pi((i,j);(k,\ell)) = \begin{cases} \pi_1(x_1)\mu_1 & , \ k = K_1-1, \ \ell = j+1 \\ \pi_2(x_2)\mu_2 & , \ k = K_1 \ , \ \ell = j-1 \\ -(\pi_1(x_1)\mu_1+\pi_2(x_2)\mu_2), & k = K_1 \ , \ \ell = j \end{cases}$$

f) If $i \neq 0,\ j = K_2$

$$A_\pi((i,j);(k,\ell)) = \begin{cases} \lambda & , \ k = i+1 \ , \ \ell = K_2 \\ \pi_2(x_2)\mu_2 & , \ k = i \ , \ \ell = K_2-1 \\ -(\lambda+\pi_2(x_2)\mu_2), & k = i \ , \ \ell = K_2 \end{cases}$$

g) If $i = K_1,\ j = K_2$

$$A_\pi((i,j);(k,\ell)) = \begin{cases} \pi_2(x_2)\mu_2 , & k = K_1 , \quad \ell = K_2-1 \\ -(\pi_2(x_2)\mu_2), & k = K_1 , \quad \ell = K_2 \end{cases}$$

Therefore, from the equation (3.6) in Chapter II, for any policy $\pi \varepsilon M$, the expected total discounted return V_π is the unique solution of the following functional equations.

$$V_\pi(0,0) = \frac{1}{\alpha + \lambda}[h(0,0) + c(\pi(0,0)) + \lambda V_\pi(1,0)]$$

For $0 < x_2 \leq K_2$, $x_1 = 0$

$$V_\pi(0,x_2) = \frac{1}{\pi + \lambda + \pi_2(x_2)\mu_2}[h(0,x_2) + c(\pi(0,x_2))$$

$$+ \lambda V_\pi(1,x_2) + \pi_2(x_2)\mu_2 V_\pi(0,x_2-1)]$$

For $0 < x_1 < K_1,\quad 0 < x_2 < K_2$

$$V_\pi(x_1,x_2) = \frac{1}{\alpha + \lambda + \pi_1(x_1)\mu_1 + \pi_2(x_2)\mu_2}[h(x_1,x_2)$$

$$+ c(\pi(x_1,x_2)) + \lambda V_\pi(x_1+1,x_2)$$

$$+ \pi_1(x_1)\mu_1 V_\pi(x_1-1,x_2+1) + \pi_2(x_2)\mu_2 V_\pi(x_1,x_2-1)]$$

For $0 < x_1 \leq K_1$, $x_2 = 0$

$$V_\pi(x_1,0) = \frac{1}{\alpha + \lambda + \pi_1(x_1)\mu_1}[h(x_1,0) + c(\pi(x_1,0))$$

$$+ \lambda V_\pi(x_1+1,0) + \pi_1(x_1)\mu_1 V_\pi(x_1-1,1)]$$

For $x_1 = K_1$, $0 \leq x_2 < K_2$

$$V_\pi(K_1,x_2) = \frac{1}{\alpha + \pi_1(x_1)\mu_1 + \pi_2(x_2)\mu_2}[h(K_1,x_2)$$

$$+ c(\pi(K_1,x_2)) + \pi_1(x_1)\mu_1 V_\pi(K_1-1,x_2+1)$$

$$+ \pi_2(x_2)\mu_2 V_\pi(K_1,x_2-1)]$$

For $0 \leq x_1 < K_1$, $x_2 = K_2$

$$V_\pi(x_1,K_2) = \frac{1}{\alpha + \lambda + \pi_2(x_2)\mu_2}[h(x_1,K_2) + c(\pi(x_1,K_2))$$

$$+ \lambda V_\pi(x_1+1,K_2) + \pi_2(x_2)\mu_2 V_\pi(x_1,K_2-1)]$$

For $x_1 = K_1$, $x_2 = K_2$

$$V_\pi(K_1,K_2) = \frac{1}{\alpha + \pi_2(x_2)\mu_2}[h(K_1,K_2) + c(\pi(K_1,K_2))$$

$$+ \pi_2(x_2)\mu_2 V_\pi(K_1,K_2-1)]$$

Since we defined as,

$$V(x_1,x_2) = \underset{\pi \varepsilon M}{\text{Min}}\ V_\pi(x_1,x_2) \quad ; \quad x_1 \varepsilon E_1, \quad x_2 \varepsilon E_2$$

Then V is the unique solution of

$$\alpha V(x_1,x_2) = \underset{\pi \varepsilon M}{\text{Min}}\{g_\pi(x_1,x_2) + A_\pi V(x_1,x_2)\}$$

These equations turns out to be similar to the original equation (III.1.5), so we can apply the algorithms, that we defined, to these equations.

(2.16) REMARK: As you notice, the basic difference between the two connected service station system and our original single station system is that the state of the system in the original case is a point, but in here the state is a two-dimensional vector. If we extend this system the state will be an n-dimensional vector and the new state space is $\{K_1 x K_2 x \ldots x K_n\}$ where K_i is the waiting-room capacity of the i station.

VI.2.2 Optimum Service Rate Selection in M/M/$\hat{I}$/K Queues

In many service systems, the overall system may consist of several types of service facilities of different capacities and different operating costs which may be used at different times. Other things being equal, it is desirable that the policy for selecting which service facility to employ be the function of the system state.

Now, we shall consider the optimal control of an $M/M/\hat{I}/K$ queue where $\hat{I}$ indicates a single server with a variable service rate. Other assumptions being the same with the original problem, we say that a policy π is any rule for selecting the service rate from the given set as a function of system state. Here service cost (operating cost) can be considered as the function of the different service rates, i.e., $c(\pi(x)) = c(r)$. Now, we are trying to find the optimum service rate as a function of system state which minimizes the infinite horizon expected discounted cost.

Let $R \equiv [r_1, r_2, \ldots, r_k]$ be the available service rate set. Then the equation (III.1.5) turns out to be,

$$V(x) = \underset{r \varepsilon R}{\text{Min}}\{ \frac{1}{\alpha + \lambda(x) + r} y_r(x) + \lambda(x)V(x+1) + rV(x-1)]\}$$

$$0 \leq x \leq K$$

where

$$y_r(x) = \begin{cases} h(x) + c(r) & , 0 \leq x < K \\ h(K) + c(r) + \gamma\ell & , x = K \end{cases}$$

For every state $x \varepsilon E$, the corresponding values which minimize the right hand side of the above equation gives the optimum policy π^*. The algorithms presented in the previous chapter are also applicable to this problem.

VI.2.3 Optimum Service Rate Selection in Series Queues With no Waiting Room

In this section, we study the model which combines the model in Section 2 with the one in Section 3 and adds a new property which does not permit any queue formation.

We consider a simple two-station single-server-at-each-station model where no queue is allowed to form at either station. If a customer is in station 2, and service is completed at station 1, the station 1 customer must wait there until the service of the station 2 customer is completed; that is, the system is blocked. Arrivals at station 1 when the system is blocked are turned away. Also if a customer is in process at station 1, even if station 2 is empty, arriving customers are turned away, since the system is a sequential one; that is, all customers require service at 1 and then service at 2.

Therefore using the assumptions in Section 2.1, the possible system state and corresponding generator is as follows,

$$A_\pi = \begin{array}{c} \\ (0,0) \\ (1,0) \\ (0,1) \\ (1,1) \end{array} \begin{array}{c} \begin{array}{cccc} (0,0) & (1,0) & (0,1) & (1,1) \end{array} \\ \left[\begin{array}{cccc} -\lambda & \lambda & & \\ & -\mu_1 & \mu_1 & \\ \mu_2 & & -(\lambda+\mu_2) & \lambda \\ & \mu_2 & & -\mu_2 \end{array}\right] \end{array}$$

where $\pi(x_1,x_2) = (\mu_1,\mu_2)$ and $\mu_1,\mu_2 \in R \equiv [r_1,\ldots,r_m]$ and also our state space $E = \{(0,1) \times (0,1)\}$.

For this problem, we do not have holding cost because of the no allowance of queue formation. So we have only operating cost depending on which service rates are used; that is, $c(\pi(x_1,x_2)) = c(\mu_1,\mu_2)$ and also lost profit incurres with $\lambda\ell$ when the system is blocked.

Therefore, the minimum expected discounted cost is the unique solution of the following equations.

$$V(0,0) = \min_{\substack{\mu_1 \in R \\ \mu_2 \in R}} \left\{\frac{1}{\alpha + \lambda}[c(\mu_1,\mu_2) + \lambda V(1,0)]\right\}$$

$$V(1,0) = \min_{\substack{\mu_1 \in R \\ \mu_2 \in R}} \left\{\frac{1}{\alpha + \mu_1}[c(\mu_1,\mu_2) + \mu_1 V(0,1)]\right\}$$

$$V(0,1) = \min_{\substack{\mu_1 \in R \\ \mu_2 \in R}} \left\{\frac{1}{\alpha + \lambda + \mu_2}[c(\mu_1,\mu_2) + \lambda V(1,1) + \mu_2 V(0,0)]\right.$$

$$V(1,1) = \underset{\substack{\mu_1 \varepsilon R \\ \mu_2 \varepsilon R}}{\text{Min}} \{\frac{1}{\alpha + \mu_2}[c(\mu_1,\mu_2) + \lambda \ell + \mu_2 V(1,0)]\}$$

Optimum policy π^* corresponds to the μ_1 and μ_2 values in each state which minimize the right-hand-side of the above equations.

IV.3 Application: Finite Population Problem

In this section we give a simple problem and write computer programs for the algorithms defined in the previous chapter to see the convergence of them and to prove the existence of the optimal policy for that problem.

As an example we study the finite population model and not the finite queue model as in the original problem. Consider a computer model like in Figure 4. Here we have M users, or computer consoles, that make demands upon the time-shared computer system. Finite population model operates as follows: when a user at a console makes a request for service of the computer, the request enters the processor's queue and proceeds to receive service. During this time the user cannot generate any new request. When finally that request is complete, the response is fed back to the console at which point the user at the console begins to generate a new request for the computer.

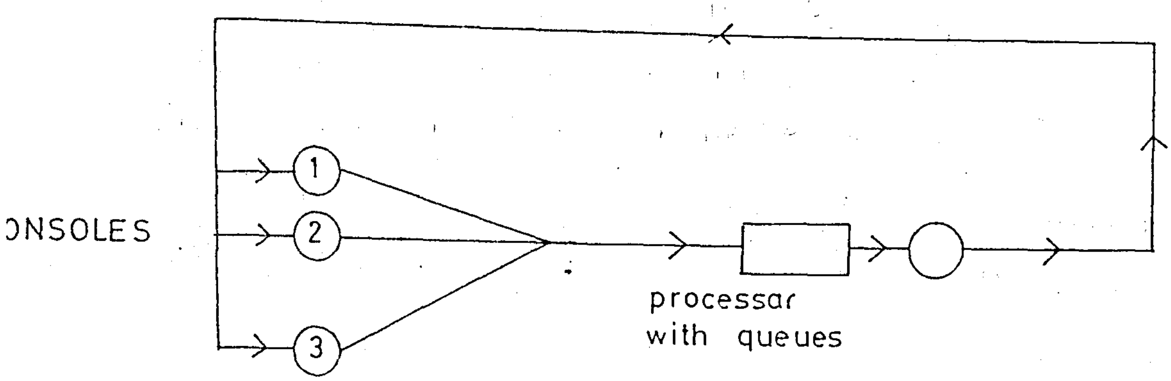

Fig. 4 - Finite input population

Another important case for this model is the maintenance systems on which we shall concentrate as an example.

EXAMPLE PROBLEM: Consider a company which has 60 machines. However, because these machines break-down and require repair frequently, the company has only enough operators to operate 50 machines at a time, so ten machines are available on a standby basis for use in cases of failures of the operating machines. Thus 50 machines are always operating whenever no more than ten machines are waiting to be repaired, but the number of operating machines is reduced by one for each additional machine waiting to be repaired.

The time until any given operating machine breaks down has an exponential distribution, with a mean of 20 days. The time required to repair a machine also has an exponential distribution with a mean of 2 days. The company has no repairman to repair these machines. However, productivity is reduced by having less than 50 operating machines. They want to make a decision about the number of server they can hire by considering the repair cost and their lost profit. But the company which gives the server to this company has at most 15 servers for this kind of repair activities. Figure 5 shows the system situation.

As you notice the queueing system to be studied has the repairmen as its servers and the machines requiring repair as its customers. This is a simple maintenance problem. In this problem, customer number is limited with 60 machines. This is the case of 'limited source' problem. Thus when the number of customers in the system (number of broken machines) is n, there are only (60-n) potential customers. The elapsed time from leaving the system until returning for the next time for a machine is given as an exponential distribution with rate $\lambda = 1/20$ machines per day. When n machines are broken, the current probability distribution of a remaining time until the next arrival to the queueing system has an exponential distribution with parameter $\lambda(n) = (50 - n)\lambda$, for $n > 10$. Because of the standby machines $\lambda(0) = \lambda(1) = \ldots. = \lambda(10) = 50\lambda$.

As you notice the state of the system (X_t) is taken as the number of broken machines in the system. This model is not the finite queue model, but this does not change anything in the formulas that we found in the original case. Only we will not have any shortage cost here. Then we use the formulas (1.4) in Chapter III and (3.5) in Chapter II, by taking $\gamma\ell$ (shortage cost) equal to zero.

The company estimates its lost profit for not having a machine operating to produce units as 36500 TL/year. And the yearly server cost list is given in the program. The discount rate is taken as 0.25.

With respect to this data we will discuss the result of the programs. The computer programs and the results for successive approximation and policy improvement algorithms are given in Appendix I and II, respectively.

ANALYSIS OF THE RESULTS:

The major difference of the two algorithms is their convergence rates. In our limited computational experience typical computer run reaches the optimal policy in 3 iterations with 60 second CPU time for the policy improvement algorithm whereas the successive approximation method does not reach the optimal in 901 iterations with 120 seconds CPU time. Even though the computational burden at each iteration of the policy improvement method is higher, its overall computational efficiency with small size problems that we experimented on is much promising.

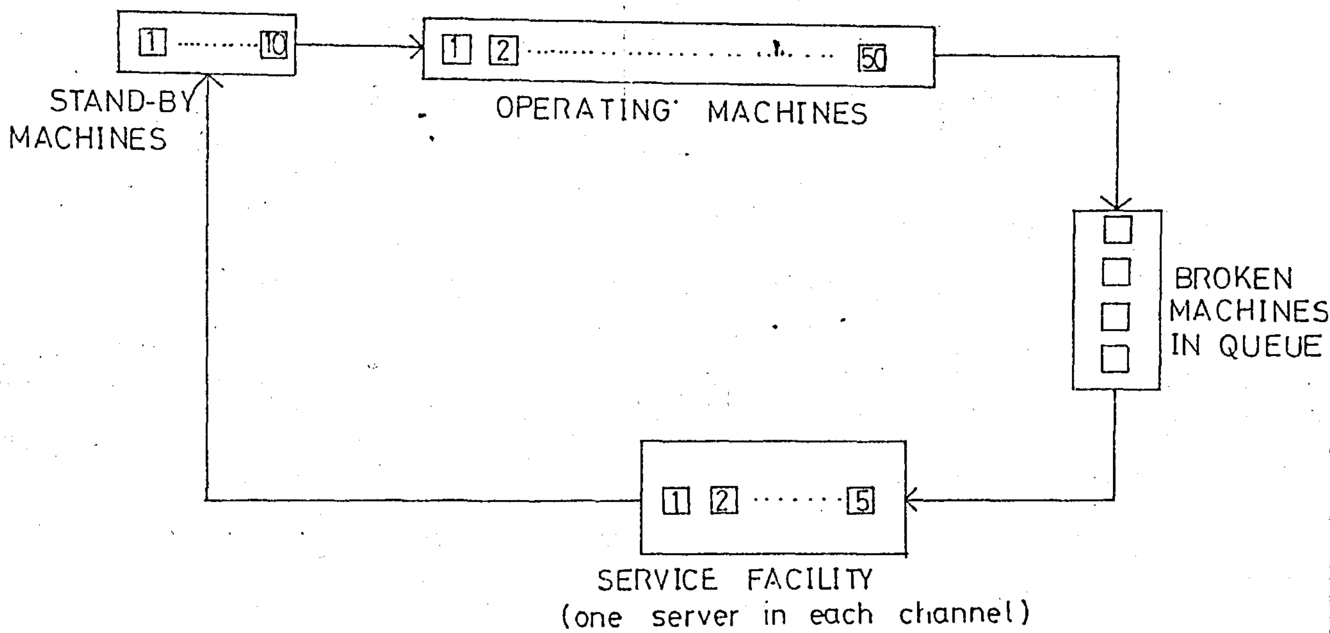

Fig. 5 - Maintenance system in the example problem

In policy improvement method no repetitions will occur. In each iteration the new policy has a smaller return than the policy under consideration. But this is not the case in successive approximation method.

It determines the values iteratively. In any iteration the new value is smaller than the previous one, but the policy could be the same.

In practice, method of successive approximation may be used when an approximation or guess to an expected discounted cost criterion is available. Then several iterations will hopefully improve it. In any case, use of the method of successive approximation never necessitates the computation of an exact discounted cost criterion. If we wish to minimize the total expected discounted cost over only a few stages of the process, not the infinite duration, then successive approximation method is preferable.

In summary, policy improvement procedure provides a monotone convergent sequence of policies and attains the optimal policy in a finite number of iterations. Its drawback is that the discounted cost function for each policy π in the sequence must be calculated. This involves solving the system of linear equations. When the size of the problem increases, this inflates the computational burden tremendously. Also each computer has different restrictions in taking the inverse of a matrix. After some specified size, computer cannot take the inverse of a matrix, then it cannot solve the system of linear equations. Therefore in such cases, the only possibility in obtaining the optimal policy is the successive approximation algorithm.

V. CONCLUSION

This thesis has accomplished two tasks. The first is the control of the queue model described in Section II.1. This model is a continuous-time Markov process, and we want to find the infinite horizon expected discounted return of this process based on the given cost structure. We specified the generator of the process and its range and domain in Section II.2. The importance of the generator can be seen in theorem (II.3.1). This theorem saves us from the integral and expectation parts of equation (II.1.4), then it turns out to be the system of linear equations. Theorem (II.3.4) gives us a complete characterization of the return function for any given policy, which is in the admissible policy set, as the unique solution of a functional dynamic equations.

Section III.1 analyzes the controlled queue model and theorem (III.1.9) states the sufficient condition of optimality. Then the results of the original problem are extended for some other queue models in Section IV.2.

The second taks is the analysis of the algorithms used in obtaining the optimal policy, which are defined in Section III.2 and III.3. We proved that successive approximation algorithm determines the values iteratively and then it converges to the minimum expected discounted cost function and also determines optimal policy. But the policy improvement algorithm is taken from Doshi(1976) and adopted to our model.

Section IV.3 gives a maintenance system as an example. This problem is solved by computer with these two algorithms, then the comparison of the two method is done in the previous chapter.

In summary, we can say that; for any Markov process with finite state space and finite action space, if you want to find the infinite horizon expected discounted cost function use theorem (II.3.1) and write the corresponding functional equations. And the optimal policy which minimizes the cost function is easily found by the policy improvement algorithm if the system size does not exceed the computer restrictions. Otherwise, use the successive approximation algorithm.

REFERENCES

1. Blackburn, J.D., "Optimal Control of a Single Server Queue with Balking and Reneging", Management Science, Vol. 19, pp. 297-313, 1972.

2. Breiman, L., Probability, Addison-Wesley, Topanga, California, 1968.

3. Brosh, I., "The Policy Space Structure of Markovian Systems with Two Types of Service", Management Science, Vol. 16, pp. 607-621, 1970.

4. Crabil, T.B., "Optimal Control of a Service Facility with Variable Exponential Service Time and Constant Arrival Rate", Management Science, Vol. 18, pp. 560-566, 1972.

5. Çınlar, E., Introduction to Stochastic Processes, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

6. Derman, C., Finite State Markovian Decision Processes, Academic Press, New York, 1970.

7. Doshi, B.T., "Continuous Time Control of Markov Processes on an Arbitrary State Space", The Annals of Statistics, Vol. 4, pp. 1219-1235, 1976.

8. Heyman, D.P., "Optimal Operating Policies for M/G/1 Queueing Systems", Operation Research, Vol. 16, pp. 362-382, 1968.

9. Kakumanu, P.K., "Continuously Discounted Markov Decision Model with Countable State and Action Space", Ann. Math. Statistics, Vol. 42, pp. 919-926, 1972.

10. Lipman, S.A., "Optimal Control of an M/M/1 Queue with Variable Service Rate", Bull. Opr. Res. Soc. Ann., 23-b479, Fall 1975.

11. Magazine, M.J., "Optimal Control of Multichannel Service Systems", Nav. Res. Log. Qua., Vol. 18, pp. 177-183, 1971.

12. Miller, B., "Finite State Continuous Time Markov Decision Process", J. Math. Anal. App., Vol. 22, pp. 552-569, 1968.

13. Miller, B., "A Queueing Reward System with Several Customer Classes", Management Science, Vol. 16, pp. 234-245, 1969.

14. Mitchell, B., "Optimal Service Rate Selection in M/G/1 Queue", Siam J. Appl. Math., Vol. 24, pp. 19-34, 1973.

15. Ross, S., Applied Probability Models with Optimization Applications, Holden-Day, San Francisco, California, 1970.

16. Sobel, M.J., "Optimal Average Cost Policy for a Queue with Start-up and Shut-down Costs", Operations Research, Vol. 17, pp. 145-162, 1969.

17. Walrand, J., and P. Varaiya, and Z. Rosberg, "Optimal Control of Service in Tandem Queues", University of California, Berkeley, 1980.

APPENDIX I

INPUT-OUTPUT STRUCTURE

OF THE COMPUTER PROGRAMS

After the process is modeled and its functional equations are written, compare them with the equation (II.3.5) if the general structure is the same, these programs can be used.

POLICY IMPROVEMENT ALGORITHM:

INPUT DATA: Necessary data to run this program is as follows:

The first card contains: ALF (discount rate), MBAR (maximum number of servers that can be employed), K (system size), L (lost profit/customer; it will be zero if you have no shortage cost), SR (common service rate of each server), GR (average arrival rate, which is necessary in the presence of shortage cost).

Second data card contains the server costs. It has (MBAR+1) data.

The third and the fourth data cards contains (K+1) numbers where the i^{th} number in the first one shows the holding cost incurred when there are i customer in the system and the second one shows the arrival rate when the system size is i.

All data is given with free format.

INITIALIZATION: Initial policy in this program is taken as,

$$\pi_o(x) = 0, \qquad x \in E$$

If you want to give different initial policy change the statements 500-530 where $V_{\pi_0}(x)$ values (V(0,x)) are computed for this given initial policy.

OUTPUT DATA: In the output, you will first see the input data. Then in the solution part, the optimal policy and the cost corresponding to this policy will be seen. At the end you will find the number of iterations required to reach the optimal policy.

SUCCESSIVE APPROXIMATION ALGORITHM:

INPUT DATA: The input structure is same with the other method. Since this method finds the epsilon-optimal policy, it needs epsilon value. Add this value to the end of the first data card.

OUTPUT DATA: Also the output structure is similar with the policy improvement algorithm. Since this method needs too more iterations to reach the optimum policy, we put a restriction on the iteration number (N = 1000). Then in the solution part, only the values for the last two preceeding iterations and the policy obtained in the last cycle is printed.

APPENDIX II

PROGRAM LISTING AND RESULTS
OF THE POLICY IMPROVEMENT ALGORITHM

```
C
C
C
C
C
C
C
C                    **************************
C                    *                        *
C                    *   POLICY  IMPROVEMENT  *
C                    *                        *
C                    *       ALGORITHM        *
C                    *                        *
C                    **************************
C
C
C
C
C
C
      DIMENSION A(0:70,0:70),V(0:150,0:70),JC(70)
      INTEGER S(0:150,0:70)
      DIMENSION ALND(0:70),VAR(0:70),C(0:70),H(0:70),Z(1)
      READ(5,*) ALF,MBAR,K,L,SR,GR
      READ(5,*) (C(KW),KW=0,MBAR)
      READ(5,*) (H(IW),IW=0,K)
      READ(5,*) (ALND(LWE),LWE=0,K)
      WRITE(6,557)
557   FORMAT(1H1,////////,50X,,INPUT  DATA  OF  THE  PROBLEM,,/,50X,
     &29(,-,))
      WRITE(6,544) L,SR,ALF,K,MBAR
544   FORMAT(///,55X,,LOST PROFIT          =,,I4,//,55X,,SERVICE RATE      ,
     &,  =,,F4.0,//,55X,,DISCOUNT RATE        =,,F4.3,//55X,,QUEUE CAPACITY
     &       =,,I4,//55X,,MAX. SERVER NUMBER=,,I4,/)
      WRITE(6,555)
555   FORMAT(/////,50X,,SERVER  NO.,,6X,,SERVER  COST,,/,50X,11(,-,)
     &,6X,12(,-,),///)
      WRITE(6,545) (KWW,C(KWW),KWW=0,MBAR)
545   FORMAT(55X,I2,12X,F8.0,/)
      WRITE(6,556)
556   FORMAT(1H1,/////,42X,,QUEUE  SIZE,,6X,,ARRIVAL RATE,,6X,,HOLDI,
     &,NG  COST,,/,42X,11(,-,),6X,13(,-,),6X,13(,-,),///)
      WRITE(6,546) (LWW,ALND(LWW),H(LWW),LWW=0,K)
546   FORMAT(47X,I2,14X,F5.0,11X,F10.0,/)
      WRITE(6,574)
574   FORMAT(1H1,////////,42X,,SOLUTION OF THE POLICY IMPROVEMENT ,
     &,ALGORITHM,,/,42X,44(,-,),/////)
      KK=K-1
      KKK=K+1
      N=0
      V(0,K)=(H(K)+C(0)+GR*L)/ALF
      DO 1 I=K,1,-1
      II=I-1
1     V(0,II)=(H(II)+C(0)+ALND(II)*V(0,I))/(ALF+ALND(II))
42    NN=N+1
      DO 2 LL=1,KK
      IF(LL.LT.MBAR) GO TO 3
      LA=LL+1
      LB=LL-1
      IA=MBAR
      GO TO 4
3     LA=LL+1
      LB=LL-1
      IA=LL


4     DO 14 M=0,IA
14    VAR(M)=(H(LL)+C(M)+ALND(LL)*V(N,LA)+M*SR*V(N,LB))/(ALF+
     &ALND(LL)+SR*M)
      EOR=VAR(0)
      S(NN,LL)=0
      DO 15 MA=1,IA
      IF(EOR.LT.VAR(MA)) GO TO 15
      S(NN,LL)=MA
      EOR=VAR(MA)
15    CONTINUE
2     CONTINUE
      S(NN,0)=0
      IB=MIN(K,MBAR)
      DO 19 MK=0,IB
19    VAR(MK)=(H(K)+C(MK)+GR*L+MK*SR*V(N,KK))/(ALF+MK*SR)
      EOR=VAR(0)
      S(NN,K)=0
```

```
      DO 20 MC=1,IB
      IF(FOR.LT.VAR(MC)) GO TO 20
      FOR=VAR(MC)
      S(NN,K)=MC
20    CONTINUE
      DO 47 NT=0,K
      DO 48 NS=0,KKK
48    A(NT,NS)=0.
47    CONTINUE
      A(0,0)=ALF+ALND(0)
      A(0,1)=ALND(0)*(-1)
      DO 31 MY=1,KK
      MZ=MY+1
      MW=MY-1
      A(MY,MW)=S(NN,MY)*SR*(-1)
      A(MY,MY)=ALND(MY)+S(NN,MY)*SR+ALF
      A(MY,MZ)=(-1)*ALND(MY)
31    CONTINUE
      A(K,KK)=(-1)*S(NN,K)*SR
      A(K,K)=ALF+SR*S(NN,K)
      DO 30 MX=0,KK
30    A(MX,KKK)=H(MX)+C(S(NN,MX))
      A(K,KKK)=H(K)+C(S(NN,K))+GR*L
      Z(1)=4.
      CALL GJR(A,71,71,61,62,$150,JC,Z)
      DO 32 LZ=0,K
32    V(NN,LZ)=A(LZ,KKK)
      DO 33 LY=0,K
      DIF=V(N,LY)-V(NN,LY)
      IF(DIF.NE.0) GO TO 40
33    CONTINUE
      GO TO 41
40    IF(NN.EQ.200) GO TO 41
      N=NN
      GO TO 42
41    WRITE(6,49)
49    FORMAT(37X,,QUEUE SIZE,,5X,,OPTIMAL POLICY,,5X,,OPTIMAL COST,,
     &/,36X,12(,-,),3X,16(,-,),3X,14(,-,),///)
      WRITE(6,50) (NE,S(NN,NE),V(NN,NE),NE=0,K)
50    FORMAT(42X,I2,15X,I2,14X,F10.0,/)
      WRITE(6,51) N
51    FORMAT(////////,55X,,ITERATION NUMBER=,,I2,/,54X,22(,-,))
      GO TO 500
150   WRITE(6,511)
511   FORMAT(/,20X,,GJR DA HATA VAR,)
500   STOP
      END
```

ANK 27284 DBANK

INPUT DATA OF THE PROBLEM

LOST PROFIT = 0.
SERVICE RATE =180.
DISCOUNT RATE =.250
QUEUE CAPACITY = 60
MAX. SERVER NUMBER= 15

SERVER NO.	SERVER COST
0	0.
1	25000.
2	75000.
3	150000.
4	200000.
5	275000.
6	350000.
7	450000.
8	550000.
9	650000.
10	750000.
11	850000.
12	950000.
13	1050000.
14	1150000.
15	1250000.

QUEUE SIZE	ARRIVAL RATE	HOLDING COST
0	900.	0.
1	900.	0.
2	900.	0.
3	900.	0.
4	900.	0.
5	900.	0.
6	900.	0.
7	900.	0.
8	900.	0.
9	900.	0.
10	900.	0.
11	882.	36500.
12	864.	73000.
13	846.	109500.
14	828.	146000.
15	810.	182500.
16	792.	219000.
17	774.	255500.
18	756.	292000.
19	738.	328500.
20	720.	365000.
21	702.	401500.
22	684.	438000.
23	666.	474500.
24	648.	511000.
25	630.	547500.
26	612.	584000.
27	594.	620500.
28	576.	657000.
29	558.	693500.
30	540.	730000.
31	522.	766500.
32	504.	803000.
33	486.	839500.
34	468.	876000.
35	450.	912500.
36	432.	949000.

37	414.	985500.
38	396.	1022000.
39	378.	1058500.
40	360.	1095000.
41	342.	1131500.
42	324.	1168000.
43	306.	1204500.
44	288.	1241000.
45	270.	1277500.
46	252.	1314000.
47	234.	1350500.
48	216.	1387000.
49	198.	1423500.
50	180.	1460000.
51	162.	1496500.
52	144.	1533000.
53	126.	1569500.
54	108.	1606000.
55	90.	1642500.
56	72.	1679000.
57	54.	1715500.
58	36.	1752000.
59	18.	1788500.
60	0.	1825000.

SOLUTION OF THE POLICY IMPROVEMENT ALGORITHM

QUEUE SIZE	OPTIMAL POLICY	OPTIMAL COST
0	0	1153254.
1	1	1153574.
2	2	1153931.
3	2	1154341.
4	4	1154709.
5	4	1155110.
6	6	1155537.
7	6	1155981.
8	6	1156446.
9	6	1156936.
10	6	1157457.
11	11	1158014.
12	12	1158589.
13	13	1159178.
14	14	1159779.
15	15	1160389.
16	15	1161011.
17	15	1161647.
18	15	1162295.
19	15	1162956.
20	15	1163629.
21	15	1164314.
22	15	1165011.
23	15	1165719.
24	15	1166439.
25	15	1167170.
26	15	1167911.
27	15	1168664.
28	15	1169427.
29	15	1170200.
30	15	1170984.
31	15	1171779.

32	15	1172581.
33	15	1173395.
34	15	1174218.
35	15	1175050.
36	15	1175892.
37	15	1176743.
38	15	1177602.
39	15	1178471.
40	15	1179349.
41	15	1180235.
42	15	1181129.
43	15	1182032.
44	15	1182943.
45	15	1183862.
46	15	1184790.
47	15	1185725.
48	15	1186668.
49	15	1187618.
50	15	1188576.
51	15	1189542.
52	15	1190514.
53	15	1191494.
54	15	1192481.
55	15	1193476.
56	15	1194477.
57	15	1195485.
58	15	1196499.
59	15	1197521.
60	15	1198549.

ITERATION NUMBER= 3

APPENDIX III

PROGRAM LISTING AND RESULTS OF THE SUCCESSIVE APPROXIMATION ALGORITHM

```
C
C
C                        *****************************
C                        *                           *
C                        *   SUCCESSIVE APPROXIMATION  *
C                        *                           *
C                        *        ALGORITHM          *
C                        *                           *
C                        *****************************
C
C
      DIMENSION V(0:70,0:70),ALND(0:70),VAR(0:70),VARK(0:70)
      DIMENSION IS(70),C(0:20),H(0:70)
      INTEGER S(0:70)
      READ(5,*)ALF,MBAR,K,L,SR,EPS,GR
      READ(5,*) (C(KW),KW=0,MBAR)
      READ(5,*) (H(IW),IW=0,K)
      READ(5,*) (ALND(LWE),LWE=0,K)
      WRITE(6,557)
557   FORMAT(1H1,///////,50X,,INPUT DATA OF THE PROBLEM,,/,50X,
     &29(,$,))
      WRITE(6,544) L,SR,ALF,K,MBAR,EPS
544   FORMAT(///,55X,,LOST PROFIT          =,,I4,//,55X,,SERVICE RATE
     &   =,,F4.0,//,55X,,DISCOUNT RATE       =,,F4.3,//55X,,QUEUE CAPACIT
     &   =,,I4,//,55X,,MAX. SERVER NUMBER=,,I4,//,55X,,EPSILON
     &   =,,F4.0,/)
      WRITE(6,555)
555   FORMAT(/////,50X,,SERVER  NO.,,6X,,SERVER  COST,,/,55X,11(,$,)
     &,6X,12(,$,),///)
      WRITE(6,545) (KWW,C(KWW),KWW=0,MBAR)
545   FORMAT(55X,I2,12X,F8.0)
      WRITE(6,556)
556   FORMAT(1H1,////,42X,,QUEUE  SIZE,,6X,,ARRIVAL RATE,,6X,
     &,HOLDING  COST,,/,42X,11(,$,),6X,13(,$,),6X,13(,$,),///)
      WRITE(6,546) (LWW,ALND(LWW),H(LWW),LWW=0,K)
546   FORMAT(47X,I2,14X,F5.0,11X,F10.0)
      WRITE(6,574)
574   FORMAT(1H1,//////,40X,,SOLUTION OF THE SUCCESSIVE APPROXIMATION
     & ALGORITHM,,/,40X,44(,$,),/////)
      S(0)=0
      N=1
      NN=1000
      KK=K-1
      KKK=K+1
      V(0,K)=(H(K)+C(0)+ALND(K)*L)/ALF
      DO 1 I=K,1,-1
      II=I-1
1     V(0,II)=(H(II)+C(0)+ALND(II)*V(0,I))/(ALF+ALND(II))
      V(1,0)=(H(0)+C(0)+ALND(0)*V(0,1))/(ALF+ALND(0))
      DO 2 LL=1,KK
      LA=LL+1
      LB=LL-1
      II=MINO(LL,MBAR)
      DO 3 IO=0,II
3     VAR(IO)=(H(LL)+C(IO)+ALND(LL)*V(0,LA)+IO*SR*V(0,LB))/(ALF+ALND(LL
     &+IO*SR)
      V(1,LL)=VAR(0)
      S(LL)=0.0
      DO 31 IB=1,II
      IF(V(1,LL).LE.VAR(IB)) GO TO 31
      V(1,LL)=VAR(IB)
      S(LL)=IB
31    CONTINUE


2     CONTINUE
      IA=MINO(K,MBAR)
      DO 4 MA=0,IA
4     VARK(MA)=(H(K)+C(MA)+GR*L+MA*SR*V(0,KK))/(ALF+MA*SR)
      V(1,K)=VARK(0)
      S(K)=0.
      DO 41 MC=1,IA
      IF(V(1,K).LE.VARK(MC)) GO TO 41
      V(1,K)=VARK(MC)
      S(K)=MC
41    CONTINUE
      DO 5 LC=0,K
```

```
100   N=N+1
      IF(N.GT.NN) GO TO 101
      V(2,0)=(H(0)+C(0)+ALND(0)*V(1,1))/(ALF+ALND(0))
      DO 12 LY=1,KK
      LYA=LY+1
      LYB=LY-1
      IB=MIN0(LY,MBAR)
      DO 13 IY=0,IB
13    VAR(IY)=(H(LY)+C(IY)+ALND(LY)*V(1,LYA)+IY*SR*V(1,LYB))/(ALF+ALN
     &D(LY)+IY*SR)
      V(2,LY)=VAR(0)
      S(LY)=0.
      DO 44 IZ=1,IB
      IF(V(2,LY).LE.VAR(IZ)) GO TO 44
      V(2,LY)=VAR(IZ)
      S(LY)=IZ
44    CONTINUE
12    CONTINUE
      DO 14 MY=0,IA
14    VARK(MY)=(H(K)+C(MY)+GR*L+MY*SR*V(1,KK))/(ALF+MY*SR)
      V(2,K)=VARK(0)
      S(K)=0.
      DO 45 MH=1,IA
      IF(V(2,K).LE.VARK(MH)) GO TO 45
      V(2,K)=VARK(MH)
      S(K)=MH
45    CONTINUE
      DO 15 NB=0,K
      DIFF=V(1,NB)-V(2,NB)
      IF(ABS(DIFF).GT.EPS) GO TO 102
15    CONTINUE
      GO TO 101
102   N=N+1
      IF(N.GT.NN) GO TO 101
      V(1,0)=(H(0)+C(0)+ALND(0)*V(2,1))/(ALF+ALND(0))
      DO 22 NY=1,KK
      NYA=NY+1
      NYB=NY-1
      IZA=MIN0(NY,MBAR)
      DO 23 IV=0,IZA
23    VAR(IV)=(H(NY)+C(IV)+ALND(NY)*V(2,NYA)+IV*SR*V(2,NYB))/(ALF+AL
     &ND(NY)+IV*SR)
      V(1,NY)=VAR(0)
      IS(NY)=0.
      DO 46 IH=1,IZA
      IF(V(1,NY).LE.VAR(IH)) GO TO 46
      V(1,NY)=VAR(IH)
      IS(NY)=IH
46    CONTINUE
22    CONTINUE
      DO 24 MF=0,IA
24    VARK(MF)=(H(K)+C(MF)+GR*L+MF*SR*V(2,KK))/(ALF+MF*SR)
      V(1,K)=VARK(0)
      IS(K)=0.
      DO 47 MU=1,IA
      IF(V(1,K).LE.VARK(MU)) GO TO 47
      V(1,K)=VARK(MU)
      IS(K)=MU
47    CONTINUE
      DO 48 MS=0,K
      FARK=V(1,MS)-V(2,MS)
      IF(ABS(FARK).GT.EPS) GO TO 100
48    CONTINUE
      GO TO 101
1001  WRITE(6,71) (V(0,NE),V(1,NE),S(NE),NE=0,K)
      PRINT*,N


      GO TO 105
101   LMK=N-1
      WRITE(6,511) LMK,N
511   FORMAT(25X,,QUEUE,,8X,,VALUE AT,,11X,,VALUE AT,,11X,,POLICY AT
     &,,/,26X,,SIZE,,5X,,ITERATION,,I4,5X,,ITERATION ,,I4,5X,,ITERATION
     & ,,I4,///)
      WRITE(6,71) (NF,V(1,NF),V(2,NF),S(NF),NF=0,K)
71    FORMAT(27X,I2,7X,F12.4,7X,F12.4,10X,I3)
105   STOP
      END
```

INPUT DATA OF THE PROBLEM

LOST PROFIT = 0

SERVICE RATE =180.

DISCOUNT RATE =.250

QUEUE CAPACITY = 60

MAX. SERVER NUMBER= 15

EPSILON =100.

SERVER NO.	SERVER COST
0	0.
1	25000.
2	75000.
3	150000.
4	200000.
5	275000.
6	350000.
7	450000.
8	550000.
9	650000.
10	750000.
11	850000.
12	950000.
13	1050000.
14	1150000.
15	1250000.

QUEUE SIZE	ARRIVAL RATE	HOLDING COST
0	900.	0.
1	900.	0.
2	900.	0.
3	900.	0.
4	900.	0.
5	900.	0.
6	900.	0.
7	900.	0.
8	900.	0.
9	900.	0.
10	900.	0.
11	882.	36500.
12	864.	73000.
13	846.	109500.
14	828.	146000.
15	810.	182500.
16	792.	219000.
17	774.	255500.
18	756.	292000.
19	738.	328500.
20	720.	365000.
21	702.	401500.
22	684.	438000.
23	666.	474500.
24	648.	511000.
25	630.	547500.
26	612.	584000.
27	594.	620500.
28	576.	657000.
29	558.	693500.
30	540.	730000.
31	522.	766500.
32	504.	803000.
33	486.	839500.
34	468.	876000.
35	450.	912500.
36	432.	949000.
37	414.	985500.
38	396.	1022000.
39	378.	1058500.
40	360.	1095000.
41	342.	1131500.
42	324.	1168000.
43	306.	1204500.
44	288.	1241000.
45	270.	1277500.
46	252.	1314000.
47	234.	1350500.
48	216.	1387000.
49	198.	1423500.
50	180.	1460000.
51	162.	1496500.
52	144.	1533000.
53	126.	1569500.
54	108.	1606000.
55	90.	1642500.
56	72.	1679000.
57	54.	1715500.
58	36.	1752000.
59	18.	1788500.
60	0.	1825000.

SOLUTION OF THE SUCCESSIVE APPROXIMATION ALGORITHM

QUEUE SIZE	VALUE AT ITERATION1000	VALUE AT ITERATION 1001	POLICY AT ITERATION
0	6396395.5000	6395663.3750	0
1	6397440.5000	6396705.0625	1
2	6398516.2500	6397784.5625	2
3	6399616.1875	6398880.5000	2
4	6400720.5000	6399988.6875	4
5	6401842.3125	6401106.5000	4
6	6402971.3125	6402239.2500	4
7	6404113.1250	6403377.0625	4
8	6405258.3125	6404526.1250	4
9	6406413.5000	6405677.2500	6
10	6407579.6875	6406839.3125	6
11	6408739.1250	6408002.7500	6
12	6409910.0625	6409177.5625	6
13	6411090.8125	6410354.3125	6
14	6412272.9375	6411540.3125	6
15	6413463.6250	6412727.1250	6
16	6414654.7500	6413922.1250	6
17	6415853.7500	6415117.1250	6
18	6417052.3750	6416319.8125	6
19	6418258.3750	6417521.6875	6
20	6419463.3125	6418730.7500	6
21	6420675.1250	6419939.3750	6
22	6421885.3125	6421152.7500	6
23	6423102.1250	6422365.3125	6
24	6424316.8750	6423584.4375	6
25	6425537.9375	6424801.1875	6
26	6426756.5625	6426024.1250	6
27	6427981.3125	6427244.6250	6
28	6429203.4375	6428471.0625	6
29	6430431.4375	6429694.8125	6
30	6431656.6250	6430924.1875	6
31	6432887.3125	6432150.8125	6
32	6434115.3125	6433382.8125	6
33	6435348.3750	6434612.0625	6
34	6436578.7500	6435846.2500	6
35	6437814.0000	6437077.8750	6
36	6439046.5625	6438314.1250	6
37	6440283.7500	6439547.7500	6
38	6441518.2500	6440785.9375	6
39	6442757.3750	6442021.3750	6
40	6443993.5625	6443261.3750	6
41	6445234.3750	6444499.4375	6
42	6446472.1875	6445740.0625	6
43	6447714.3750	6446978.6875	6
44	6448953.6250	6448221.6250	6
45	6450197.3125	6449461.5625	6
46	6451437.9375	6450706.0625	6
47	6452683.0625	6451947.3125	6
48	6453924.7500	6453193.0625	6
49	6455170.8750	6454435.4375	6
50	6456414.0000	6455682.1875	6
51	6457661.1250	6456925.8750	6
52	6458905.3125	6458173.5625	6
53	6460153.3750	6459418.3125	6
54	6461398.6875	6460666.9375	6
55	6462647.5000	6461912.7500	6
56	6463893.7500	6463162.1250	6
57	6465143.6875	6464408.8750	6
58	6466390.6250	6465659.3125	6
59	6467641.6875	6466906.6875	6
60	6468890.1875	6468158.2500	6

VITAE

NAME : Nurizer Özekici (Müceldili)

DATE OF BIRTH : February 9, 1957

PLACE OF BIRTH: Erzurum

EDUCATION : M.S. Industrial Engineering
Boğaziçi University
June 1982.

B.S. Industrial Engineering
Boğaziçi University
June 1979.