4967-5-T

Technical Report No. 153

OPTIMAL MARKOVIAN QUEUEING SYSTEMS

by

K. Chuang

Approved by: ____________________
for B. F. Barton

for

COOLEY ELECTRONICS LABORATORY

Department of Electrical Engineering
The University of Michigan
Ann Arbor, Michigan

Air Force Systems Command
Rome Air Development Center
Contract No. AF 30(602)-2661
Griffiss Air Force Base, New York

June 1963

engm

UMR1172

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENT

The author wishes to acknowledge the valuable comments on this paper made by Mr. V. L. Wallace of the Cooley Electronics Laboratory.

1. INTRODUCTION

Due to the tremendous cost involved in constructing and operating a multiple-computer system, it is obvious that preliminary analysis of the performance of a proposed system should be made before the actual system is built. Furthermore, even with an existing system it is advantageous to explore analytically the capability of the system without interrupting the daily operation of the system. Because of the varying natures of jobs, their equipment requirements, service durations, and times of arrival can only be described in terms of statistics. At present, as far as the state-of-the-art allows, it seems the two most effective means of analyzing a large-scale multiple-computer system are: (1) To examine the system analytically by queueing theory and (2) To simulate the system on a computer and run the simulated system sufficiently long that reliable statistics on system performance can be obtained.

The principal merit of the simulation method is that it can be applied to any system regardless of its complexity, while it is generally difficult to model a complex system by the queueing theory method or to obtain its analytical solution. However, the simulation method is not without drawbacks. The worst defect of this method is that in order to obtain reliable and meaningful statistical results, a tremendously large number of runs should be performed. This imposes a serious financial limitation on its use and requires the availability of the large amount of time spent on the computer runs. On the other hand, the queueing theory approach is analytical in nature and consequently the reliability of its results (which are usually easy to interpret) depends only on the faithfulness with which the theoretical model describes the actual system.

According to our experience in both simulation and queueing analysis in previous work, it is highly desirable to study multiple-computer systems by queueing methods because of the high reliability of their statistical conclusions.

The purpose of this report is to describe the extension of the analytical aspect of queueing theory to its synthesis aspect. That is, one not only analyzes the performance of a given multiple-computer system but also devises a scheme or a strategy to optimize the given system against a set of preassigned return or cost functions. This approach contrasts with the ordinary queueing method, and may be considered as an active queueing method in the sense that it changes the structure of the computer system to meet the current demand (state) of the system.

2. MARKOVIAN MODEL FOR QUEUEING ANALYSIS

In order to facilitate an understanding of the meaning of the transition from an analysis model to a synthesis model a brief review of a commonly used analysis model--a Markovian model--is given in this section.

The simplest Markovian model used in queueing analysis is one which possesses stationary transition probabilities. The behavior of this system is characterized by a single variable which is called the state-of-the-system. In ordinary queueing theory this state variable or state usually indicates the number of jobs waiting (both in the waiting line and in the computer proper). Mathematically, the system can be described by the following state equation

$$q(t+\Delta t) = q(t) + a(\Delta t) - s(\Delta t)$$

where

$q(t)$ = number of jobs in waiting at time t

= state variable

$a(\Delta t)$ = number of jobs arriving in time interval Δt

$s(\Delta t)$ = number of jobs completing service and leaving the system in time interval Δt.

In the above equation we use the fact that the system is a stationary system so that the number of jobs arrived in time interval Δt and the number of jobs completed in time interval Δt are independent of present time t. Otherwise, a and s would be functions of both t and Δt, i. e., $a(t, \Delta t)$ and $s(t, \Delta t)$.

Suppose that $a(\Delta t)$ and $s(\Delta t)$ are statistically independent and suppose furthermore that in time interval Δt, the probability that one job arrives is $(\Delta t)\lambda$ and that two or more jobs arrive is $0(\Delta t)$, where the symbol $0(\Delta t)$ denotes the quantity that becomes negligible compared with Δt as $\Delta t \to 0$. Similarly, assume that the probability that one job has completed the service and left in time interval Δt is $(\Delta t)\mu$ and that the probability that two or more jobs have completed the service and left is $0(\Delta t)$. Then under these assumptions it can be shown that as $\Delta t \to 0$, the probability of finding the system in any one of the states at time t satisfies the following Chapman-Kolmogorov equations:

(A) If infinite states are allowed

$$\frac{dP_o}{dt} = -\lambda P_o + \mu P_1$$

$$\frac{dP_1}{dt} = \lambda P_o - (\mu + \lambda) P_1 + \mu P_2$$

$$\vdots$$

$$\frac{dP_n}{dt} = \lambda P_{n-1} - (\mu + \lambda) P_n + \mu P_{n+1}$$

$$\vdots$$

where $P_n(t)$ = the probability that the state of the system is n at time t.

(B) If only a finite number N+1 of states are allowed

$$\frac{dP_o}{dt} = -\lambda P_o + \mu P_1$$

$$\vdots$$

$$\frac{dP_k}{dt} = \lambda P_{k-1} - (\mu + \lambda) P_k + \mu P_{k+1}$$

$$\vdots$$

$$\frac{dP_N}{dt} = \lambda P_{N-1} - \mu P_N .$$

A multiple-computer system which allows simultaneous operations may be considered as a multiplicity of service channels in parallel, and the total service rate changes as a function of state. For this system the above Chapman-Kolmogorov equations continue to hold, but with different values replacing some of the μ-coefficients. For systems which cannot be described by a Markovian queueing model for every instant of time, a consideration of the state at a sequence of special time points will often yield the system as a Markovian model. This technique is known as the imbedded Markovian chain method. In view of the above brief description of Markovian queueing models, it is clear that Markovian stochastic processes are powerful tools for queueing analysis. However, it is worthwhile to note that in queueing analysis we assume that the queueing systems studied are fixed in structure; that is, we analyze the system without attempting to change the structure of the system.

3. MARKOVIAN DECISION PROCESSES WITH CONTINUOUS-TIME PARAMETERS

If a stochastic system such as a multiple-computer system can be changed by the command of a central executive system, then obviously the following two approaches for studying this type of system are possible:

(1) Assume all possible environments for the system. Then use queueing methods to study the behavior of the system against a set of return functions under various possible sequences of commands.

(2) Assume all possible environments for the system, but use a dynamic programming method to find the sequence of commands which will optimize the system against a set of given return or cost functions.

The first approach is essentially analytical in nature and is time-consuming, because we are forced to analyze all possible sequences of commands, some of which may not be useful for actual operation. The second approach is most efficient because it allows us to select directly only the commands that are optimal.

Loosely speaking, the difference between these two types of approaches resembles the difference between analysis and synthesis techniques in circuit theory.

When the system under study is a Markovian system, the second approach (dynamic programming) is called a Markovian decision process. This process was first studied by Bellman and developed into a useful tool by Howard (Refs. 1-3). In order to make this report self-sufficient, a brief discussion on Markovian decision processes will be given in the next paragraphs.

A Markovian decision process is a generalization of an ordinary Markovian process such that the transitional probabilities of the process depend on a command signal u(t) which is a member of the set U of allowable commands. Mathematically, this process can be represented in the following manner.

Let us consider a stochastic service system which can be at any one of the states 0, 1, 2, . . . , N. Let $P_k(t)$ be the probability that the system is in kth state at time t and let $(p_{ij})\Delta t$ be the transitional probability of the system from ith state to jth state in the time interval Δt. Since every p_{ij} is assumed to be a function of the command signal u(t), it is clear that $p_{ij} = p_{ij}(u)$. Furthermore, since it is assumed that the system is a Markovian system, the probabilities of the system satisfy the following Chapman-Kolmogorov equations as $\Delta t \to 0$

$$\begin{aligned} \frac{dP_o}{dt} &= \sum_{i=1}^{N} p_{io}(u)\, P_i - [1-p_{oo}(u)]\, P_o \\ &\;\vdots \qquad\qquad\qquad\qquad\qquad\qquad u \epsilon U \\ \frac{dP_N}{dt} &= \sum_{i=0}^{N-1} p_{iN}(u)\, P_i - [1-p_{NN}(u)]\, P_N \end{aligned}$$

The process of selecting a control process $u(t) \epsilon U$ such that it maximizes or minimizes a return function is called a Markovian decision process.

4. MARKOVIAN DECISION PROCESSES WITH DISCRETE-TIME PARAMETERS

For computational ease, we often treat Markovian decision processes as discrete-time processes; that is, the time t in these processes is a member of a countable set $\{n\Delta t \,|\, n = 0,1,2,\ldots\}$ and consequently, a command signal can be applied only at each time instant $n\Delta t$. Furthermore, a discrete time process may naturally arise in modeling a system in which observations and decisions are made periodically.

The associated Chapman-Kolmogorov equations reduce to the following set of differential equations:

$$\begin{aligned} P_o[(n+1)\Delta t] &= \sum_{i=0}^{N} p_{io}(u)\, P_i(n\Delta t)\Delta t \\ &\;\;\vdots \qquad\qquad\qquad\qquad\qquad\qquad u(n\Delta t)\epsilon\, U \\ P_N[(n+1)\Delta t] &= \sum_{i=0}^{N} p_{iN}(u)\, P_i(n\Delta t)\Delta t \end{aligned}$$

The process of selecting a control process u(t) such that it maximizes or minimizes a set of return functions reduces to a sequence of controls $\{u(n\Delta t)\}$ where n = 0, 1, 2, . . . , . In subsequent text the Δt will be assumed equal to unity, without loss in generality.

5. RETURN FUNCTIONS

In the preceding two sections, return functions were mentioned in connection with the study of both discrete- and continuous-time parameter Markovian processes. As in all engineering system studies, a return function indicating "the best performance" inevitably involves a certain amount of subjective judgement. Moreover, in choosing between two different candidates for a return function, we are sometimes forced to accept the less meaningful one simply because it can be solved mathematically and the other can not.

Here we shall choose our return as a value [depending on u(t) or $u(n\Delta t)$] associated with each transition of states. For example, whenever a transition is made in time interval Δt at time instant $n\Delta t$ with control $u(n\Delta t)$, a value $r_{ij}[u(n\Delta t)]$ will be assigned to this transition if state i is the initial state and j is the final state.

The physical significance of these return functions for discrete-time cases can be explained as follows. With a discrete-time system the occurrence of a transition within a time interval Δt usually serves as a rough indication of the service efficiency or inefficiency of the multiple-computer system. For example, from a user's point of view, the queue in the system would ideally be zero at all times; i. e., the computer system would always be available. This means that a large positive value (return) should be assigned to a state transition from greater to lesser queue length, and a large negative value (return) should be assigned to a transition from lesser to greater queue length. From the point of view of the management of the multiple-computer system, however, there should always be someone in the queue to guarantee that at least part of the system is never idle; yet to maintain good will or to keep from losing customers, the queue must not be allowed to grow too long. This means that a different value (return) should be assigned to each transition of states. Consequently, in making the proper choice and assignment of values (returns) to various state transitions, a compromise must usually be made between the interest of the customers and that of the management.

In general, the return functions associated with transition of a finite state system can be expressed as a (N+1) by (N+1) square matrix function of control u(nΔt).

$$[R(u)] = \begin{bmatrix} r_{00}(u) & r_{01}(u) & \cdots & r_{0N}(u) \\ r_{10}(u) & r_{11}(u) & \cdots & r_{1N}(u) \\ \vdots & \vdots & & \vdots \\ r_{N0}(u) & r_{N1}(u) & \cdots & r_{NN}(u) \end{bmatrix}$$

For continuous-time cases, $r_{ij}(u)$ can be interpreted as a fixed rate of return during the time of a transition from state i to state j with control u(t). For a finite system, return functions again can be expressed in a (N+1) by (N+1) square matrix function of control.

6. DYNAMIC PROGRAMMING AND RECURRENCE RELATION

We are interested here only in a discrete-time parameter system. This system reduces to the continuous parameter system if $\Delta t \rightarrow 0$, but otherwise is applicable to systems which are Markovian and in which only periodic decisions can be made.

The basic tool for establishing a recurrence relation is the principle of optimality which states that an optimal sequence of controls has the property that given any initial state and initial control, the remaining controls must constitute an optimal sequence of controls with regard to the state resulting from the first control.

Using the principle cited above, we can derive the recurrence relation as follows:

Suppose that the general transition probability matrix $[P(u)]$ is known and that the return matrix $[R(u)]$ associated with the transition of states is specified. Let the system be in state i; and $f_n(i)$ be the expected optimum total return in the next n transitions. Then, by the principle of optimality, the following recurrence relations are obtained:

$$f_n(i) = \underset{u \epsilon U}{\text{Max}} \left\{ \sum_{j=0}^{N} p_{ij}(u) \left[r_{ij}(u) + f_{n-1}(j) \right] \right\}$$

$$f_o(i) = C_i \quad i = 0, 1, 2, \ldots, N \text{ and } n = 1, 2, \ldots$$

where C_i are specified boundary conditions, usually set equal to zero. If we define

$$\sum_{j=0}^{N} p_{ij}(u)\, r_{ij}(u) = b_i(u)$$

then these recurrence relations can be written as

$$f_n(i) = \underset{u \epsilon U}{\text{Max}} \left\{ b_i(u) + \sum_{j=0}^{N} p_{ij}(u)\, f_{n-1}(j) \right\}$$

$$f_o(i) = C_i \quad i = 0, 1, 2, \ldots, N \text{ and } n = 1, 2, \ldots$$

The problem now is to choose a control u at the beginning of each transition such that the sequence of controls over the total time interval $n(\Delta t)$ maximizes the total expected return of the system. If the system is allowed to operate indefinitely, then $n(\Delta t) \rightarrow \infty$ and the problem is to determine which sequence of controls maximizes the total average return over the infinite time interval. The case of optimization over an infinite time interval is closer to reality than is the case over a finite time interval, especially in computer systems, because the operating life of the system is much longer than the interval between each state transition. Therefore, the long-term return is the more important and corresponds mathematically to the asymptotic solution of the recurrence relations given above.

7. ASYMPTOTIC SOLUTION OF RECURRENCE RELATIONS

The following theorem on asymptotic behavior of the sequence $\{f_n(i)\}$ implied by Bellman in his discussion on Markovian processes has been proven in detail. Both the theorem and the proof are essentially the same as those of Bellman except for the important condition (c), where Bellman requires the much more restrictive condition $[P(u)] > 0$.

Theorem

Let us assume the following:

(a) $b_i(u)$ and $p_{ij}(u)$ are of finite dimension and are functions of vectors u, whose components assume only a finite set of values.

(b) $b_i(u) \geq 0$ for all i and all u, and $b_i(u) > 0$ for some i and all u.

(c) $[P(u)]^k > 0$, where k is some finite number, for all u. (Every element of $[P(u)]^k$ is greater than zero.)

(d) $[P(u)]$ is a stochastic matrix; i.e., $\sum_{j=1}^{N} p_{ij}(u) = 1$, $i = 0, 1, 2, \ldots, N$ and $p_{ij}(u) \geq 0$.

Under these conditions, we have the asymptotic result

$$f_n(i) \sim nr , \qquad n \to \infty , \qquad i = 0,1,2,\ldots,N \tag{1}$$

where the scalar quantity r is obtained as follows:

$$r\underline{1} = \underset{u \in U}{\text{Max}} \ \lim_{n \to \infty} \frac{\underline{b}(u) + [P(u)]\,\underline{b}(u) + \ldots + [P(u)]^{n-1}\,\underline{b}(u)}{n} \tag{2}$$

Here

$$\underline{1} = \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \qquad \underline{b}(u) = \begin{bmatrix} b_o(u) \\ b_1(u) \\ \cdot \\ \cdot \\ \cdot \\ b_N(u) \end{bmatrix} , \qquad [P(u)] = [p_{ij}(u)] \ .$$

Proof

(A) On the Markov Matrix

From the stochastic matrix theorem it can be shown that a dominant characteristic vector $\underline{1}$ exists. Furthermore, by iteration, for any vector $\underline{y}$ which is nonnegative and nontrivial, we have, for large n,

$$[P(u)]^n \ \underline{y} \sim r\,\underline{1}$$

where r is a scalar quantity depending on $\underline{y}$.

It can be shown that

$$\lim_{m \to \infty} \left(\sum_{k=0}^{m} [P(u)]^k \ \underline{y} - mr\,\underline{1} \right) = \underline{x}$$

exists.

Now let us consider the following:

$$\underline{y} + [P(u)] \left\{ \sum_{k=0}^{m} [P(u)]^k \ \underline{y} - mr\,\underline{1} \right\} = \sum_{k=0}^{m} [P(u)]^{k+1} \ \underline{y} - mr \ [P(u)] \ \underline{1} + \underline{y}$$

Since 1 is the characteristic value of $[P(u)]$ and $\underline{1}$ is its associated characteristic vector, then it follows that $[P]\ \underline{1} = \underline{1}$. This results in the relation

$$y + [P(u)] \left\{ \sum_{k=0}^{m} [P(u)]^k \underline{y} - mr\underline{1} \right\} = \sum_{k=0}^{m+1} [P(u)]^k \underline{y} - (m+1)r\underline{1} \quad + r\underline{1}$$

Now letting $m+1 \to \infty$, we obtain

$$\underline{y} + [P(u)]\ \underline{x} = \underline{x} + r\underline{1} \quad .$$

If we let $\underline{y} = \underline{b}(u)$ we obtain the following equation:

$$\underline{b}(u) + [P(u)]\ \underline{x} = \underline{x} + r\underline{1}$$

which is a result useful in the remaining proof of the theorem.

(B) Equivalence of Equations

We have the following equations:

$$\underline{b}(u) + [P(u)]\ \underline{x} = \underline{x} + r\underline{1}$$

where u is the control which maximizes the limit in Eq. 2 and

$$\underset{u \epsilon U}{\text{Max}} \left\{ \underline{b}(u) + [P(u)]\ \underline{x} \right\} = \underline{x} + r1 \quad .$$

In order to prove the equivalence of these two equations, we examine the consequence of their nonequivalence. In such a case

$$\underset{u \epsilon U}{\text{Max}} \left\{ \underline{b}(u) + [P(u)]\ \underline{x} \right\} \geqq \underline{x} + r\underline{1} \quad . \tag{3}$$

If we let $u = u^o$ be a value of u which maximizes $\underline{b}(u) + [P(u)]\ \underline{x}$, then

$$\underset{u \epsilon U}{\text{Max}} \left\{ \underline{b}(u) + [P(u)]\ \underline{x} \right\} = b(u^o) + [P(u^o)]\ \underline{x} \tag{4}$$

From Eqs. 1 and 2

$$\underline{b}(u^o) + [P(u^o)]\ \underline{x} \geqq \underline{x} + r\underline{1} \tag{5}$$

Without loss of generality we can say that, for $i = 0, 1, 2, \ldots, N$, the strict inequality occurs in the first component of the vector inequality in Eq. 5:

$$b_o(u^o) + \sum_{j=0}^{N} p_{oj}(u^o)x_j > x_o + r \tag{6}$$

$$b_i(u^o) + \sum_{j=0}^{N} p_{ij}(u^o)x_j \geq x_i + r \tag{7}$$

Equation 6 can be changed to $\geq$ by adding a positive constant a to the right-hand side:

$$b_o(u^o) + \sum_{j=0}^{N} p_{oj}(u^o)x_j \geqq x_o + (1 + a)r$$

$$x_o \leqq b_o(u^o) + \sum_{j=0}^{N} p_{oj}(u^o)x_j - (1 + a)r \tag{8}$$

Now iterate the inequalities of Eqs. 7 and 8. For Eq. 7, the following inequality results:

$$\begin{aligned} b_i(u^o) &+ \sum_{j=1}^{N} p_{ij}(u^o)\left[b_j(u^o) + \sum_{k=0}^{N} p_{jk}(u^o)\, x_k - r\right] \\ &+ p_{io}(u^o)\left[b_o(u^o) + \sum_{k=0}^{N} p_{ok}(u^o)\, x_k - (1 + a)\, r\right] \\ &\geqq r + x_i \end{aligned} \tag{9}$$

For Eq. 8, the following inequality results:

$$\begin{aligned} b_o(u^o) &+ \sum_{j=1}^{N} p_{oj}(u^o)\left[b_j(u^o) + \sum_{k=0}^{N} p_{jk}(u^o)\, x_k - r\right] \\ &+ p_{oo}(u^o)\left[b_o(u^o) + \sum_{k=0}^{N} p_{ok}(u^o)\, x_k - (1 + a)\, r\right] \\ &\geqq r + x_o + a\, r \geqq r + x_o \end{aligned} \tag{10}$$

Combining Eqs. 9 and 10 into a vector equation, we obtain the following vector inequality:

$$r\,\underline{1} + \underline{x} \leqq \underline{b}(u^o) + [P(u^o)]\ \underline{b}(u^o) + [P(u^o)]^2\ \underline{x} - r\,\underline{1} - a\ r\ [P(u^o)] \begin{pmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}$$

or

$$\underline{x} \leqq \underline{b}(u^{0}) + [P(u^{0})]\,\underline{b}(u^{0}) + [P(u^{0})]^{2}\,\underline{x} - 2r\,\underline{1} - a\,r\,[P(u^{0})] \begin{pmatrix}1\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix} \tag{11}$$

Reiterating (N=2), we obtain:

$$\underline{x} \leqq \underline{b}(u^{0}) + [P(u^{0})]\,\underline{b}(u^{0}) + [P(u^{0})]^{2}\,\underline{b}(u^{0}) + [P(u^{0})]^{3}\,\underline{b}(u^{0})$$

$$+ [P(u^{0})]^{4}\,\underline{x} - 2r\,[P(u^{0})]^{2}\,\underline{1} - ar\,[P(u^{0})]^{3} \begin{pmatrix}1\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix} - 2r\,\underline{1} - ar[P(u^{0})] \begin{pmatrix}1\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}$$

$$\underline{x} \leqq \underline{b}(u^{0}) + \sum_{\ell=1}^{3} [P(u^{0})]^{\ell}\,\underline{b}(u^{0}) + [P(u^{0})]^{4}\,\underline{x} - 4r\underline{1} - ar\left\{P(u^{0}) + [P(u^{0})]^{3}\right\} \begin{pmatrix}1\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix} \tag{12}$$

Combining Eqs. 11 and 12 and iterating (N=3), we obtain

$$\underline{x} \leqq \underline{b}(u^{0}) + \sum_{\ell=1}^{5} [P(u^{0})]^{\ell}\,\underline{b}(u^{0}) + [P(u^{0})]^{6}\,\underline{x} - 6r\underline{1} - ar\left\{[P(u^{0})]^{5} + [P(u^{0})]^{3} + [P(u^{0})]\right\} \begin{pmatrix}1\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}.$$

In general, for the Nth iteration we have the following inequality:

$$\underline{x} \leqq \underline{b}(u^{0}) + \sum_{\ell=1}^{2N-1} [P(u^{0})]^{\ell}\,\underline{b}(u^{0}) + [P(u^{0})]^{2N}\,\underline{x} - 2Nr\,\underline{1} - ar \sum_{i=1}^{2N-1} [P(u^{0})]^{2i-1} \begin{pmatrix}1\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}$$

Since $[P(u^{0})]^{k} > 0$, then for $2i - 1 \geq k$, all

$$[P(u^o)]^{2i-1} \geq \epsilon \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

where $\epsilon > 0$ is the greatest lower bound of all $p_{ij}^{(2i-1)}$ for $2i - 1 \geq k$. Therefore

$$\underline{x} \leqq \underline{b}(u^o) + \sum_{\ell=1}^{2N-1} [P(u^o)]\, \underline{b}(u^o) + [P(u^o)]^{2N}\, \underline{x} - 2Nr\underline{1} - ar\epsilon(2N - k/m_k)\underline{1}$$

where m_k is the smallest integer $\geqq (k+1)/2$.

The maximal property of $r = r(u)$ asserts that as N increases, the vector

$$\underline{b}(u^o) + [P(u^o)]\, \underline{b}(u^o) + \dots + [P(u^o)]^{2N-1}\, \underline{b}(u^o) - 2Nr\,(1 - a\epsilon)\underline{1} + ar\epsilon\ m_k\underline{1}$$

becomes an arbitrarily large negative vector. This contradicts the inequality of Eq. 13. Hence part (B) of the theorem is proved.

(C) $f_n(i) \sim nr$, for large n

Assume that $nr + x_i - k \leqq f_n(i) \leqq nr + x_i + k$ for $n = 0, 1, \dots, \ell$. Then from the principle of optimality

$$\begin{aligned} f_{\ell+1}(i) &\leqq \underset{u \in U}{\text{Max}} \left\{ b_i(u) + \sum_{j=0}^{N} p_{ij}(u)\, [\ell r + x_i + k] \right\} \\ &\leqq \ell r + k + \left[\underset{u \in U}{\text{Max}}\ b_i(u) + \sum_{j=0}^{N} p_{ij}(u) x_i \right] \\ &\leqq \ell r + k + r + x_i \ = (\ell+1)\, r + x_i + k \qquad \text{QED} \end{aligned}$$

Therefore $f_n(i) \sim nr$.

Remarks

For many finite-state queueing matrices, $[P(u)]$, the condition that $[P(u)]^k > 0$, where k is a finite constant, holds. Therefore, it follows from the above asymptotic theorem that in such a case a steady-state strategy exists.

8. A PROPERTY OF SIMPLE QUEUEING MATRIX

In this section we want to show that a simple queueing matrix has the property that $[P(u)]^k > 0$.

It can be shown for the simple queueing system with a negative exponential service time distribution and Poisson arrivals that if Δt is small, the transition matrix $[P(u)]$ for the simple queue system is merely nonnegative and only its main diagonal elements and the two adjacent elements are nonzero. For this type of matrix it can be shown by the following theorem that $[P(u)]^k > 0$ for $k > N$ where N+1 is the number of states.

Theorem

Any finite state simple queueing matrix $[P(u)]$ has the following property

$$[P(u)]^k > 0 \quad \text{for} \quad k \geq N$$

where N+1 is the order of the matrix $[P(u)]$.

Proof

For a simple queueing matrix

$$\begin{aligned} p_{ij} &> 0 \quad \text{if} \quad j = i-1, i, \ i+1 \quad \text{and} \quad 0 \leq i, \ j \leq N \\ &= 0 \text{ otherwise.} \end{aligned}$$

(For simplicity, $p_{ij}(u)$ is written as p_{ij} .)

Let us denote the elements of $[P(u)]^m$ by $p_{ij}^{(m)}$; then, for m = 2,

$$p_{ij}^{(2)} = \sum_{k=0}^{N} p_{ik}\, p_{kj} = p_{i(i-1)}\, p_{(i-1)j} + p_{ii}\, p_{ij} + p_{i(i+1)}\, p_{(i+1)j} \, .$$

Then, from the assumed property of the queueing matrix it follows that

$$\begin{aligned} p_{ij}^{(2)} &\geq 0 \quad \text{for} \quad j = i-2, \ i-1, i, \ i+1, \ i+2 \quad \text{and} \quad 0 \leq i, \ j \leq N \\ &= 0 \text{ otherwise.} \end{aligned}$$

For m = 3,

$$p_{ij}^{(3)} = \sum_{k=0}^{N} p_{ik}\, p_{kj}^{(2)} = p_{i(i-1)}\, p_{(i-1)j}^{(2)} + p_{ii}\, p_{ij}^{(2)} + p_{i(i+1)}\, p_{(i+1)j}^{(2)} .$$

From the properties of p_{ij} and $p_{ij}^{(2)}$ it can be readily shown that

$$p_{ij}^{(3)} > 0 \text{ for } j = i-3,\ i-2,\ i-1, i,\ i+1,\ i+2,\ i+3 \quad \text{and} \quad 0 \leq i,\ j \leq N$$

$$= 0 \text{ otherwise.}$$

In order to use mathematical induction, we assume that $p_{i,j}^{(m)} > 0$ for $j = i - m,\ i - (m-1),\ \ldots, i, \ldots, i + (m-1),\ i + m$, and that $p_{ij}^{(m)} = 0$ otherwise. The element $p_{ij}^{(m+1)}$ of the (m+1)th iteration of matrix [P(q)] has the following form:

$$p_{ij}^{(m+1)} = \sum_{k=0} p_{ik}\, p_{kj}^{(m)} = p_{i(i-1)}\, p_{(i-1)j}^{(m)} + p_{ii}\, p_{ij}^{(m)} + p_{i(i+1)}\, p_{(i+1)j}^{(m)}$$

On the basis of the induction hypothesis and the above iteration formula, it is evident that

$$p_{ij}^{(m+1)} > 0 \text{ for } j = i - (m+1),\ i - m, \ldots, i, \ldots, i + m,\ i + m + 1 \text{ and } 0 \leq i,\ j \leq N$$

$$= 0 \text{ otherwise.}$$

Thus the number of nonzero elements in a row of the matrix $[P(u)]^m$ is increasing with m and every iteration of the matrix [P(u)] increases the number of nonzero elements by two,. until one end of the row is filled and then by one until the entire row is filled. The extreme left nonzero element in the ith row of the $[P(u)]^m$ matrix is $p_{i(i-m)}^{(m)}$, and the extreme right nonzero elements in the same row of the $[P(u)]^m$ matrix is $p_{i(i+m)}^{(m)}$. For an (N+1)-order queueing matrix the furthest right element of $[P(u)]^m$ is $p_{0,m}^{(m)}$. If this element is nonzero then $[P(u)]^m > 0$; if $p_{0,m}^{(m)} > 0$, then m = N. Therefore,

$$[P(u)]^K > 0 \text{ for } k \geq N . \qquad \text{QED}$$

9. THE COMPUTATIONAL METHOD OF FINDING OPTIMUM CONTROLS BY POLICY ITERATION

A method of finding the asymptotic returns and optimum controls (developed by

R. Howard) is summarized in the following paragraphs.

9.1 Asymptotic Solution of a Markovian Decision Process as $n \to \infty$

It can be shown that for a given policy (not necessarily optimal) the expected return, $f_n(i)$, is given by the following recurrence equation:

$$f_n(i) = b_i + \sum_{j=0}^{N} p_{ij} f_{n-1}(j) \qquad i = 0,1,2,\dots,N$$
$$n = 1,2,3,\dots$$

Earlier we proved by the asymptotic theorem that if $[P(u)]^k > 0$, then $f_n(i) \sim nr$. Making use of this result, we can write $f_n(i)$ into the following equality form:

$$f_n(i) = nr + v_i \qquad i = 0,1,2,\dots,N \text{ as } n \to \infty$$

where r and v_i are some constants. Substitution of this asymptotic form back into the above recurrence equations yields the following equation:

$$nr + v_i = b_i + \sum_{j=1}^{N} [(n-1)\, r + v_j] p_{ij},$$

Since $\sum_{j=0}^{N} p_{ij} = 1$

$$r + v_i = b_i + \sum_{j=0}^{N} p_{ij}\, v_j \qquad i = 0,1,2,\dots,N$$

where $v_0, v_1, v_2, \dots, v_N$ and r are unknowns. Now we are facing an (N+2) unknown and (N+1) equation problem. However, if we let $v_N = 0$, we can solve for $v_0, v_1, v_2, \dots, v_{N-1}$ and r. Indeed, if a constant a is added to v_i, i.e.,

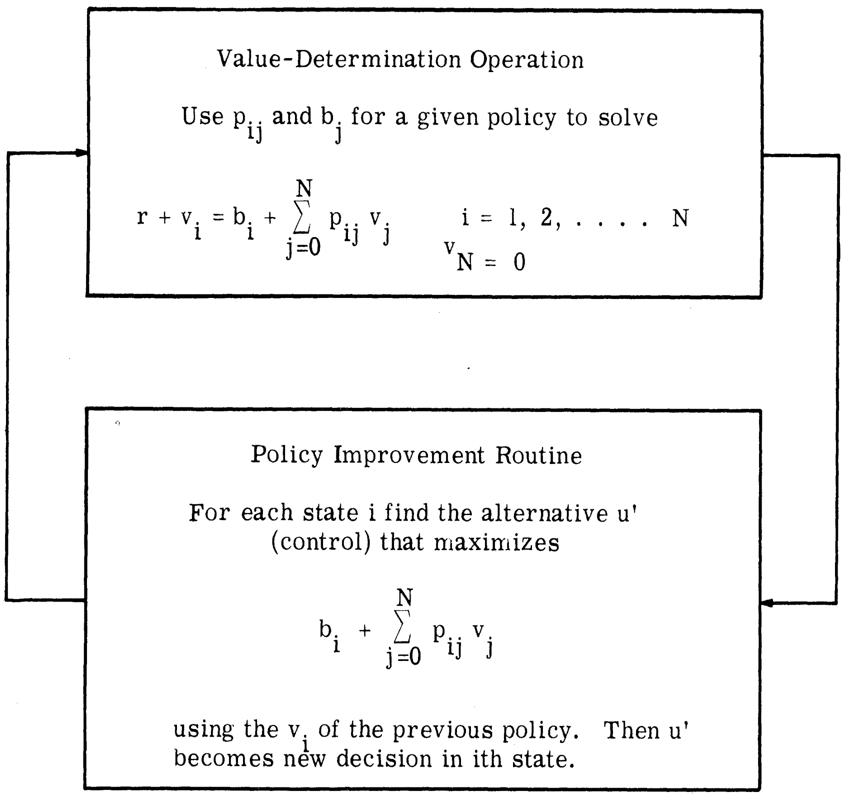
$$r + (v_i + a) = b_i + \sum_{j=0}^{N} p_{ij}(v_j + a) = b_i + \sum_{j=0}^{N} p_{ij}\, v_j + a$$

the original recurrence equation is obtained. This shows that the absolute value of v_i does not enter the calculation; only the relative value can be obtained.

9.2 Howard Iteration Cycle

The computational techniques developed by R. Howard are shown in Fig. 1.

If there is no a priori reason for selecting a particular initial policy, it is convenient to start the policy improvement routine with all $v_i = 0$. Then the starting decision will be the decision which maximizes b_i, and the routine will be stopped when the policies on two successive iterations are identical.



Value-Determination Operation

Use p_{ij} and b_j for a given policy to solve

$$r + v_i = b_i + \sum_{j=0}^{N} p_{ij} v_j \qquad i = 1, 2, \ldots . \; N$$

$$v_N = 0$$

Policy Improvement Routine

For each state i find the alternative u' (control) that maximizes

$$b_i + \sum_{j=0}^{N} p_{ij} v_j$$

using the v_i of the previous policy. Then u' becomes new decision in ith state.

Fig. 1 Howard Iteration Cycle

10. CONCLUSIONS

It has been shown that under the assumption that the transition probabilities of a queueing system are controllable, a steady-state optimal control exists which is a function of current-state only. This conclusion is important for operating a multiple-computer system because the dependence of the optimal control on current-state alone allows the management of the system to construct an optimal control table a priori.

Optimum operation of the system can be obtained by automatically observing the current-state of the system and by changing the organization of the multiple-computer system according to the optimum control table.

REFERENCES

1. R. A. Howard, Dynamic Programming and Markov Processes, Technology Press and John Wiley, New York, New York, 1960.

2. R. Bellman, Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1960.

3. R. Bellman, "A Markovian Decision Process," J. Math and Mech., Vol. 6, 1957, pp. 679-684.