Math Meth Oper Res (2002) 55:121–142

Mathematical Methods
of Operations Research

Threshold control policies for heterogeneous server systems

Hsing Paul Luh[1] **and Ioannis Viniotis**[2]

[1] Department of Mathematical Sciences, National ChengChi University, District Wen-Shan, Taipei 116, Taiwan, R.O.C. (e-mail: paul@math.nccu.edu.tw)
[2] Center for Advanced Computing and Communications, Department of Electrical Computer Engeering, NCSU, Raleigh, NC 27695 and Director of Architecture, Vitesse Semiconductor Corporation, USA (e-mail: yannis.viniotis@vitesse.com)

Manuscript received: April 2001/Final version received: September 2001

Abstract. We study the problem of optimally controlling a multiserver queueing system. Customers arrive in a Poisson fashion and join a single queue, served by N servers, $S_1, S_2, \dots, S_N$. The servers have different rates. The service times at each server are independent and exponentially distributed. The objective is to determine the policy which minimizes the average number of customers in the system. We show that any optimal, nonpreemptive policy is of threshold type, i.e., it assigns a customer to server S_i, if this server is the fastest server available and the number of customers in the queue is m_i or more. The threshold m_i may depend on the condition of other (slower) servers at the decision instant. In order to establish the results, we reformulate the optimal control problem as a linear program and use a novel argument based on the structure of the constraint matrix.

Key words: Queueing, Linear Programming, Dynamic Programming, Markov Processes

1. Introduction

We consider the queueing system which consists of N parallel heterogeneous exponential servers. Customers arrive at an infinite capacity buffer in a Poisson stream of rate λ. The buffer is served by N servers $S_1, S_2, \dots, S_N$, of different capacities, $\mu_1, \mu_2, \dots, \mu_N$, respectively. Without loss of generality, we assume that $\mu_1 > \mu_2 > \cdots > \mu_N > 0$. The service requirements are exponentially distributed, with parameter 1. Thus the time a customer spends at server S_i is exponentially distributed, with parameter μ_i. To avoid trivial cases, we assume that $0 < \lambda < \mu_1 + \mu_2 + \cdots + \mu_N$.

The motivation for studying this model comes from applications in resource allocation problems and dynamic routing in computer networks and communication systems. This problem is a generalization of the $M/M/2$ model with unequal service rates studied by Lin and Kumar (1984). They have shown that there exists an optimal policy (i.e., the one that minimizes the mean sojourn time of customers in the system), which is of threshold type. In other words, there exists an optimal policy that keeps the faster server busy, whenever possible, and activates the slower server only when the number of waiting customers exceeds a certain threshold. A simple algorithm to calculate the optimal threshold, as a function of the statistical parameters of the model, was provided in their paper. Agrawala et al. (1984) considered a related problem, with an arbitrary number, N, of servers, but no arrivals. They have shown that a threshold type policy minimizes the expected total flow time (sum of all finishing times). They also provided a simple formula to calculate the threshold for each server. An extension of this model, for "small" arrival rates is studied by Rosberg and Makowski (1990). It is shown there that threshold type policies are still optimal. Other control problems with N servers have been considered by many researchers, such as Weber (1978), Coffman et al. (1987), and Xu, Righter and Shanthikumar (1992). Bertsimas (1995) characterized the region of achievable performance in a stochastic control problem by formulating it as a linear or nonlinear programming problem on the performance vectors that all policies satisfy.

Our goal is to minimize the expected number of customers in the system, by properly selecting the customer allocation strategy. Whenever one or more servers become idle, and there are customers waiting for service, one has to decide whether to forward (or not) customer(s) for service. Idling a slower server provides one with the opportunity of finishing the customer's service earlier, through a faster server (if it becomes available.) Customer allocation decisions are not based on exact service time requirements, since we assume that this knowledge does not become available to the decision maker.

We show in this paper that there exists an optimal policy that is of threshold type, i.e., it assigns a customer to server S_i, if this server is the fastest server available and the number of customers in the queue is m_i or more. On the other hand, it may idle a server, even when there are waiting customers. The threshold m_i may depend on the condition of other (slower) servers at the decision instant.

The term 'optimal' is applied over the set of nonanticipative, nonpreemptive policies. If we allow preemptions, the optimal policy has a simple form: keep all servers busy and preempt the slowest (currently busy) server whenever a faster server becomes idle; reallocate the preempted customer to the recently available server. In computer systems preemptions may be allowed; however, in communication networks preemption of a message is typically not allowed. We were not able to utilize Dynamic Programming or Stochastic Dominance arguments, as was done by Lin and Kumar (1984), and Walrand (1984) for the special case of $N = 2$. It seems impossible to extend these arguments for the general case $N \geq 3$. The number of possible actions grows exponentially with the number of servers; Dynamic Programming arguments must deal with each action separately. Since the underlying Markov chain is multidimensional now, sample path matching, which is essential in using Stochastic Dominance arguments, is no longer possible. We use instead Linear Programming (LP) arguments to establish this result. As we show in section 4, the constraint matrix, that incorporates queue size constraints, associated with the linear program has a special structure; we are able to take advantage of this structure in order to prove the threshold type form of optimal policies. Arguments based on Linear Programming may be used in other models, where Dynamic Programming or Stochastic Dominance techniques fail.

The paper is organized as follows. In section 2 we describe the queueing model of the system and introduce some notation. In section 3 we provide the LP formulation of the optimal control problem. Finally, in section 4 we discuss the structure of optimal policies and numerical examples are given in section 5. Rather lengthy proofs of technical results are provided in an appendix.

2. The system model

The system consists of a single queue, served by N servers of unequal speed. Arrivals are Poisson, with rate λ. The service discipline is nonpreemptive. Service time at server S_i is exponential, with rate μ_i. To ensure stability of the system, we assume that $\lambda < \mu_1 + \mu_2 + \cdots + \mu_N$. As mentioned above, we may assume without loss of generality that $\mu_1 > \mu_2 > \cdots > \mu_N > 0$. Let

x_{0t} denote the number of customers waiting in queue, at time t,
x_{it} denote the busy-idle condition of server S_i, $i = 1, 2, \ldots, N$.

If $x_{it} = 0$, (resp. 1), we say that server S_i is idle (resp. busy).

Under the statistical assumptions we adopted, the $(N+1)$-dimensional vector

$$x_t \triangleq (x_{0t}, x_{1t}, x_{2t}, \ldots, x_{Nt})$$

is a suitable state description for the evolution of the queue content and of the activity level of the servers. Although x_t is a continuous-time Markov Chain, the number of customers in system is integral and sufficient to define the criterion for our problem. Then, the state space of the system is

$$X \triangleq \{0, 1, \ldots\} \times \{0, 1\}^N.$$

The decision epochs are the times of transition and described as follows. Let k be the time point when there is an arrival or a departure at which the system is observed and decisions may be made. Let T be the collection of all such k's.

For each time instant $k \in T$, there is a set X_k of all possible states, i.e., the total number of customers in the system for our models. For each $x \in X_k$ there is a set $A_{k,x}$ of feasible alternative actions, e.g., sending a customer to an idle server or not. Let $A_k = \bigcup_{x \in X_k} A_{k,x}$ be the set of possible actions at time k. Thus, X_k and A_k for our model are discrete and $X = \bigcup_k X_k$.

Although our problem is a continuous time MDP involving infinite countable number of states, state transitions and control selections take place at discrete times only. Thus we choose to consider the discrete-time problem. In specific, under the assumption we adopted, it observes:

(a) the process is statistically stationary;
(b) if the system is in state $x \in X_k$ and control $a \in A_k$ is applied, the next state will be y with probability $q(y|x, a)$;

(c) the time interval τ between the transition to state x and transition to the next state is exponentially distributed with parameter $v(x, a)$; that is the probability density function of τ is independent of earlier transition times, states, and controls;
(d) the parameters $v(x, a)$ are uniformly bounded in the sense that for $\bar{v} = \lambda + \sum_{i=1}^{N} \mu_i$, we have

$$v(x, a) \le \bar{v} < \infty, \quad \forall x, a.$$

The uniformization procedure for transforming a continuous MDP to a discrete MDP is described in Lippman (1975) in detail.

To define the discrete-time decision process for our problem, consider that at any given instant, each server is working either on a real customer, if the server is active, or on a dummy customer, if the server is idle. Dummy customers always return to the queue upon completing service and incur no contribution to delay. Transitions are associated either with arrivals or with service completions at one of the servers serving a customer – either real or dummy. These transitions occur according to a Poisson process of $\bar{v}$. A transition due to an arrival occurs with probability $\lambda/\bar{v}$, whereas a transition due to service completion at server i occurs with probability $\mu_i/\bar{v}$.

The $M/M/N$ model can be statistically described by a continuous-time Markov chain, the transitions of which occur at arrival and service completion instants. We convert a continuous time MDP problem with transition rate $v(x_k, a_k)$ and discounted factor $0 < \alpha < 1$ into a discrete-time Markov process with discounted factor

$$\beta = \frac{\bar{v}}{\alpha + \bar{v}}.$$

Notice that the action taken in every stage is unchanged during the uniformization procedure. It is well-known that a uniformizable continuous time MDP can be replaced by a discrete-time MDP if the criterion is expected discounted and average cost. The optimal stationary policies will be the same for both processes, see Serfozo (1979).

Knowledge of the system state at transition instants only, suffices to characterize the state evolution completely. Let, therefore, x_k be the system state at time t_k, when the k-th transition (i.e., arrival or service completion) occurs. Let ξ_k denote the k-th transition and let v_k, z_k denote control variables that represent actions taken at the k-th transition instant. We will define these variables shortly. The evolution of the system is described by the following equation:

$$x_{k+1} = x_k + \xi_{k+1} * z_{k+1} + v_{k+1} B, \tag{1}$$

where $x_0 \triangleq x = (x^0, x^1, \ldots, x^N)$ is the initial state of the system and B is a $(N+1)$ by N constant matrix (to be defined later.) The multiplication between ξ_k and z_k is componentwise (i.e., the ith element of the product $\xi_k * z_k$ is equal to $\xi_{ki} \cdot z_{ki}$).

The control variables $\{z_k\}$ specify whether a transition should be allowed or not; the control variables $\{v_k\}$ describe how customers are assigned to servers. For example, assume that all vectors have dimensionality one. Then, a control value $z_{k+1} = 0$ (resp. 1) could be used to disable (resp. allow) the transition ξ_{k+1}. Suppose that x_k denotes the condition of a server. Then a control value $v_k = 1$ (resp. 0) could be used to denote assignment of a customer to a server (resp. idling of the server).

A *policy* π is any (nonanticipative, nonpreemptive) rule, which at every time k decides whether to activate one or more idle servers, given that the queue is nonempty. We will refer to allocation decisions as *actions* of the policy.

Let $\|x_k\| \triangleq x_k^0 + x_k^1 + \cdots + x_k^N$ denote the total number of customers in the system (queue plus servers,) at time k. Given a discount factor $\beta > 0$ while assuming that the system starts from state $x_0 = x$, at time $k = 0$. We define the expected, β-discounted cost, incurred by policy π, as

$$J_\beta^\pi(x) \triangleq E_x^\pi\left[\sum_{k=0}^{\infty} \beta^k \|x_k\|\right], \quad x \in X \tag{2}$$

Here E_x^π denotes expectation with respect to the probability law of the process x_k, when the policy π is used and the initial state is x. A scheduling policy which is optimal for the β-discounted problem associated with (2) is called a β-optimal policy. Since the total number of customers in the system changes linearly by at most one at every transition, the optimal policy for (2) exists. We are primarily interested in the average cost criterion; however, in view of the results in the paper written by Lippman (1973), one can work with the discounted cost criterion (2) first, and then obtain the results for the average cost case by simple limiting arguments.

As mentioned in the introduction, we are interested in finding the policy which minimizes the expected number of customers in the system. From Little's law, the same policy will minimize the average time a typical customer spends in the system. We will call a policy *optimal*, if it minimizes the cost given in equation (2). Under the stability condition $\lambda < \mu_1 + \cdots + \mu_N$, there exists an optimal policy for this problem that is Markov, deterministic and stationary. This was shown by Lippman (1973).

It was not possible to extend the Dynamic Programming or Stochastic Dominance arguments in order to study optimal policies for this model. We resorted to Linear Programming based arguments instead. Equation (1) was the basis for the LP reformulation. The threshold policy, the optimality of which we want to show, stipulates that a server must be always activated when $x^0 \geq m_i$, where m_i is the optimal threshold associated with S_i. As we shall see in more detail in the next section, this property translates into the following proper of the solution of the LP: if it is optimal to have $v^i = 1$ for x^0, then it is again optimal to have $v^i = 1$ for $x^0 + 1$.

3. LP formulation

Our intention is to determine the optimal rule for selection of the control variables z_k, v_k. Under the statistical assumptions we adopted, the problem becomes a Markovian Decision Process (MDP) problem. It is well known that an MDP problem is equivalent to a linear program, possibly with an infinite number of variables. This may be referred to Ross (1983). Details of the reformulation may be also found in Rosberg, Varaiya and Walrand (1982), and Viniotis (1988).

In summary, our approach is to study the properties of the linear program one may obtain by considering all the sample paths (generated by eq. (1)) as constraints, and the cost (generated by eq. (2)) as objective function. As we will see, the constraint matrix we obtain from the sample paths of the system has a very special structure, which enables us to state and prove the threshold properties of optimal policies. In that sense, our approach differs from that given by Ross (1983), for example.

In order to avoid technical difficulties that arise when one considers linear programs with an infinite number of variables (namely, duality gaps,) we will consider a finite horizon problem first. Throughout this discussion, the horizon has a fixed value n, and thus only a finite number of variables is involved. We then will derive the results for (the discrete version of) cost functional (2) by letting the horizon approach infinity.

We describe now the LP reformulation. The idea is similar to that presented by Rosberg, Varaiya and Walrand (1982). We need a few definitions first. Let ω_k represent the kth transition of the queueing system (i.e., the kth jump of the Markov chain). Let $\Omega \triangleq \{A, D_1, \ldots, D_N\}$ be the set of all transitions. Here A represents an arrival and D_i represents a departure from server S_i (potentially a "dummy" one, if the server is idle.) Denote the basic sample space for the MDP problem as

$$\Omega^n \triangleq \{\omega^n \triangleq (\omega_1, \omega_2, \ldots, \omega_n) : \omega_i \in \Omega\}.$$

We define ω^k and Ω^k for $2 \le k < n$ in a similar manner. When a particular transition, say the lth one, of an element ω^k of Ω^k has to be singled out, ω^k will be alternately denoted as $\omega^{k-1}\omega_l \ldots$. Let $Pr(\cdot)$ denote the probability distribution on Ω^n.

Let F_n denote the σ-field generated by a random variable f_n and consist of all ω^n. Define a *process* f as a sequence of random variables $f = (f_1, f_2, \ldots, f_n)$, where f_k is F_k-measurable for every k. We can regard f_k as a function on Ω^k.

Let $\xi(\omega)$ represent the change in the system state, incurred by transition ω. The function ξ is given by

$$\xi(\omega) = \begin{cases} (1, 0, \ldots, 0), & \omega = A, \\ (0, -1, \ldots, 0), & \omega = D_1, \\ \vdots & \\ (0, 0, \ldots, -1), & \omega = D_N. \end{cases}$$

Define the *transition process* $\hat{\xi}$ as

$$\hat{\xi} = (\xi_1, \xi_2, \ldots, \xi_n),$$

where $\xi_k(\omega^k) \triangleq \xi(\omega_k)$.

Define a *transition* (or *multiplicative*) control, as the process

$$z = (z_1, z_2, \ldots, z_n)$$

Here $z_k(\omega^k) \triangleq z_k(\omega_k)$ and each $z_k(\omega^k)$ is a $N+1$-dimensional vector,

$$z_k(\omega^k) \triangleq (z_k^0(\omega^k), z_k^1(\omega^k), \dots, z_k^N(\omega^k)),$$

where

$z_k^0(\omega^k)$ controls transitions of the queue size, and
$z_k^i(\omega^k)$ controls transitions of server S_i, $i = 1, \dots, N$.

As mentioned before, $z_k^i(\omega^k)$ specifies whether the transition ω_k is allowed to affect the ith element of the state vector or not. Since in this model arrivals are not controlled, we have

$$z_k^0(\omega^k) = \begin{cases} 1, & \omega_k = A, \\ 0, & \omega_k \neq A. \end{cases} \tag{3}$$

It is more natural to define allocation of customers to idle server(s) as the only control in this system. The standard uniformization procedure shown by Lippman (1975), which is used to convert the problem to a discrete time one, however, forces us to introduce these "artificial" control variables at departure instants. Thus, departures are controlled and we let in general

$$z_k^i(\omega^k) = \begin{cases} \in [0,1], & \omega_k = D_i, \ i = 1, \dots, N, \\ 0, & \omega_k \neq D_i. \end{cases} \tag{4}$$

Define a *server allocation* (or *additive*) control, as the process

$$v = (v_1, v_2, \dots, v_n).$$

Here $v_k(\omega^k) \triangleq v_k(\omega_k)$ and each $v_k(\omega^k)$ is a N-dimensional vector,

$$v_k(\omega^k) \triangleq (v_k^1(\omega^k), \dots, v_k^N(\omega^k)),$$

where $v_k^i(\omega^k)$ acts on the condition of server S_i, $i = 1, \dots, N$. We have

$$v_k^i(\omega^k) \in [0,1] \quad \omega_k \in \{A, D_1, \dots, D_N\}. \tag{5}$$

Remark: It seems "more natural" to require that $z_k^i(\omega^k), v_k^i(\omega^k) \in \{0,1\}$, since we either block or not a transition, assign or not a customer to a server. By allowing these variables to take fractional values, we "enlarge" the action space; in fact, from eq. (2) or (6) below, we can see that we allow for fractional customers in the queueing system. This enlargement of the action space is necessary for the linear program formulation of the problem. Without it, the problem becomes an Integer Programming problem. However, this enlargement will be legitimate, if we show that the resulting linear program admits an integer-valued solution (see Lemma 1).

Define next the *trajectory* $\hat{x}$, that corresponds to an initial state x, as the process

$$\hat{x} = (x_1, x_2, \dots, x_n),$$

where $x_0 \triangleq x$ and

$$x_{k+1}(\omega^{k+1}) = x_k(\omega^k) + z_{k+1}(\omega^{k+1}) * \xi_{k+1}(\omega^{k+1}) + v_{k+1}(\omega^{k+1})B. \tag{6}$$

In scalar notation, the queue size evolves as

$$x^0_{k+1}(\omega^{k+1}) = x^0_k(\omega^k) + \xi^0_{k+1}(\omega^{k+1}) z^0_{k+1}(\omega^{k+1}) - v^1_{k+1}(\omega^{k+1}) - \cdots - v^N_{k+1}(\omega^{k+1}), \tag{6a}$$

(i.e., the queue size at the next "event" ω_{k+1} is given by the previous queue size plus one, if ω_{k+1} was an arrival, minus the number of customers allocated to servers.) The server condition evolves as

$$x^i_{k+1}(\omega^{k+1}) = x^i_k(\omega^k) + \xi^i_{k+1}(\omega^{k+1}) z^i_{k+1}(\omega^{k+1}) + v^i_{k+1}(\omega^{k+1}), \quad i = 1, \ldots, N. \tag{6b}$$

(i.e., the next state of server S_i is its previous one minus 1, if ω_{k+1} was an (allowed) departure from this server, plus an allocated customer, if any).

In eq. (6), matrix B has the form

$$B \triangleq \begin{pmatrix} -1 & +1 & \ldots & 0 \\ -1 & 0 & \ldots & 0 \\ & & \ddots & \\ -1 & 0 & \ldots & +1 \end{pmatrix}$$

i.e., the first column contains entries that are all equal to -1; the remaining columns form an identity matrix.

We will refer to the vector (z, v) as a *policy*. Suppose that the system starts from state x at time $k = 0$, and is allowed to "move" for n steps (i.e., perform n transitions). Define the (β-discounted, finite horizon, expected) cost of policy (z, v), as

$$J_n(x, z, v) \triangleq E \sum_{k=0}^{n-1} \beta^k (x^0_k + x^1_k + \cdots + x^N_k), \tag{7}$$

We can rewrite the state trajectory equation (6) as a function of the initial state as

$$x^0_{k+1}(\omega^{k+1}) = x^0 + \sum_{j=1}^{k+1} z^0_j(\omega^j)\xi^0_j(\omega^j) - \sum_{j=1}^{k+1} v^1_j(\omega^j) - \cdots - \sum_{j=1}^{k+1} v^N_j(\omega^j)$$

$$x^i_{k+1}(\omega^{k+1}) = x^i + \sum_{j=1}^{k+1} z^i_j(\omega^j)\xi^i_j(\omega^j) + \sum_{j=1}^{k+1} v^i_j(\omega^j), \quad i = 1, \ldots, N.$$

Then, we may rewrite the cost in eq (7) as

$$J_n(x,z,v) = (1+\cdots+\beta^{n-1})(x^0+x^1+\cdots+x^N) + \sum_{k=1}^{n-1}\sum_{\omega^k\in\Omega^k}[\gamma_k^0(\omega^k)z_k^0(\omega^k) + \gamma_k^1(\omega^k)z_k^1(\omega^k)+\cdots+\gamma_k^N(\omega^k)z_k^N(\omega^k)], \tag{7a}$$

where the constants $\gamma_k^i(\omega^k)$ depend on system parameters only. More specifically,

$$\gamma_k^i(\omega^k) \triangleq Pr(\omega^k)(\beta^k+\cdots+\beta^{n-1})\xi_k^i(\omega^k), \quad i=0,1,2,\ldots,N.$$

For example, we have

$$\gamma_k^0(\omega^k) = \begin{cases} Pr(\omega^k)(\beta^k+\cdots+\beta^{n-1}), & \omega_k = A, \\ 0, & \omega_k \neq A. \end{cases} \tag{8}$$

$$\gamma_k^i(\omega^k) = \begin{cases} -Pr(\omega^k)(\beta^k+\cdots+\beta^{n-1}), & \omega_k = D_i,\ i=1,\ldots,N, \\ 0, & \omega_k \neq D_i. \end{cases} \tag{9}$$

The reader may consult Viniotis (1988) for more details. Since we do not control arrivals, we get from equation (7a) that the cost associated with policy (z,v) can be rewritten as

$$J_n(x,z,v) = (1+\cdots+\beta^{n-1})(x^0+x^1+\cdots+x^N) + \sum_{k=1}^{n-1}(\beta^k+\cdots+\beta^{n-1})\sum_{\substack{\omega^k\in\Omega^k\\ \omega_k=A}} Pr(\omega^k) + \sum_{k=1}^{n-1}\sum_{\substack{\omega^k\in\Omega^k\\ \omega_k=D_1}}\gamma_k^1(\omega^k)z_k^1(\omega^k) + \cdots + \sum_{k=1}^{n-1}\sum_{\substack{\omega^k\in\Omega^k\\ \omega_k=D_N}}\gamma_k^N(\omega^k)z_k^N(\omega^k). \tag{10}$$

Hence the optimal (z,v) policy is the one which minimizes the right hand side of equation (10). Notice that the controls $\{v_k^i(\omega^k)\}$, which denote customer assignment to the servers, do not appear directly in eq. (10); they do appear in the constraints, however. The minimization is constrained, since for example a policy which results in negative queue sizes is not feasible.

Let $|a|$ denote the absolute value of the real number a. Since the coefficients $\gamma_k^i(\omega^k)$ are nonpositive, the optimal policy can be found as the solution to the following linear program:

$$\max_{\{z_k^i(\omega^k)\},\{v_k^i(\omega^k)\}} \sum_{k=1}^{n-1} \sum_{\substack{\omega^k \in \Omega^k \\ \omega_k = D_1}} |\gamma_k^1(\omega^k)| z_k^1(\omega^k)$$

$$+ \cdots + \sum_{k=1}^{n-1} \sum_{\substack{\omega^k \in \Omega^k \\ \omega_k = D_N}} |\gamma_k^N(\omega^k)| z_k^N(\omega^k) \tag{11}$$

The constraints of the linear program (11) are the following:

For each element $\omega^k \in \Omega^k$, where $1 \le k \le n$, the selected policy (z, v) must result in:

(a) Nonnegative queue size,

$$x^0 + \sum_{j=1}^{k} z_j^0(\omega^j)\xi_j^0(\omega^j) - \sum_{j=1}^{k} v_j^1(\omega^j) - \cdots - \sum_{j=1}^{k} v_j^N(\omega^j) \ge 0 \tag{12}$$

(b) Nonnegative server condition,

$$x^i + \sum_{j=1}^{k} z_j^i(\omega^j)\xi_j^i(\omega^j) + \sum_{j=1}^{k-1} v_j^i(\omega^j) \ge 0, \quad i = 1, \ldots, N. \tag{13}$$

(c) Not "overfull" server condition,

$$x^i + \sum_{j=1}^{k} z_j^i(\omega^j)\xi_j^i(\omega^j) + \sum_{j=1}^{k-1} v_j^i(\omega^j) + v_k^i(\omega^k) \le 1, \quad i = 1, \ldots, N. \tag{14}$$

Notice that two separate conditions are imposed on the server state x_{k+1}^i: one before an action is taken (relation 13) and one after an action is taken (relation 14). A single condition like

$$1 \ge x_{k+1}^i(\omega^{k+1}) \ge 0, \tag{15}$$

does not suffice, since we may possibly have that

$$x_k^i(\omega^k) + \xi_{k+1}^i(\omega^{k+1}) z_{k+1}^i(\omega^{k+1}) = -1 \quad \text{and} \quad v_{k+1}^i(\omega^{k+1}) = 1,$$

which fulfills condition (15), but is clearly not feasible.

From eqs. (6a)–(6b), and the constraints (a), (b), (c) imposed on queue size and server conditions, it is easy to see that taking $z_k(\omega^k) = z_k(\omega_k)$ in section 3, amounts to restricting our attention to Markovian policies only. This is no loss of generality, however, since it is known that for this problem there exists an optimal policy which is Markovian.

The linear program (11), with constraints as in (12)–(14), is the basis for the results we present in section 4. The key idea is to study the structure of the constraint set and show that the linear program admits an optimal solution of a specific form.

4. The structure of optimal policies

Observe that in the linear program (11), only the $z_k^i(\omega^k)$ variables appear in the objective function. Thus we will at first focus our attention to constraints (13)–(14). The variables $v_k^i(\omega^k)$, which appear in constraint (12) (and of course constraints (13)–(14),) will be treated initially as fixed parameters, the value of which is simply chosen to satisfy the constraints. Later we will determine their optimal values.

Suppose that the system is in state $(x^0, x^1, \ldots, x^N)$ at time $t = 0$. This initial state is fixed throughout this discussion. A decision is made to allocate a customer to server S_i, and thus the system moves into state $(x^0 - \sum_{i=1}^{N} v^i, x^1 + v^1, \ldots, x^N + v^N)$. From eqs (13) and (14), we see that (for a fixed value of the variables $v_k^i(\omega^k)$,) the optimal selection of $\{z_k^1(\omega^k), \ldots, z_k^N(\omega^k)\}$ variables should satisfy the constraints

$$v^1 + \sum_{l=1}^{k-1} v_l^1(\omega^l) + v_k^1(\omega^k) - 1 \leq \sum_{\substack{l=1 \\ \omega_l = D_1}}^{k} z_l^1(\omega^l) \leq v^1 + \sum_{l=1}^{k-1} v_l^1(\omega^l) \tag{16}$$

$$\vdots$$

$$v^N + \sum_{l=1}^{k-1} v_l^N(\omega^l) + v_k^N(\omega^k) - 1 \leq \sum_{\substack{l=1 \\ \omega_l = D_N}}^{k} z_l^N(\omega^l) \leq v^N + \sum_{l=1}^{k-1} v_l^N(\omega^l) \tag{17}$$

Of course, the customer allocation variables $\{v_k^1(\omega^k), \ldots, v_k^N(\omega^k)\}$ should satisfy the additional constraints

$$\sum_{j=0}^{k} v_j^1(\omega^j) + \cdots + \sum_{j=0}^{k} v_j^N(\omega^j) \leq x^0 + \sum_{j=1}^{k} \xi_j^0(\omega^j), \tag{18}$$

$$v^i \leq 1 - x^i, \quad i = 1, \ldots, N. \tag{18a}$$

In the following three Lemmas we investigate the structure of the constraints and of the optimal solution. The proof of Lemma 1 is given in the appendix.

Lemma 1: *The constraints* $z_j^i(\omega^j), v_j^i(\omega^j) \in [0, 1]$, *and the constraints in ineqs. (12), (13), (14), (18a) form a totally unimodular matrix.*

Remark: An immediate consequence of Lemma 1 is that the optimal solution of the linear program (11) is integer-valued. A similar proof can be found in Nemhauser and Wolsey (1988). The "enlargement" of the action space, and the reformulation of the problem as a linear program, instead of a more natural integer program, is therefore justified.

We investigate next the relationship between the optimal z variables and the chosen v values. This relationship (see Lemma 2,) is the key step towards formulating a new LP, with a more simplified, suitably structured constraint matrix.

Naturally, for each server the assignments between two consecutive departures can be only made at most once. Let ω_p denote the pth transition in the sample path ω^{j-1}. Define $l^*(\omega^{j-1})$ as

$$l^*(\omega^{j-1}) \triangleq \max_p \{p \leq j-1 : \omega_p = D_i\},$$

or 0 if such a p does not exist. Also, we define $v_0^i(\omega^0) = v^i$. Let $\not{D}_i$ denote any transition that is not a departure from server S_i. Then, for a given i, $1 \leq i \leq N$, and ω^j, where $\omega_j = \not{D}_i$, we must have that

$$\sum_{k=l^*(\omega^{j-1})}^{j} v_k^i(\omega^k) \leq 1. \tag{19}$$

Lemma 2: *Suppose that $\{v_k^i(\omega^k)\}$ are chosen to satisfy constraints (16), (17), (18) and (18a). Then the optimal solution $\{\tilde{z}_k^i(\omega^k)\}$ of program (11) is given by:*

$$\tilde{z}_j^i(\omega^{j-1}D_i) = \sum_{k=l^*(\omega^{j-1})}^{j-1} v_k^i(\omega^k).$$

The lower bound of the summation in eq. (19) is as explained above. Intuitively, at time j, the transition control "clears out" the customer that has been assigned to server S_i, since the last time that server became idle. One can verify, with a little algebra, that the optimal value of $\tilde{z}_j^i(\cdot)$ in (19) satisfies the right hand side of inequality (16)–(17) with equality.

Observe that since

$$\sum_{k=l^*(\omega^{j-1})}^{j-1} v_k^i(\omega^k) + v_j^i(\omega^{j-1}\not{D}_i) \leq 1,$$

we have that $\sum_{k=l^*(\omega^{j-1})}^{j-1} v_k^i(\omega^k) \leq 1$, and thus $0 \leq \tilde{z}_j^i(\omega^{j-1}D_i) \leq 1$, which does not violate the constraints $z_j^i(\omega^j) \in [0,1]$.

We are now ready to express the original objective function in (11), in terms of the server allocation variables $\{v_k^i(\omega^k)\}$ only. In order to establish a suitable structure for the constraint matrix, let v denote a vector that contains all the variables $\{v_k^i(\omega^k)\}$, for all k, ω^k. A convenient ordering of these variables, that gives matrix A a special structure, can be derived in the following way:

Order first the set Ω as $\{A, D_1, \ldots, D_N\}$. Order Ω^k, $k > 1$ as follows: fix an element of the already ordered set Ω^{k-1}. To that element, append in sequence, all elements of Ω, in order to generate Ω^k. For example, the ordered space Ω^2 is given below

$$\Omega^2 = \{AA, D_1A, \ldots, D_NA, AD_1, \ldots, D_ND_1, AD_N, \ldots, D_ND_N\}.$$

Define $c_j^i(\omega^j)$ as the coefficient of the variable $v_j^i(\omega^j)$ in the objective function (20). Let c denote the corresponding cost coefficient vector. Then, the optimal customer assignment variables are given by the solution of the following linear program:

$$\max_{\{v_j^i(\omega^j)\}} \quad c \cdot v \tag{20}$$

$$\mathbf{A} \cdot v \leq b(x^0) \tag{21}$$

$$0 \leq \sum_{k=l^*(\omega^{j-1})}^{j} v_k^i(\omega^k) \leq 1$$

$$\text{where} \quad 1 \leq i \leq N, \quad \omega^j = \omega^{j-1}\not{D}_i.$$

The value of $c_j^i(\omega^j)$ in terms of the original cost coefficients $\gamma_k^i(\omega^k)$ is given in the following Lemma. However, its proof can be derived by straightforward algebraic manipulations.

Lemma 3: *The coefficient* $c_j^i(\omega^j)$*, associated with the variable* $v_j^i(\omega^j)$*, in the linear program (20), is given by*

$$c_j^i(\omega^j) = Pr(\omega^j) \sum_{k=j+1}^{n} \beta^k [1 - (1 - Pr(D_i))^{k-j}].$$

Moreover,

$$c_j^i(\omega^j) > \sum_{\omega} c_{j+1}^i(\omega^j \omega).$$

Note: The expression for $c_j^i(\omega^j)$ is really just $c_j^i(\omega^j) = E(\beta^{j+X_i} + \cdots + \beta^n)$, where X_i is the service time of a job assigned to server S_i at time j. We interpret this sum to be 0 if $j + X_i > n$. The formulation of the objective function, (20), should be understood as follows. The idea is that if customer p arrives a time j and then sits in the queue until time n, its total discounted holding cost is $\beta^j + \cdots + \beta^n$. So the problem of minimizing total expected discounted holding costs is equivalent to one in which we pay this cost at the start and then try to maximize a compensating total expected reward, where a reward of $\beta^{j+X_i} + \cdots + \beta^n$ accrues when one assigns a customer to server S_i at time j.

The form of the vector b and the constraint matrix $\mathbf{A}$ are constructed as follows. In ineq. (21) we have one constraint for every k and ω^k, where $1 \leq k \leq n$ and $\omega^k \in \Omega^k$. A convenient form for the constraint matrix $\mathbf{A}$ is obtained if we list these constraints in n groups, with the constraints for Ω first and the constraints for Ω^n last. Within group k, the order is specified by the enumeration of Ω^k. The form of $\mathbf{A}$ for the special case $N = 2$, $n = 3$ is given in section 4. The form of the right hand side vector $b(x^0)$ is obtained as follows: consider the constraint for the sample path ω^k. From eq (18) we have immediately that the corresponding element, $b_i(x^0)$ of $b(x^0)$ where i is associated with a specific ω^k, is given by,

$$b_i(x^0) = x^0 + \sum_{j=1}^{k} \xi_j^0(\omega^j)$$

For more details about systematic orderings of $b, c, \mathbf{A}$, for more complex queueing systems, the reader is referred to Viniotis (1988).

For the purposes of this discussion, it suffices to mention that the right hand side vector b is a function of the initial queue size x^0. Observe that constraints (16)–(17) of the original LP are already satisfied by the solution in (19). Thus, the constraints we have to consider, are the queue size constraints (21) and the server constraints (19). In these constraints, only server allocation variables appear. The set of constraints in (21) has a special "block diagonal" structure. According to this structure, constraints (19) is easy to be calculated with direct algebra. This structure can be visualized in terms of a block diagonal matrix I, which we define below. Let $\varepsilon \triangleq (1, \ldots, 1)$, i.e., let ε be a row vector with all N entries equal to 1. Define the block diagonal identity matrix I as follows:

$$I = \begin{pmatrix} \varepsilon & & & \\ & \varepsilon & & \\ & & \ddots & \\ & & & \varepsilon \end{pmatrix}$$

Matrix I has $N + 1$ rows (i.e., as many as the number of state transitions,) and $N \cdot (N + 1)$ columns. The form of the constraint matrix $\mathbf{A}$ for a general horizon is easily derived by its special structure. We show here the matrix for the simple to visualize case of $N = 2$ servers, and a horizon $n = 3$. Entries that are not shown are equal to zero.

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & & & & & & & & & & & & & \\ \mathbf{1} & \mathbf{1} & I & & & & & & & & & & & & \\ \mathbf{1} & \mathbf{1} & I & I & & & & & & & & & & & \\ \mathbf{1} & \mathbf{1} & I & 0 & I & & & & & & & & & & \\ \mathbf{1} & \mathbf{1} & I & 0 & 0 & I & & & & & & & & & \\ \mathbf{1} & \mathbf{1} & I & I & 0 & 0 & I & & & & & & & & \\ \mathbf{1} & \mathbf{1} & I & 0 & I & 0 & 0 & I & & & & & & & \\ \mathbf{1} & \mathbf{1} & I & 0 & 0 & I & 0 & 0 & I & & & & & & \\ \mathbf{1} & \mathbf{1} & I & I & 0 & 0 & 0 & 0 & 0 & I & & & & & \\ \mathbf{1} & \mathbf{1} & I & 0 & I & 0 & 0 & 0 & 0 & 0 & I & & & & \\ \mathbf{1} & \mathbf{1} & I & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & I & & & \\ \mathbf{1} & \mathbf{1} & I & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & & \\ \mathbf{1} & \mathbf{1} & I & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & \\ \mathbf{1} & \mathbf{1} & I & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{pmatrix} \tag{22}$$

In eq (22), a 0 represents a matrix with all entries equal to zero; a boldface 1 represents a column vector with both (in general all N) entries equal to one. Observe that the first 2 (in general N) columns of $\mathbf{A}$ correspond to the assignment variables v^i, i.e., the initial server allocations.

Before the main results is derived, we will make use of the following intuitive Lemma.

Lemma 4: *There exists an optimal policy that activates faster than slower servers, whenever possible.*

Proof: Suppose that at some time k, $0 \le k \le n-1$, it is feasible to activate server S_i or S_j; in other words, both $v_k^i(\omega^k) = 1$ and $v_k^j(\omega^k) = 1$ are feasible. Suppose that server S_i is faster than server S_j, i.e., $\mu_i > \mu_j$. From Lemma 3, since $Pr(D_i) > Pr(D_j)$, we have that

$$c_k^i(\omega^k) > c_k^j(\omega^k). \tag{23}$$

Consider two vectors s and $\bar{s}$, such that

$$s_k^i(\omega^k) = 1, \quad s_k^j(\omega^k) = 0,$$

$$\bar{s}_k^i(\omega^k) = 0, \quad \bar{s}_k^j(\omega^k) = 1,$$

that have all other elements equal. The form of matrix A (eq. (22),) suggests that the variables $v_k^i(\omega^k)$ and $v_k^j(\omega^k)$ appear together in every constraint. Therefore, both vectors s and $\bar{s}$ are feasible solutions. From ineq. (23), we can immediately see that solution s gives a larger objective function value. Therefore, a faster rather than a slower server should be activated whenever possible. q.e.d.

Based on this lemma, we derive a maximal expected reward at each time. Consider a customer is assigned to server S_i at time j if S_i is the fastest server available at time j. According to eq (20), this will lead to the customer obtain a expected reward $c_j^i(\omega^j)$. However, if he is held in queue and considered to obtain a maximal future reward at time $j+1$, i.e. he will be assigned to any faster server available at time $j+1$ despite that S_i is idle at time j. The expected maximal reward at time $j+1$, $\mathscr{R}_j^i$, is

$$\sum_{l=1}^{i} c_{j+1}^l(\omega^j D_l) + \sum_{l=i+1}^{N} c_{j+1}^i(\omega^j D_l) + c_{j+1}^i(\omega^j A).$$

Generally, if $c_j^i(\omega^j) \ge \mathscr{R}_j^i$, the customer should be assigned to S_i at time j for an optimal policy, because by Lemma 3 the relation $c_k^i(\omega^k) \ge \mathscr{R}_k^i$ holds for every $k > j$. However, if $c_j^i(\omega^j) < \mathscr{R}_j^i$, the optimal policy is not to activate S_i at time j, but may activate it later which can be some time $k > j$. Notice that if server S_i is not activated at time j, it is not necessary to be activated at time $j+1$, because the relation $c_{j+1}^i(\omega^{j+1}) < \mathscr{R}_{j+1}^i$ may also be applied. Though, both $c_j^i(\omega^j)$ and $\mathscr{R}_j^i$ are not functions of the initial queue sizes, the initial queue sizes play a role in decisions such that the server S_i may be activated at time k where $j < k < n$. This will be explained in the following lemma by looking into the structure of $\mathbf{A}$ with eq (19) and eq (21). Moreover, the time k is non-increasing with the initial queue sizes. We are now ready to associate properties of the optimal customer allocation policy, to properties of the constraint matrix $\mathbf{A}$. Let $\tilde{v}$ denote the optimal policy when the initial queue size is x^0 and server S_i is free. Let $\bar{v}$ denote the optimal policy when server S_i is again free but the initial queue size is $x^0 + 1$. The particular structure of $\mathbf{A}$ is essential in the proof of the following fundamental result:

Lemma 5: *Suppose that for initial queue size x^0, and a given server configuration, it is optimal to have $\tilde{v}^i = 1$, i.e., activate server S_i. Then, for queue size $x^0 + 1$ and the same server configuration, it is optimal to have $\bar{v}^i = 1$.*

Proof: We prove it by considering the following two cases.

Case 1: If $c^i \geq \mathscr{R}_0^i$, then clearly the optimal policy $\bar{v}$ should activate S_i, i.e. $\bar{v}^i = 1$ since $\tilde{v}$ activates S_i, i.e. $\tilde{v}^i = 1$.

Case 2: If $c^i < \mathscr{R}_0^i$, then assume $\bar{v}^i$ is equal to 0 when the initial queue size is $x^0 + 1$, while $\tilde{v}^i = 1$ when the initial queue size is x^0. From Lemma 4, policy $\bar{v}$ may also idle other servers, faster than server S_i. Let S_{i_1} be the fastest such server, where $i_1 \leq i$. Suppose therefore, that $\bar{v}^j = 0$, for $i_1 \leq j$. We will arrive at a contradiction, derived from this assumption.

Since $\tilde{v}$ is optimal and $\tilde{v}^{i_1} = 1$, there are at least two customers are available to be assigned at some $k > 0$. We will first show that server S_{i_1} must be activated for at least one ω^k, with $1 \leq k \leq n-1$. To see this, let us check the feasibility and optimality of the assignment $\bar{v}_k^{i_1}(\omega^k) = 1$, for some $k > 0$. Clearly, if $\bar{v}_k^{i_1}(\omega^k) = 0$, for all k and all ω^k, $1 \leq k \leq n-1$, the objective function can be improved. Consider for example any path ω^{n-1}. Since policy $\bar{v}$ idles server S_j, $j \geq i_1$ at time 0, it must idle them at all times $k > 0$ also. Thus, along the path ω^{n-1}, we have

$$\mathrm{A} \triangleq \sum_{i=1}^{N} \tilde{v}^i + \sum_{i=1}^{N} \sum_{j=1}^{n-1} \tilde{v}_j^i(\omega^j) > \sum_{i=1}^{N} \bar{v}^i + \sum_{i=1}^{N} \sum_{j=1}^{n-1} \bar{v}_j^i(\omega^j) \triangleq \mathrm{B}.$$

However, since $\mathrm{A} \leq x^0 + \sum_{j=1}^{n-1} \xi_j^0(\omega^j)$, we conclude that $\mathrm{B} \leq x^0 + 1 + \sum_{j=1}^{n-1} \cdot \xi_j^0(\omega^j)$, and thus letting $\bar{v}_{n-1}^{i_1}(\omega^{n-1}) = 1$ is feasible.

Consider a particular event u^{k_1} where the integer k_1 has the *smallest* possible value such that $\bar{v}_{k_1}^{i_1}(u^{k_1}) = 1$, but $\bar{v}_k^{i_1}(\omega^k) = 0$, for all $\omega^k \in \Omega^k$ and $k < k_1$. Furthermore, B is strictly less than A, we have that $\bar{v}_{k_1}^{i_1}(u^{k_1-1}\omega) = 1$, for every $\omega \in \Omega$. From this fact and Lemma 3, we will show that the value of k_1 can be decreased (i.e., activation of server S_{i_1} must be done earlier). By induction, it follows that policy $\bar{v}$ must activate server S_{i_1} at time $k = 0$, and we arrive at the desired contradiction.

It follows that $\bar{v}$ must activate server S_i at time $k = 0$. Using the same arguments, we may further improve the optimal solution by induction which is

$$\bar{v}^{i_1+1} = \bar{v}^{i_1+2} = \cdots = \bar{v}^i = 1, \quad q.e.d.$$

Essentially Lemma 5 says that there exists an optimal policy that is of threshold type; note that the optimal threshold for server S_i may depend on the condition of the other servers. The next theorem follows immediately from Lemma 5.

Theorem: *There exists an optimal policy, for the discounted, finite horizon problem, that is of threshold type, with thresholds that depend on (slower) server condition.*

We now extend this result for the average cost case. Let $m_i(n)$ denote the optimal threshold for server S_i, when the horizon is n. We first show that the optimal thresholds are nondecreasing functions of the horizon, and thus they possess a limiting (finite) value.

Lemma 6: *The optimal threshold for server S_i is a nondecreasing, bounded function of the horizon n.*

Proof: When the horizon increases (to $n+1$), the basic linear program (20) is affected as follows:

- $N \cdot |\Omega|^{n+1}$ more variables of the form $v^i_{n+1}(\omega^{n+1})$ are introduced.
- All the cost coefficients are increased (by a factor $\beta^{n+1}[1-(1-Pr(D_i))^{n+1-j}]$.)
- The constraint matrix A has now $|\Omega|^{n+1}$ more rows and $N \cdot |\Omega|^{n+1}$ more columns.

Consider an initial state $(x^0, \ldots, 0, \ldots)$ in which it is optimal to activate server S_i, when the horizon is n. It is of course possible that $c^j_{n+1}(\omega^{n+1}) > c^i$, for some $1 \leq j \leq N,\ j \neq i$. Consider now a feasible solution s with the following property:

- $s^i = 0$,
- $s^j_{n+1}(\omega^{n+1}) = 1$,
- all other elements of s are equal to those of $\tilde{v}$, where $\tilde{v}$ is an optimal solution for the LP problem with horizon, $n+1$.

Since $c^j_{n+1}(\omega^{n+1}) > c^i$, s improves the objective function. Therefore, when the horizon increases, it may no longer be optimal to activate server S_i at that initial state. In other words,

$$m_i(n) \leq m_i(n+1),$$

and the limit as the horizon approaches ∞ exists. Finiteness of the limit is easily shown via a stochastic coupling argument similar to that presented for the $N=2$ case in Walrand (Lemma 3.2b, 1984). *q.e.d.*

Remark: For an infinite horizon, the proof is a straightforward extension of the sample path argument given in Walrand (1984), for the special case of $N=2$ servers. We present below the proof for a finite horizon, and an arbitrary number of servers. Under the stability condition $\lambda < \mu_1 + \cdots \mu_N$, we posed in the introduction, the Markov chain that the optimal policy induces is ergodic. The main result of this paper follows easily as a special case of corollary (1) in Lippman (1973).

Proposition: *There exists a policy that minimizes the average number of jobs in the $M/M/N$ queueing system with heterogeneous servers and it is of threshold type, with thresholds that may depend on (slower) server condition.*

Remark: It is intuitive that the thresholds m_i are nonincreasing functions of the service rates; in other words, we have that $m_1 \leq m_2 \leq \cdots \leq m_N$. This fact is a direct consequence of Lemma 4.

5. Computational experiments

The thresholds can be found using computations on the LP or Dynamic Programming Equations (DPE). In this section, the threshold phenomenon is shown by three test problems. However, there is a memory problem with both the DPE and LP formulations. This is because it is hard to observe the threshold behavior of optimal policies with varied horizons through both techniques, since the number of variables and constraints goes exponentially with horizons. As least, we are able to run a computer program of LP with horizons up to 7. Hence, this restricts us to investigate a single multiserver queueing system, only.

Three different test problems were run, and their results are shown in the tables below. The initial state is specified at the first row in each test problem. With every horizon n, the row of letters, e.g. eghhh, represents the actions to be taken as the number of customers in the queue grows from 1 to 5. The actions are depicted by letters a, b, c, d, e, f, g, and h each representing the state after actions $000, 001, 010, \ldots, 111$, respectively. For example, the state after action 000 means not to feed anyone, 100 means feed server S_1 only, while 011 means feed servers S_2 and S_3.

Table 3.1. Test problem one

Initial state $(x^1, x^2, x^3) = (0, 0, 0)$						
(μ_1, μ_2, μ_3)	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$
$(1.0, 0.1, 0.01)$	eghhh	eeghh	eeegh	eeegg	eeeeg	eeeeg
$(1.0, 0.1, 0.01)$	eghhh	eeghh	eeggh	eeegg	eeegg	eeegg
$(5.0, 2.0, 0.50)$	eghhh	egghh	egggh	egggh	eeggg	eeggg

Table 3.2. Test problem two

Initial state $(x^1, x^2, x^3) = (1, 0, 0)$						
(μ_1, μ_2, μ_3)	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$
$(1.0, 0.1, 0.01)$	cdddd	acddd	aacdd	aaccd	aaacc	aaacc
$(1.0, 0.1, 0.01)$	cdddd	acddd	accdd	aaccd	aaccc	aaccc
$(5.0, 2.0, 0.50)$	cdddd	ccddd	cccdd	cccdd	acccd	acccd

Table 3.3. Test problem three

Initial state $(x^1, x^2, x^3) = (1, 0, 1)$						
(μ_1, μ_2, μ_3)	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$
$(1.0, 0.1, 0.01)$	ccccc	acccc	aaccc	aaccc	aaacc	aaacc
$(1.0, 0.1, 0.01)$	ccccc	acccc	acccc	aaccc	aaccc	aaccc
$(5.0, 2.0, 0.50)$	ccccc	ccccc	ccccc	ccccc	ccccc	acccc

For each row, the threshold structure is clear in each test problem; the threshold numbers are nondecreasing by horizon n. It is worthwhile to notice that the threshold numbers are weakly affected by the slow server conditions. For example, given $\lambda = 0.9$, $\beta = 1.0$, $n = 6$, and $x^0 = 1$, it is optimal to activate server S_2 at $x^3 = 1$, but it is optimal not to activate server S_2 at $x^3 = 0$. These results are consistent with that solved by DPE in Sanyal (1990) where bounds and algorithmic procedures to approximate the optimal thresholds were investigated.

6. Conclusions

We have shown that the optimal customer allocation policy, for a system with N heterogeneous servers, is of threshold type. We have made novel use of linear programming arguments, in order to establish this result. The threshold for activating a server may depend on the condition of slower servers. We were not able to study this dependence using LP arguments. We strongly believe that the dependence is weak (i.e., the threshold is not sensitive to the condition of slower servers.) We suspect that the optimal threshold may vary by at most 1 when the condition of a slower server changes.

We strongly believe that similar, LP based arguments, can be used to establish the form of optimal policies in related optimal queueing control models as well. From the implementation point of view, computation or approximation of the optimal thresholds is an interesting question. Numerical experiments given in section 5 show the threshold structure of an $M/M/3$ with unequal service rates.

Appendix
A proof for totally unimodularity of queueing constraint matrix

We shall show that the constraints $z_j^i(\omega^j), v_j^i(\omega^j) \in [0, 1]$ and the constraints (12), (13), (14) and (18a) form a totally unimodular matrix. As explained in Nemhauser and Wolsey (1988), if the matrix F is totally unimodular, then the matrix [F I] is totally unimodular, where I is an identity matrix. Since the constraints (18a) only form an identity matrix in the whole constraint matrix, we shall consider the constraints (12), (13), and (14).

By the ordering of v, we arrange z by the same manner as

$$z = (z_1, z_2, \ldots, z_n).$$

Note that the cardinality of v_k is $|v_k| = N(N+1)^k$ and that of z_k is $|z_k| = N(N+1)^{k-1}$, for $k \geq 1$. In total, the cardinality of v is $|v| = N(1 + (N+1) + \cdots + (N+1)^n)$ and that of z is $|z| = N(1 + (N+1) + \cdots + (N+1)^{n-1})$. Denote by q the number of variables. Then $q = |v| + |z|$. Denote by p the number of constraints. Then $p = 3[(N+1) + (N+1)^2 + \cdots + (N+1)^n] + 1$. Let matrix Q consist of constraints (12), (13), and (14). The size of Q is therefore $p \times q$. Since each server's contraints are decoupled and the control variables v and z arranged as $(z\ v)$ in each row, we consider the following matrix. Let

$$Q = \begin{pmatrix} & & & & Q_{(12)v^1} & \cdots & \cdots & Q_{(12)v^N} \\ Q_{(13)z^1} & & & & Q_{(13)v^1} & & & \\ Q_{(14)z^1} & & & & Q_{(14)v^1} & & & \\ & Q_{(13)z^2} & & & & Q_{(13)v^2} & & \\ & Q_{(14)z^2} & & & & Q_{(14)v^2} & & \\ & & \ddots & & & & \ddots & \\ & & & Q_{(13)z^N} & & & & Q_{(13)v^N} \\ & & & Q_{(14)z^N} & & & & Q_{(14)v^N} \end{pmatrix}$$

where the submatrix $Q_{(12)v^i}$ consists of the coefficients of v for constraint (12), and $Q_{(13)z^i}, Q_{(13)v^i}, Q_{(14)z^i}, Q_{(14)v^i}$ correspond to the constraints (13), (14) with respect to variables z^i, v^i, for $i = 1, 2, \ldots, N$. Observe that $Q_{(12)v^i}$, $Q_{(13)z^i}$, $Q_{(14)v^i}$ contain zeros and $+1$ elements, but $Q_{(13)v^i}$, and $Q_{(14)z^i}$ contain zeros and -1 elements for $1 \le i \le N$. Denote the l-th element of a column c by $(c)_l$. Then

$$Q_{(j)v^1} = Q_{(j)v^2} = \cdots = Q_{(j)v^N} \quad \text{for } j = (12), (13), (14),$$

$$Q_{(j)z^1} = Q_{(j)z^2} = \cdots = Q_{(j)z^N} \quad \text{for } j = (13), (14).$$

We shall concentrate our attention to the submatrices $Q_{(j)v^1}$ for $j = (12)$, (13), (14) and $Q_{(j)z^1}$ for $j = (13)$ and (14). As a matter of fact, these five different submatrices are the sets of constraints in terms of the combination of the transitions of the Markov chains. They have the following property.

Lemma 7: *Let e_i, e_j be two columns belonging to one of those five submatrices as described above, with $i < j$. If there exists a l such that $(e_i)_l$, $(e_j)_l$ are nonzeros, then $(c_j)_k \neq 0 \Rightarrow (c_i)_k \neq 0$ for every k.*

Proof: Case 1: Suppose $e_i, e_j \in Q_{(12)v^1}$. Let the l-th constraint in $Q_{(12)v^1}$ correspond to a fixed history $\omega_1, \omega_2, \ldots, \omega_i, \ldots, \omega_j$. From eq. (12), we have

$$v^1 + v_1^1(\omega_1) + v_2^1(\omega_1\omega_2) + \cdots + v_i^1(\omega_1\omega_2\ldots\omega_i)$$
$$+ \cdots + v_j^1(\omega_1\omega_2\ldots\omega_j) \le x^0 + \sum_{k=1}^{j} \xi_k^0(\omega^k)$$

where column e_j refers to the coefficients of $v_j^1(\omega_1\ldots\omega_i\ldots\omega_j)$. If another constraint, say the k-th one, contains $\omega_1\ldots\omega_i\ldots\omega_j$ as part of the history ω^n, then it necessarily contains $\omega_1\ldots\omega_i$. Therefore $(e_i)_k = (e_j)_k$. The submatrices $Q_{(13)v^1}$ and $Q_{(14)v^1}$ can be treated in the same way.

Case 2: If $e_i, e_j \in Q_{(13)z^1}$, let the lth constrain in $Q_{(13)z^1}$ correspond to a fixed history $\omega_1, \omega_2, \ldots, \omega_i, \ldots, \omega_j$. From eq. (13), we have

$$-[z_1^1(\omega_1)\xi(\omega_1) + \cdots + z_i^1(\omega_1\omega_2\ldots\omega_i)\xi(\omega_i)$$

$$+ \cdots + z_j^1(\omega_1\omega_2\ldots\omega_j)\xi(\omega_j)] - \sum_{k=1}^{j-1} v_k^1(\omega^k) \le x^1$$

with $\xi(\omega_i) = \xi(\omega_j) = 1$. Therefore the same argument in Case 1 is applied. Then it asserts that any two columns in submatrix $Q_{(.).}$ of Q will be either 'disjoint' or 'contained' one into the other (in the sense that their nonzero elements do so). q.e.d.

Proof of Lemma 1: Since Q is a matrix with entries $1, -1$, and $0, Q$ is totally unimodular if and only if each collection of columns of Q can be split into two parts such that the sum of the columns in one part minus the sum of the columns in the other part is a vector with entries $1, -1$ and 0 only. This proof is provided in Nemhauser and Wolsey (1988). Thus let K be an arbitrary collection of columns of Q. Partition K as $K = K_v \cup K_z$, where K_v and K_z contain columns in which nonzero entries correspond to variables v and z respectively. Let U_1^v, U_2^v, U_1^z and U_2^z be the columns that are initially assigned as zeroes, and will be computed in the following. Arrange the columns in increasing order of horizon in set K_v and K_z individually. Now add the first column in K_v to U_1^v. If the second column in K_v has a common nonzero with the first column, put (add) it into U_2^v; otherwise put (add) it into U_1^v. This arrangement guarantees that K_v has been split into two parts, U_1^v and U_2^v such that $U_1^v - U_2^v$ is a vector with ones and zeros corresponding to contraints (12) and (14) but with -1 and 0 corresponding to contraints (13).

Repeat the procedure arranging this time the set K_z. Thus $K_z = U_1^z \cup U_2^z$, with $U_1^z - U_2^z$ a column vector with -1 and 0 corresponding to constraints (14) but 1 and 0 corresponding to constraints (13). In conclusion, after the process, we have

	$U_1^z - U_2^z$	$U_1^v - U_2^v$
	0	1 or 0
(13)	1 or 0	-1 or 0
(14)	-1 or 0	1 or 0
	:	:
(13)	1 or 0	-1 or 0
(14)	-1 or 0	1 or 0.

Let $U_1 = U_1^z + U_1^v$ and $U_2 = U_2^v + U_2^z$. Therefore $U_1 - U_2 = U_1^v + U_1^z - (U_2^v + U_2^v) = U_1^v - U_2^v + U_1^z - U_2^z$ contains $1, -1$, and 0 only. Thus Q is totally unimodular. q.e.d.

Acknowledgements. This research is supported in part by the National Science Council under grant NSC 84-2121-M-004-006. We are indebted to many people for their comments and support throughout the study.

References

[1] Agrawala A, Coffman EG, Carey MR, Tripathi SK (1984) A stochastic optimization algorithm minimizing expected flow times on uniform processors. IEEE Trans. on Comp. C-33:351–356
[2] Bertsimas D (1995) The achievable region method in the optimal control of queueing systems; formulations, bounds, and policies. Queueing Systems 21:337–389
[3] Coffman EG, Flatto L, Garey MR, Weber RR (1987) Minimizing expected makespans on uniform processor systems. Adv. Applied Prob. 19:177–201
[4] Lin W, Kumar PR (1984) Optimal control of a queueing system with two heterogenous servers. IEEE Trans. on Automatic Control. AC-29:696–703
[5] Lippman SA (1973) Semi-Markov decision processes with unbounded rewards. Mgmt. Sci. 19:717–731
[6] Lippman SA (1975) Applying a new device in the optimization of exponential queueing systems. Opns. Res. 23:687–710
[7] Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization. John Wiley
[8] Rosberg Z, Makowski AM (1990) Optimal routing to parallel heterogeneous servers – small arrival rates. IEEE Trans. on Automatic Control 35–7:789–796
[9] Rosberg Z, Varaiya P, Walrand JC (1982) Optimal control of service in tandem queues. IEEE Trans. on Automatic Control AC-27(3):600–610
[10] Ross SM (1983) Stochastic dynamic programming. Academic Press
[11] Sanyal A (1990) Optimal control of heterogeneous server queueing systems. MS thesis, CSC Department, North Carolina State University
[12] Serfozo RF (1979) An equivalence between continuous and discrete time Markov decision processes. Opns. Res. 27:616–620
[13] Walrand J (1984) A note on optimal control of a queueing system with two heterogenous servers. Systems and Control Letters 4:131–134
[14] Weber RR (1978) On the optimal assignment of customers to parallel servers. Journal Appl. Prob. 15:406–413
[15] Viniotis I (1988) Optimal control of integrated communication systems via Linear programming techniques. Ph. D. dissertation, Univ. of MD, College Park
[16] Xu SH, Righter R, Shanthikumar JG (1992) Optimal dynamic assignment of customers to heterogeneous servers in parallel. Opns. Res. 40:1126–1138