To appear in IEEE Trans. on Auto. Control, July 1991

OPTIMAL ALLOCATION OF CUSTOMERS IN A TWO SERVER QUEUE WITH RESEQUENCING

by

Subir Varma[1]

Electrical Engineering Department and Systems Research Center

University of Maryland, College Park, Maryland 20742

ABSTRACT

This paper deals with the problem of optimal allocation of customers in a two server queue with heterogeneous service rates and resequencing. The resequencing constraint ensures that the customers leave the system in the order in which they entered it. It is shown that the optimal policy that minimizes the average end-to-end delay of customers in the system is independent of the number of customers in the resequencing buffer. Morever it is also shown that the faster server should be kept busy whenever possible.

[1] The work of this author was supported partially through NSF Grants NSFD CDR-85-00108 and ECS-83-51836 and partially through a grant from AT&T Bell Laboratories

1. Introduction

Consider a $M/M/2$ queue with heterogeneous servers which operates under the resequencing constraint that the customers should leave the system in the order in which they entered it. If a customer goes out of sequence after receiving service, it waits in a special buffer—the so-called resequencing buffer—until the customers who entered the queue prior to it have completed service, at which time it leaves the system. The queueing system described above can be used to model the communication between two nodes in a computer network, which are connected together by two independent channels. Multiple channels are useful because if one of the links becomes faulty, then the other links can take on its function, thus paving the way for better fault tolerance. Multiple paths also help in distributing the traffic more evenly in the network. Networks such as IBM's Systems Network Architecture (SNA) and French PTT's public network TRANSPAC, employ multiple paths between two nodes.

An interesting question is that of finding the optimal strategy for assigning customers to the servers in order to minimize their end-to-end delay, which is defined as the sum of their queueing and resequencing delays. When the resequencing constraint is not present this problem has been solved by Lin and Kumar [4]. They showed that it is optimal to keep the faster server busy whenever possible, while the slower server should be assigned a customer only if the number in the buffer exceeds a certain threshold. In this paper, we show that in the presence of the resequencing constraint the decision to assign customers to either server is *not* influenced by the number of customers present in the resequencing buffer. Moreover as in [4] we show the faster server should be kept busy whenever possible.

In this paper we restrict ourselves to FCFS allocation policies for the two servers. Under this restriction we have been unable to obtain the optimal allocation rule for the slower server. However, if the FCFS restriction is removed, then preliminary results (not presented here), suggest that the optimal allocation to the slower server should be of the threshold type.

One of the first two server models to incorporate resequencing is due to Lien [7]. He considered a server allocation policy independent of the number of customers in the resequencing buffer, and of the threshold type in the number in the queue buffer. He derived a closed form expression for the average resequencing delay for various values of the threshold,and showed that the end-to-end delay is minimized for a certain threshold value, which depended on the system parameters. Lien also assumed that when the threshold value is exceeded, the customer to be assigned to the slower server is taken from the first position in the queue. This last assumption was relaxed by Iliadis and Lien [2], who again assumed a threshold type policy, but also considered the case when the customer assigned to the slower server was taken from the last position the buffer, when the threhold value was exceeded. They obtained an expression for the resequencing delay distribution and showed that depending upon the ratio of the service rates, it may be optimal to assign a customer to the slower server from either the first position or the last position in the buffer.

The paper is organized as follows: In Section 2 the model is introduced and a description of the optimization problem is given. In Section 3, the model is discretized and an explicit description of control actions and events is provided. The dynamic programming equation is formulated in Section 4. In Section 5 it is shown that the optimal control of the two servers is independent of the number of customers in the resequencing buffer and in Section 7 the optimal control of the faster server is identified.

2. Model

The system under investigation is a $M/M/2$ queue with heterogeneous servers. The input process is Poisson with rate λ and the service time distribution at server 1 (resp. server 2) is exponential with rate μ_1 (resp. μ_2). We assume $\mu_1 > \mu_2$ so that server 1 and server 2 can be called the fast and slow servers respectively. Following [7], a state space representation of the system is provided by the quintuple $x = (x_0, x_1, x_2, x_3, Z)$ where

$x_0 =$ Number of customers in the buffer of the $M/M/2$ queue where customers await service.

$x_1 =$ 1 (0) if the fast server is busy (idle).

$x_2 =$ 1 (0) if the slow server is busy (idle).

$x_3 =$ Number of customers in the resequencing buffer.

$Z =$ I (O) if the customer being served by server 1 (server 2) arrived earlier into the system than the one being served by server 2 (server 1).

When there is a single customer being served by either one of the two servers, we will adopt the interpretation that $Z = I$ if the customers is with server 1 and $Z = O$ if the customer is with server 2.

In this model, jockeying of customers between the two servers is not permitted, so that once a customer commences service at some server, it remains there for the duration of its service.

The aim is to find the optimal policy which assigns customers to the two servers, so as to minimize their average end-to-end delay. To that end, define the cost incurred per unit time at time t, when the system is in state $x(t)$, by $c(t) = x_0(t) + x_1(t) + x_2(t) + x_3(t)$.

The problem in now cast as a Markov Decision Process [9]. A policy γ is any rule which at $t \geq 0$ decides on the basis of $\{x(s), 0 \leq s \leq t\}$, whether to send a queued message to the idle servers. Let E_x^γ denote the expectation corresponding to the measure P_x^γ on the space of trajectories $\{x(s), 0 \leq s < \infty\}$ such that $x(0) = x$ and the policy γ is used. Since we consider only the discounted cost case, let $\alpha > 0$ be the interest rate used for discounting the future cost, i.e., the present value of cost $c(t)$ incurred at time t is $c(t)e(-\alpha t)$ and the cost incurred by a policy γ over the interval $[0, \infty)$ is given by

$$J(x, \gamma) = E_x^\gamma \int_0^\infty e^{-\alpha t} c(t) dt \tag{2.1}$$

when the system is initially in state x. A policy π is optimal if $J(x, \pi) = \inf_\gamma J(x, \gamma)$ and it is well known that an optimal policy can always be choosen to be Markov and stationary [5], and therefore identifiable with a single mapping from the state space to the space of control actions.

3. The discrete time problem

It is useful to give a discrete time formulation of the problem in order to facilitate the backward induction of Dynamic Programming. The uniformization procedure for doing this is standard and the reader may for instance consult [5] or [11].

A detailed description of the events and control actions in the discrete time problem are now provided. The system state $x = (x_0, x_1, x_2, x_3, Z)$ is an element of the state space

$$
\begin{aligned}
E = \{0\} &\cup I\!N \times \{0,0,0,Z\} \\
&\cup I\!N \times \{1,0\} \times I\!N \times \{I\} \\
&\cup I\!N \times \{0,1\} \times I\!N \times \{O\} \\
&\cup I\!N \times \{1,1\} \times I\!N \times \{I,O\}
\end{aligned}
\tag{3.1}
$$

where $\{0\}$ represents the 'empty' state and $I\!N$ is the set of non-negative integers.

Define the operations $A, D_1, D_2 : E \to E$ by

$$A(x_0, x_1, x_2, x_3, Z) = (x_0 + 1, x_1, x_2, x_3, Z) \tag{3.2}$$

$$D_1(x_0, 1, x_2, x_3, I) = (x_0, 0, x_2, 0, O) \tag{3.3a}$$

$$D_1(x_0, 0, x_2, x_3, O) = (x_0, 0, x_2, x_3, O) \tag{3.3b}$$

$$D_1(x_0, 1, x_2, x_3, O) = (x_0, 0, x_2, x_3 + 1, O) \tag{3.3c}$$

$$D_2(x_0, x_1, 1, x_3, O) = (x_0, x_1, 0, 0, I) \tag{3.4a}$$

$$D_2(x_0, x_1, 0, x_3, I) = (x_0, x_1, 0, x_3, I) \tag{3.4b}$$

$$D_2(x_0, x_1, 1, x_3, I) = (x_0, x_1, 0, x_3 + 1, I) \tag{3.4c}$$

It is plain that A is the arrival operator and that D_i is the departure operator from server $i = 1, 2$. Depending on whether $Z = I$ or O, a departure from server 1 either adds to the number in the resequencing buffer by one, or clears it of all customers. The same behaviour is exhibited by a departure from server 2. The changes in the Z component are illustrated by the following example: When $x_1 = x_2 = 1$ and $Z = I$, then a departure from server 1 changes I to O, since following the departure, server 2 is left serving the customer who started service earlier. Note that a departure of a dummy customer does not change the state of the system.

We now define the control operators for assigning customers to the servers.

- The hold operator P_h is first defined by

$$P_h(x_0,x_1,x_2,x_3,Z) = (x_0,x_1,x_2,x_3,Z).$$

- The operator P_1 defines customer assignment to server 1 and is given by

$$P_1(x_0,x_1,x_2,x_3,O) = \begin{cases} (x_0-1,x_1+1,x_2,x_3,O), & \begin{array}{l}\text{if } x_0 \geq 1, x_1 = 0\\ x_2 = 1, x_3 \geq 0\end{array} \\ (x_0-1,x_1+1,x_2,x_3,I), & \begin{array}{l}\text{if } x_0 \geq 1, x_1 = 0\\ x_2 = 0, x_3 \geq 0\end{array} \end{cases}$$

- The operator P_2 defines customer assignment to server 2 and is given by

$$P_2(x_0,x_1,x_2,x_3,I) = \begin{cases} (x_0-1,x_1,x_2+1,x_3,I), & \begin{array}{l}\text{if } x_0 \geq 1, x_1 = 1\\ x_2 = 0, x_3 \geq 0\end{array} \\ (x_0-1,x_1,x_2+1,x_3,O), & \begin{array}{l}x_0 \geq 1, x_1 = 0;\\ x_2 = 0, x_3 \geq 0\end{array} \end{cases}$$

- Finally, the operator P_b defines customer assignment to both servers at the same time and is given by

$$P_b(x_0,x_1,x_2,x_3,Z) = (x_0-2,x_1+1,x_2+1,x_3,Z)$$

$$\text{if} \quad x_0 \geq 2, x_1 = 0, x_2 = 0, x_3 \geq 0, Z = I \quad or \quad O$$

Let $U = \{u = (u_0,u_1,u_2) : u_i \in \{h,1,2,b\}\}$ be the set of available control actions where control action u_0 is to be taken on the occurrence of an arrival and the control action

$u_i, i = 1,2$ is to be taken on the occurrence of a departure from server i. Let

$$U(x) = \{u \in U : A(x) \in Dom(P_{u_0}), D_i(x) \in Dom(P_{u_i}), i = 1,2\} \tag{3.5}$$

be the set of admissble control actions when the system state is x. Note that $U(x)$ can be expressed as the cartesian product

$$U(x) = U_0(x) \times U_1(x) \times U_2(x) \tag{3.6}$$

where

$$\begin{aligned} U_0(x) =& \{u_0 : A(x) \in Dom(P_{u_0})\} \\ U_i(x) =& \{u_i : D_i(x) \in Dom(P_{u_i})\}, \quad i = 1,2. \end{aligned}$$

In order to completely specify the Markov decision process we define the one-step transition probability function of the underlying discrete-time Markov chain as

$$\begin{aligned} P(x(t+1) = y \mid x(t) = x, u(t) = u) &= \lambda \qquad \text{if } y = P_{u_0}Ax \\ &= \mu_i \qquad \text{if } y = P_{u_i}D_i x \end{aligned} \tag{3.7}$$

The model operates as follows: If the present state is x, then a control action (u_0, u_1, u_2) in $U(x)$ is chosen such that if an arrival occurs the state changes to $P_{u_0}Ax$ while if a departure from server i occurs, then the state changes to $P_{u_i}D_i x$.

4. The Dynamic Programming Formulation

In the discrete-time formulation, the discounted cost criterion becomes

$$E_x^{\gamma}[\sum_{t=0}^{\infty} \beta^t (x_0 + x_1 + x_2 + x_3)] \tag{4.1}$$

where $\beta = \frac{c}{(c+\alpha)}$ is the discount factor. This equation is an easy consequence of discretizing the continous-time cost criterion of Section 2 by the uniformization procedure of Section 3.

Define a stationary policy π as a function $\pi : E \to U$ with $\pi(x)$ in $U(x)$ for every x in E. When a stationary policy π is used, the control $u = \pi(x)$ is applied whenever the system is in state x.

Denote by $\mathbb{F}$ the collection of all functions $f : E \to R$ so that

$$||f|| = \sup_x \frac{f(x)}{\max(x_0 + x_1 + x_2 + x_3, 1)} < \infty \tag{4.2}$$

and observe that $(\mathbb{F}, ||.||)$ is a Banach space. For any stationary policy π, define the operator $T_\pi : \mathbb{F} \to \mathbb{F}$ by

$$T_\pi f(x) = x_0 + x_1 + x_2 + x_3 + \beta\lambda f(P_{u_0} Ax) + \beta\mu_1 f(P_{u_1} D_1 x) + \beta\mu_2 f(P_{u_2} D_2 x) \tag{4.3}$$

for all x in E, where $\pi(x) = (u_0, u_1, u_2)$ and consider the dynamic programming operator $T : \mathbb{F} \to \mathbb{F}$ given by

$$(Tf)(x) = \min_\pi (T_\pi f)(x) \tag{4.4}$$

The operator T, acting on the space of functions $\mathbb{F}$, is a contraction mapping, so that if J^β is the optimal cost function, then $J^\beta = TJ^\beta$, and for any $f$in $\mathbb{F}$, $\lim_{n\to\infty} T^n f = J^\beta$. The dynamic programming equation of interest for the minimization problem associated with (4.1) can be written as

$$J^\beta(x) = \min_{u \in U(x)} \left[x_0 + x_1 + x_2 + x_3 + \beta\lambda J^\beta(P_{u_0} Ax) \right.$$
$$+ \beta\mu_1 J^\beta(P_{u_1} D_1 x) \tag{4.5}$$
$$\left. + \beta\mu_2 J^\beta(P_{u_2} D_2 x) \right].$$

In view of (3.6), this can be re-written as

$$J^\beta(x) = x_0 + x_1 + x_2 + x_3 + \min_{u_0 \in U_0(x)} \beta\lambda J^\beta(P_{u_0} Ax)$$
$$+ \min_{u_1 \in U_1(x)} \beta\mu_1 J^\beta(P_{u_1} D_1 x) \tag{4.6}$$
$$+ \min_{u_2 \in U_2(x)} \beta\mu_2 J^\beta(P_{u_2} D_2 x).$$

The following notation will be used in the rest of the paper: For any f in $I\!F$, define

$$\begin{aligned} \Delta_a^Z f(i,j,k) &= f(i,1,0,j,Z) - f(i,1,0,k,Z) \\ \Delta_b^Z f(i,j,k) &= f(i,1,1,j,Z) - f(i,1,1,k,Z) \qquad i,j,k \geq 0 \\ \Delta_c^Z f(i,j,k) &= f(i,0,1,j,Z) - f(i,0,1,k,Z) \end{aligned} \tag{4.7}$$

Also given $\underline{x} = (x_0, x_1, x_2, x_3, Z)$ and $\underline{y} = (y_0, y_1, y_2, y_3, Z)$, we say that $\underline{x} \geq \underline{y}$ if $x_i \geq y_i, i = 0,1,2,3$.

5. A key property

In the present section we show that the optimal control of server 2 is independent of the number of customers in the resequencing buffer. A similar proof holds for server 1 and is omitted. The proof proceeds by the technique of value iteration, wherein it is assumed that a function f satisfies certain inequalities. These inequalities are satisfied identically by the zero function and further $\lim_{n\to\infty} T^n 0 = J^\beta$. Hence if we show that Tf also satisfies these inequalities, it would follow that the same is true for $T^n f$. Since $\lim_{n\to\infty} T^n f = J^\beta$, it follows that the optimal value function also satisfies these inequalities. The main result of this section is now stated.

Lemma 5.1. *The following relations propagate under the dynamic programming operator, i.e., if f in $\mathbb{F}$ satisfies the relations*

$$\Delta_a^I f(x_0, x_3, x_3') = \Delta_b^I f(y_0, y_3, y_3') \tag{5.1a}$$

$$\Delta_b^I f(x_0, x_3, x_3') = \Delta_b^I f(y_0, y_3, y_3') \tag{5.1b}$$

$$\Delta_a^I f(x_0, x_3, x_3') = \Delta_a^I f(y_0, y_3, y_3') \tag{5.1c}$$

for $x_0, x_3, y_0, y_3, x_3', y_3' \geq 0, x_3 - x_3' = y_3 - y_3'$, so does Tf.

Proof. We provide a proof for (5.1a) and leave the proofs for (5.1b-c) to the reader. We have to show that,

$$Tf(x_0, 1, 0, x_3, I) - Tf(x_0, 1, 0, x_3', I) = Tf(y_0, 1, 1, y_3, I) - Tf(y_0, 1, 1, y_3', I). \tag{5.1a'}$$

In view of (4.6)-(4.7), $(5.1a')$ holds if Eqns. (5.2)(5.3) and (5.4) given below are satisfied, where

$$\min_{h,2} f(P_{u_0}(x_0+1, 1, 0, x_3, I)) - \min_{h,2} f(P_{u_0}(x_0+1, 1, 0, x_3', I))$$

$$= f(P_{u_0}(y_0+1, 1, 1, y_3, I)) - f(P_{u_0}(y_0+1, 1, 1, y_3', I)) \tag{5.2}$$

$$\min_{1,b} f(P_{u_1}(x_0,0,0,0,O)) - \min_{1,b} f(x_0,0,0,0,O) = \min_{1,b} f(P_{u_1}(y_0,0,1,0,O)) - \min_{1,b} f(y_0,0,1,0,O)) \tag{5.3}$$

and

$$\min_{h,2} f(P_{u_2}(x_0,1,0,x_3,I)) - \min_{h,2} f(P_{u_2}(x_0,1,0,x_3',I))$$

$$= \min_{h,2} f(P_{u_2}(y_0,1,0,y_3+1,I)) - \min_{h,2} f(P_{u_2}(y_0,1,0,y_3'+1,I)) \tag{5.4}$$

If we apply the hold operator on the LHS of (5.2), then it is true because of (5.1a). If we apply the operator P_2, then it is true because of (5.1b). Eqn. (5.3) is obviously true by inspection. Expanding (5.4), we obtain

$$\min[f(x_0,1,0,x_3,I), f(x_0-1,1,1,x_3,I)]$$

$$-\min[f(x_0,1,0,x_3',I), f(x_0-1,1,1,x_3',I)]$$

$$= \min[f(y_0,1,0,y_3+1,I), f(y_0-1,1,1,y_3+1,I)]$$

$$-\min[f(y_0,1,0,y_3'+1,I), f(y_0-1,1,1,y_3'+1,I)] \tag{5.4'}$$

and (5.4$'$) will clearly be true if the following equations hold:

$$f(x_0,1,0,x_3,I) - f(x_0,1,0,x_3',I) = f(y_0,1,0,y_3+1,I) - f(y_0-1,1,0,y_3'+1,I) \tag{5.5}$$

$$f(x_0-1,1,1,x_3,I) - f(x_0-1,1,1,x_3',I) = f(y_0-1,1,1,y_3+1,I) - f(y_0-1,1,1,y_3'+1,I) \tag{5.6}$$

$$f(x_0,1,0,x_3,I) - f(x_0,1,0,x_3',I) = f(y_0-1,1,1,y_3+1,I) - f(y_0-1,1,1,y_3'+1,I) \tag{5.7}$$

and

$$f(x_0-1,1,1,x_3,I)-f(x_0-1,1,1,x_3',I)=f(y_0,1,0,y_3+1,I)-f(y_0,1,0,y_3'+1,I) \quad (5.8)$$

But (5.5)-(5.8) are just Eqns. $(5.1a-c)$ which are assumed to be true. Hence equation (5.1a) is proved.

∎

Eqn. (5.1a) indicates that the decision to allocate a customer to server 2 is independent of the number of customers in the resequencing buffer. This can be shown as follows. Suppose there are x_0 customers in the queue buffer and x_3 customers in the resequencing buffer, and for these values of x_0 and x_3 it is optimal to assign a customer to server 2, in which case

$$f(x_0,1,0,x_3,I)-f(x_0-1,1,1,x_3,I)\geq 0$$

But by (5.1a) this implies that

$$f(x_0,1,0,x_3+1,I)-f(x_0-1,1,1,x_3+1,I)\geq 0.$$

so that it is also optimal to assign a customer to the slower server when there are x_3+1 customers in the resequencing buffer. Eqns. (5.1b-c) were found to be necessary to make (5.1a) propagate. These conclusions are summarized in the following theorem.

Theorem 5.1. *The optimal control of the two servers is independent of the number of customers in the resequencing buffer.*

6. The effect of the Z variable on the value function

In this section we show that if both servers are busy, then an out-of-sequence state has a greater value function than the corresponding in-sequence state. This property will come in use later when we prove that the faster server should be kept busy whenever possible.

Lemma 6.1. *The inequalities (6.1a-e) below propagate under the dynamic programming operator, where*

$$f(x_0, 1, 1, 0, O) \geq f(x_0, 1, 1, 0, I) \tag{6.1a}$$

$$\triangle_b^O f(x_0, x_3, x_3') \geq \triangle_b^I f(y_0, y_3, y_3') \tag{6.1b}$$

$$\triangle_b^O f(x_0, x_3, x_3') \geq \triangle_a^I f(y_0, y_3, y_3') \tag{6.1c}$$

$$\triangle_c^O f(x_0, x_3, x_3') \geq \triangle_a^I f(y_0, y_3, y_3') \tag{6.1d}$$

$$\triangle_c^O f(x_0, x_3, x_3') \geq \triangle_b^I f(y_0, y_3, y_3') \tag{6.1e}$$

for all $x_0, x_3, x_3', y_0, y_3, y_3', x_3 - x_3' \geq y_3 - y_3'$.

Proof. We only provide a proof for (6.1a). The proof for the other cases may be found in [13]. We have to prove that

$$Tf(x_0, 1, 1, 0, O) \geq Tf(x_0, 1, 1, 0, I). \tag{6.1a'}$$

We will consider the case $x_0 \geq 1$. Eqn. (6.1a') will be true provided Eqns. (6.2) and (6.3) below are satisfied, where

$$f(P_{u_0}(x_0 + 1, 1, 1, 0, O)) \geq f(P_{u_0}(x_0 + 1, 1, 1, 0, O)) \tag{6.2}$$

and

$$\beta\mu_1 f(x_0 - 1, 1, 1, 1, O) + \beta\mu_2 \min_{h,2} f(P_{u_2}(x_0, 1, 1, 1, I))$$

$$\geq \beta\mu_1 f(x_0 - 1, 1, 1, 0, O) + \beta\mu_2 \min_{h,2} f(P_{u_2}(x_0, 1, 0, 1, I)). \tag{6.3}$$

Eqn. (6.2) follows from (6.1a). Eqn. (6.3) reduces to proving the folowing two equations (6.3') and (6.3"), where

$$\mu_1[f(x_0 - 1, 1, 1, 1, O) - f(x_0 - 1, 1, 1, 0, O)] \geq \mu_2[f(x_0 - 1, 1, 1, 1, I) - f(x_0 - 1, 1, 1, 0, I)] \tag{6.3'}$$

and

$$\mu_1[f(x_0 - 1, 1, 1, 1, O) - f(x_0 - 1, 1, 1, 0, O)] \geq \mu_2[f(x_0, 1, 0, 1, I) - f(x_0, 1, 0, 0, I)] \tag{6.3''}$$

But Eqns. $(6.3')$ and $(6.3'')$ follow from $(6.1a)$ and $(6.1b)$, and $(6.1a)$ is verified.

∎

Eqn. (6.1a) contains the main result of this section. Eqns. (6.2b-e) were found to be necessary to make (6.1a) propagate. An intuitive explanation of why (6.1a) holds is that when $Z = O$, there is a greater potential for a large number of customers to accumulate in the resequencing buffer, compared to the case when $Z = I$.

7. Identification of the Optimal Control of the Faster Server

In the present section, we identify the inequalities that the value function must satisfy in order to specify the optimal control of the faster server.

Lemma 7.1. *The inequalities*

$$f(P_h x) \geq f(P_1 x), \quad x \in Dom(P_1) \tag{7.1}$$

$$f(P_2 x) \geq f(P_1 x), \quad x \in Dom(P_1) \cap Dom(P_2) \tag{7.2}$$

are propagated under the dynamic programming operator.

∎

Inequality (7.1) implies that if server 1 is idle, then it is always optimal to assign a customer to it, if one is available in the buffer, irrespectively of whether server 2 is busy or idle. Inequality (7.2) implies that if both servers are idle, then it is optimal to assign a customer to server 1 rather than to server 2. Together these inequalities say that it is always optimal to keep server 1 busy whenever possible. These conclusions are summarized in the following theorem.

Theorem 7.1. *Whenever server 1 is idle, it is optimal to assign to it a customer if one is waiting for service.*

∎

We now provide a proof for Lemma 7.1.

Proof.

In order to show that the optimal value function satisfies (7.1)-(7.2), it is sufficient to prove that these inequalities are propagated under the dynamic programming operator, i.e. if f satisfies (7.1)-(7.2), so does Tf. In the process of doing so, we found that it was necessary for the value function to satisfy additional properties $(7.3a)-(7.3c)$ below, namely

$$f(\underline{x}) \geq f(\underline{y}) \quad \text{if} \quad \underline{x} \geq \underline{y} \tag{7.3a}$$

$$f(x_0,1,1,x_3,O) \geq f(x_0,0,1,x_3+1,O) \qquad x_0,x_3 \geq 0 \tag{7.3b}$$

$$f(x_0,1,1,x_3,O) \geq f(x_0-1,1,1,x_3+1,O) \quad x_0 \geq 1, x_3 \geq 0 \tag{7.3c}$$

We first show that

$$Tf(x_0,0,1,x_3,O) \geq Tf(x_0-1,1,1,x_3,O), \qquad x_0 \geq 1. \tag{7.1'}$$

We will consider the case $x_0 \geq 2$. The following inequalities (7.4), (7.5) and (7.6) have to be verified, where

$$\min_{h,1} f(P_{u_0}(x_0+1,0,1,x_3,O)) \geq \min_{h} f(P_{u_0}(x_0,1,1,x_3,O)) \tag{7.4}$$

$$\min_{h,1} f(P_{u_1}(x_0,0,1,x_3,O) \geq \min_{h,1} f(P_{u_1}(x_0-1,0,1,x_3+1,O) \tag{7.5}$$

and

$$\min_{1,b,h} f(P_{u_2}(x_0,0,0,0,I)) \geq \min_{2,h} f(P_{u_2}(x_0-1,1,0,0,I)). \tag{7.6}$$

By Eqn. (7.1), Eqn. (7.4) reduces to the comparison

$$f(x_0,1,1,x_3,O) \geq f(x_0,1,1,x_3,O)$$

which holds with equality. Similarly again by (7.1), Eqn. (7.5) reduces to

$$f(x_0-1,1,1,x_3,O) \geq f(x_0-2,1,1,x_3+1,O)$$

which is true by (7.3c). As for (7.6), if we take action P_1 on the left-hand-side and P_h on the right-hand-side, (7.6) reduces to an equality. The other cases can be treated in a similar fashion.

Since the proofs for (7.2)-(7.3) proceed by a similar technique, they are omitted. The interested reader may refer to [13] for further details. ∎

An intuitive interpretation is now provided for some these properties. Inequality (7.3a) expresses the monotonicity property of the optimal value function. The value function increases if the magnitude of any one the first four states is increased while the other states remain unchanged, as should be intuitively clear by the linearity of the cost in the first four states.

Eqn. (7.3b) reveals that the value function decreases if the total number of customers in the $M/M/2$ queue and the resequencing buffer is kept constant, while the number in the queue buffer is decreased and that in the resequencing buffer is increased. Hence it is 'better' to have a customer in the resequencing buffer rather than in the queue buffer. This is also intuitively appealing since a customer in the resequencing buffer has already finished its service, while the one in the queue buffer is yet to receive service. Eqn. (7.3c) has a similar interpretation.

9. Conclusions

The result in this paper may seem to be counter-intuitive at first, because the reader may expect that the decision to assign a customer to the servers ought to depend on the number in the resequencing buffer. It is now argued why this is not the case. Consider the situation when server 2 is idle, server 1 is busy and there are x_0 and x_3 customers in the queue and resequencing buffers, respectively. The controller has to decide whether to assign a customer to server 2 or not. Note that all the x_3 customers in the resequencing buffer are being 'held up' by the customer in service at server 1. Hence the assignment of a customer to server 2 will in no way influence the epoch of their departure from the system. Hence the controller is not influenced by x_3 while making its decision.

The same kind of reasoning also explains why server 1 should be kept busy whenever possible. Indeed suppose that the customer in server 2 is holding up x_3 customers in the resequencing buffer and server 1 is idle. If the controller assigns a waiting customer to server 1, and it finishes service before the customer in server 2, then the number in the resequencing buffer will increase further. However, when the customer at server 2 does end service, a larger number of customers will leave the resequencing box than in the case when the controller does not assign a customer to server 1. This is the intuitive explanation behind equations (7.5a-b) which stated that the value function decreases as the number in the resequencing buffer increases and that in the main buffer decreases.

As stated in the introduction, we have been unable to identify the optimal control of the slower server under the restriction of FCFS policies. However we conjecture that the optimal allocation rule is of the threshold type in the number of customers in the main queue buffer. We would expect the value of the threshold at which the controller starts assigning customers to the slower server to increase, as compared to the situation without the resequencing constraint. The reason for this is as follows. Without the resequencing constraint, use of server 2 may be harmful because the loss due to the additional time that a customer spends at this server may outweigh the gain due to the decrease in waiting time of the customers which were behind him at the time when he was assigned to server 2. With the resequencing constraint, the use of server 2 is even more harmful because not only does the customer spend more time getting served at 2, but also a few customers behind him may be unable to leave the system after service at server 1, due to the resequencing constraint. This effect is illustrated in Table 1. below. Note that $T1$ is the threshold for the system without resequencing, while $T2$ is the threshold for the system with resequencing. The thresholds have been calculated with the help of the explicit formulae for the average number in the system which were derived by Iliadis and Lin [2].

Table 1

λ	μ_1	μ_2	$T1$	$T2$
13	15	5	0	2
13	15	4	1	3
13	15	3	1	5
13	15	2	2	10
13	15	1	4	33

When there are more than two servers, the optimal server allocation is still unsolved, but it may still be possible to prove that the optimal policy is independent of the number in the resequencing buffer. Investigation of this problem is underway, and some preliminary results are available.

Acknowledgment

The author is grateful to Dr. Armand Makowski for continued interest in this work and also for numerous suggestions which helped in improving the contents of this paper.

REFERENCES

[1] B. Hajek, "Optimal control of two interacting service stations," *IEEE Trans. on Auto. Control*, Vol. 29, pp. 491-499 (1984).

[2] I. Iliadis and Y.C. Luke Lien,"Resequencing delay for a queueing system with two heterogeneous servers under a threshold type scheduling," *IEEE Trans. on Comm*, Vol. 36, No. 6, pp. 692-702 (1988).

[3] P.R. Kumar and P. Varaiya, *Stochastic Systems*, Prentice-Hall Inc. (1987).

[4] W. Lin and P.R. Kumar, "Optimal control of a queueing system with two heterogenous servers," *IEEE Trans. on Auto. Control*, Vol. 28, pp. 696-703 (1984).

[5] S.A. Lippman, "Applying a new device in the optimization of exponential queueing systems," *Oper. Res.*, Vol. 23, pp. 687-710 (1975).

[6] S.A. Lippman, "Semi-Markov decision processes with unbounded rewards," *Manag. Sci.*, Vol. 19, pp. 717-731 (1973).

[7] Y.C. Luke Lien, "Evaluation of the resequencing delay in a Poisson queueing system with two heterogeneous servers," IBM Research Centre Report, Yorktown Heights, (1985).

[8] B.S. Maglaris, "An optimal local policy for two-level adaptive routing in computer networks," *IEEE INFOCOM'84*, pp. 284-290 (1984).

[9] S.M. Ross *Stochastic Dynamic Programming*, Academic Press, (1983).

[10] Z. Rosberg, P. Varaiya and J.C. Walrand, "Optimal control of service in tandem queues,"*IEEE Trans. on Auto. Control*, Vol. 27, No. 3, pp. 600-610 (1982).

[11] R. Serfozo, "An equivalence between discrete and continous time Markov decision processes," *Op. Res.*, Vol. 27, pp. 616-620 (1979).

[12] S. Stidham Jr., "Optimal control of admission to queueing system," *IEEE Trans. on Auto. Control*, Vol. 30, No. 8, pp. 705-713 (1985).

[13] S. Varma, "Some problems in queueing systems with resequencing," MS Thesis, University of Maryland, College Park (1987), Also available as Technical Report No. TR-87-192, Systems Research Center, University of Maryland.

[14] J. Walrand, "A note on optimal control of a queueing system with two heterogenous servers," *Sys. and Contr. Letters*, Vol. 4, pp. 131-134 (1984).

[15] Z.J. Wu, P.B. Luh, S.C. Chang and D.A. Castanon,"Optimal control of a queueing system with two interacting service stations and three classes of impatient tasks," *IEEE Trans. on Auto. Control*, Vol. 33, No. 1, pp. 42-49 (1988).