OPTIMALCONTROL OF A TWO-STAGE TANDEM QUEUING SYSTEM WITH FLEXIBLE SERVERS

SOL-AKUBUDE, VINCENT I.P.
Department of Mathematics Chukwuemeka Odumegwu Ojukwu University, Anambra State, Nigeria

ABSTRACT
We consider the optimal control of two parallel servers in a two-stage tandem queuing system with two flexible servers. New jobs arrive at station 1, after which a series of two operations must be performed before they leave the system. Holding costs are incurred at rate $h_1$per unit time for each job at station 1 and at rate $h_2$per unit time for each job at station 2.The system is considered under two scenarios; the collaborative case and the noncollaborative case. In the prior, the servers can collaborate to work on the same job, whereas in the latter, each server can work on a unique job although they can work on separate jobs at the same station. We provide simple conditions under which it is optimal to allocate both servers to station 1 or 2 in the collaborative case. In the noncollaborative case, we show that the same condition as in the collaborative case guarantees the existence of an optimal policy that is exhaustive at station 1. However, the condition for exhaustive service at station 2 to be optimal does not carry over. This case is examined via a numerical study.

Keywords: Optimal control, Tandem queuing system, flexible server's average cost, collaborative case, non-collaborative case

1. INTRODUCTION

The use of a cross-trained workforce has become prevalent in the manufacturing and service industries. Despite this fact, very little work has been done to examine how to optimally allocate such flexible workers or resources to tasks on hand. This is especially true for systems with more than one flexible server. Because the set of possible policies in these models can be intractable even in small examples, findingconditions under which the structure of optimal policies is simple can be crucial.

Consider a two-stage tandem queuing system with jobs arriving only at station 1 according to a Poisson process of rate $\lambda > 0$. Each arriving job must receive service at each station before leaving the system. However, the service may be performed by either of two identical servers. When either server 1 or 2 processes jobs at station 1, the service time is exponential with rateμ_1. Similarly, when either server processes jobs at station 2, the service time is exponential with rateμ_2. Holding costs are incurred at rate h_1, per unit time for each job in queue 1 and at rate $h_2$per unit time for each job in queue 2. We further assume that preemption is permitted and that no setup is required for the server to switch from processing jobs at one station to another. The problem then is how to allocate the servers over time to jobs at stations 1 and 2. In this article, we analyze two versions of this problem. In the first version, called the collaborative case, the servers may collaborate to work on the same job at the same time, thereby doubling the service rate at that station, whereas in the second version, the noncollaborative case, there must be a job available for each server at a station to avoid idling a server. We present conditions in the collaborative case for the optimal policy to be exhaustive at station 1; that is, both servers process jobs at station 1 as long as there is a job to be served. By simply reversing an inequality, we also provide the condition for optimality of exhaustive service at station 2. Thus, in this case, we obtain a complete characterization of the optimal policy. The non-collaborativecase is more subtle. We show that the same inequality provided in the collaborative case suffices to guarantee the existence of an optimal policy that is exhaustive in station 1. However, the reverse inequality no longer guarantees optimality of an exhaustive policy at station 2. Furthermore, because we discuss these issues in the context of long-run average optimality, we are also able to show how the stability region of the network can be relaxed with a flexible workforce in place.

Among the few works that have considered the optimal control of flexible servers, Duenyas, Gupta, and Olsen (1998) and Iravani and colleagues (1996, 1997) in parallel, consider a tandem queuing system with only one flexible server and characterize the optimal policy to be a monotone switching curve. The control of a flexible work-' force in minimizing the cycle time of each job is discussed in Van Oyen, Senturk-Gel, and Hopp (1996). Andradottir, Ayhan, and Down (2001) consider the control of flexible servers to maximize throughput when the service rate is additive. Javidi, Song, and Teneketzis(2001) consider two interconnected queues withidentical machines and derive sufficient conditions under which the policy that prioritizes jobs in queue 1 minimizes the expected value of the first time that the system becomes empty (i.e. the dynamic version of make span). None of these articles consider the average cost case with multiple flexible servers in a dynamic environment.

In parallel with the work described above, there has been another stream of research that analyzes the control of a flexible workforce in clearing system problems. In a clearing system, no external arrivals are permitted after time zero and the objective is to minimize the expected holding costs incurred until all jobs leave the system. Farrar (1993) focused on two-stage tandem queues with a fixed server at each queue and a single flexible server. He showed that the optimal allocation of the flexible server is transition monotone and can be characterized by a switching curve. Paridelis and Teneketzis (1994) characterize the sufficient condition for the optimality of an exhaustive policy in the upstream queue in a two-stage interconnected queue. Essentially, the problem considered in Ahn, Duenyas, and Zhang (1991) is a clearing system version of our problem. They presented necessary and sufficient conditions under which an exhaustive policy at the upstream (or downstream) queue minimizes 'the expected value of weighted holding costs. Although the model is similar, we focus on the case when arrivals are introduced. The reader is referred to this article and the references therein for a more detailed literature review.

The rest of this article is organized as follows. Section 2 provides a Markov decision process formulation of the problem under the average cost optimality criterion. We also present results on stability for both cases and show the existence of a solution to the average cost optimality equations. Section 3 contains our main results: We characterize the structure of the optimal policy under the collaborative system and give sufficient conditions under which an exhaustive policy at the upstream queue is optimal in the noncollaborative system. In Section 4, we present several examples that display the benefits of a flexible workforce.

2. PRELIMINARIES

Throughout the rest of the article (unless otherwise stated), we assume that *uniformization*has been applied in the spirit of Lippman (1975) with uniformizationconstant $\Psi \equiv \lambda + 2\max\{\mu_1, \mu_2\}$. Without loss of generality we assume $\Psi = 1$.Thus, instead of considering the continuous-time problem defined earlier, we consider a discrete-time equivalent. Furthermore, the arrival and service rates can be interpreted as probabilities. A non-anticipating policy πdescribes where each server should serve next, given the current time and the number of customers in each queue. Thus, a policy $\pi$is a sequence of decisions $\pi = \{d_1, d_2, \dots\}$,where $d_k$is a vector of actions to be taken depending on the current state. Let II be the set of all such policies. The objective is to find a policy that minimizes the average cost per unit time over an infinite horizon. We next formally define the long-run average cost.

Let $X_n$be the state of the system at time n and $d_n$be the decision rule at time n under a particular policy π. The n-stage expected cost is

$$V_n(\pi, x) \equiv \mathbb{E}_x^{\pi}\left[\sum_{k=0}^{n-1} c\ \ (X_k, d_k(X_k))\right].$$

The optimal n-stage expected cost is

$$V_n^*(x) = \inf_{\pi \in \Pi} V_n(\pi, x). \qquad (2.1)$$

DEFINITION 2.1: The long-run average reward or gain of a policyπ, given that the initial state of the system started is x, denoted $g(\pi, x)$, is given by

$$g(\pi, x) = \lim_{n\to\infty} \frac{V_n(\pi, x)}{m}. \qquad (2.2)$$

Furthermore, let the optimal expected average cost be denoted by $g^*(x)$:

$$g^*(x) = \inf_{\pi \in \Pi} g(\pi, x).$$

A policy $\pi^*$is called long-run average or gain optimal if$g^*(x) = g^*(x)$for all$x \in \mathbb{X}$, where $\mathbb{X}$denotes the state space.

We formulate the control problem as a Markov decision process. The set of decision epochs corresponds to the set of all arrivals, service completions, and dummy transitions due to uniformization. The preemptive discipline restricts the optimal policy to be non-idling because any unforced idling is suboptimal.

The state space is denoted by $\mathbb{X} = \{(i, j)|i, j \in \mathbb{N}\}$where $\mathbb{N}$ is the set of non-negative integers; the first element is the number of jobs at station 1 and the second is the number of jobs at station 2. The average cost optimality equations (ACOE) for the collaborative system (cf. Sennott (1999) are for$i, j \geq 1$,

$$\begin{aligned} g + v(i,j) = \min_{s=\{0,1,2\}} [h_1 + h_2 j + \lambda_v(i+1, j) + s\mu_1 v(i-1, j+1) \\ +(2-s)\mu_2 v(i, j-1) \\ +(1 - \lambda - s\mu_1 - (2-s)\mu_2)v(i,j)] \qquad (2.3) \end{aligned}$$

where action s corresponds to how many servers are processing jobs at station 1. Because idling is suboptimal, the ACOE for the remaining states is

$$g + v(i,j) = \begin{cases} h_2 j + \lambda v(1,j) + 2\mu_2 v(0, j-1) + 1 - \lambda - 2\mu_2) v(0,j) & \text{for } i = 0 \text{ and } j \geq 1, \\ h_1 + \lambda v(i+1, 0) + 2\mu_1 v(i-1, 1) + 1 - \lambda - 2\mu_1) v(i, 0) & \text{for } i \ \geq 1 \text{ and } j = 0, \\ \lambda v(1,0) + (1-\lambda) v(0,0) & \text{for } i = 0 \text{ and } j = 0 \end{cases}$$

In the noncollaborative case, the ACOE must consider the number of servers that can be assigned to each queue. For example, in state (1,1) the servers must be assigned to different stations. Thus, let $\mathcal{A} = \{s\ \in \{0,1,2\}|\max\{2-j, 0 + \leq s \leq \min\{i, 2\}\}$represent the allowable actions in each state. The ACOE for the noncollaborative case are given by

$$\begin{aligned} g + v(i,j) = \min_{s \in \mathcal{A}} [h_1 + h_2 j + \lambda v(i+1, j) + s\mu_1 v(i-1, j+1) + \ (2-s)\mu_2 v(i, j-1) \\ +\big(1 - \lambda - s\mu_1 - (2-s)\mu_2\big) v(i,j)] \qquad (2.5) \end{aligned}$$

and

$$g + v(i,j) = \begin{cases} h_1 j + \lambda v(1,0) + \big(1_{\{i\geq 1\}} + 1_{\{i\geq 2\}}\big)\mu_1 v(i-1, 1) \\ +\big(1 - \lambda - (1_{\{i\geq 1\}}\big) + 1_{\{i\geq 2\}})\mu_1) v(i, 0) \text{for } i \geq 0 \text{ and } j = 0 \\ h_2 j + \lambda v(1, j) + \big(1_{\{i\geq 1\}} + 1_{\{i\geq 2\}}\big)\mu_1 v(0, j-1) \\ +\big(1 - \lambda - \big(1_{\{i\geq 1\}} + 1_{\{i\geq 2\}}\big)\mu_2\big) v(0, j) \text{for } i = 0 \text{ and } j \geq 0, \end{cases}$$

(2.6)

where$1_A$is the indicator function of the set *A*.

When a solution (g, v) to the ACOE exists, $g = g^*$and v is called a relative valuefunction.Furthermore, given that $(g^*, v)$is a solution to the ACOE, a decision rule d^*that attains the minimum (componentwise) of the ACOE generates a stationary, deterministic policy $\pi^* = \{d^*, d^*, \ldots\}$such that $\pi^*$is gain optimal.

We next derive a condition under which there exists a solution to the ACOE. We begin by giving a stability condition for non-idling policies. Note that this condition is similar to that derived independently byAndradottir, Ayhan, and Down (2001) using alternative methods, but our focus is on showing the existence of a solution to the ACOE rather than deriving general stability conditions for this network.

First, note that under any non-idling policy there is but one closed, communicating class. For example, consider the policy that allocates both servers to station 2 -whenever there is work to be done at station 2. Assume that the initial state is (0, 0) (i.e. no jobs in the system). Then, in the noncollaborative case, there can never be more than two jobs at station 2, whereas in the collaborative case, there can never be more than one. States with more than two customers at station 2 are not part of the communicating class and are necessarily transient.

PROPOSITION 2.2: Under any stationary, non-idling policy, the system is stable if and only if$\lambda(1/\mu_1 + 1/\mu_2) < 2$;that is, there exists a stationary distribution.

PROOF: To prove sufficiency, we fix an arbitrary, stationary, non-idling policy IT, find a Lyapunovfunction, and apply Foster's criterion (cf.(1995) Thm. 3.7). This guarantees that all recurrent states are positive recurrent. To this end, consider the Markov chain $\{X_n, n \geq 0\}$ generated by πrestricted only to those states that belong to its communicating class. Denote this chain by $\{Z_n, n \geq 0\}$audits state space by S, and note that S is irreducible. Let $G = \{(i, j) | i \leq 1, j \leq 1\}$. Let $y(i, j)$be the workload process in state(i, j); that is,

$$y(i, j) \equiv \frac{i}{\mu_1} + \frac{i + j}{\mu_2} \qquad (2.7)$$

Now, note that for any action chosen and $(i, j) \notin G$.

$$\mathbb{E}\big(y(Z_{n+1}) - y\,(Z_n) | Z_n = (i, j)\big) = \lambda\left(\frac{1}{\mu_1} + \frac{1}{\mu_2}\right) - \mu_1\left(\frac{1}{\mu_1}\right) - \mu_2\left(\frac{1}{\mu_2}\right)$$

$$= \lambda\left(\frac{1}{\mu_1} + \frac{1}{\mu_2}\right) - 2. \qquad (2.8)$$

Because *G* is finite, we may now apply Theorem 3.7 of [10] to $\{Z_n\}$ to get that all states in *S* are positive recurrent when the right-hand side of (2.8) is strictly negative. Furthermore, because (2.8) holds on all states of the original chain, applying Proposition C. 1 .5 of [16] yields that the expected time to reach the communicating class is finite.

Thus, the result holds for $\{X_n, n > 0\}$, and sufficiency is proven.

To show the necessity of the inequality, we note that Theorem 11.5.1 of [14]implies that when $\lambda\left(\left(\frac{1}{\mu_1}\right) + \left(\frac{1}{\mu_2}\right)\right) - 2 \geq 0$, the expected time to reach the set $G = \{(i, j)/i, j \leq 1\}$ from outside of this set is infinite; thus, a stationary distributioncannot exist.

The next result states that when the system is stable, the ACOE have a solution. We find useful the fact that, for a fixed policy 77, if there exists a function $V: \mathbb{X} \to \mathbb{R}_+$ and a constant *J* such that

$$P_\pi V(x) \equiv \mathbb{E}_\pi\{V(X_{n+1}) | X_n = x] \leq V(x) - c\big(x, d(x)\big) + J, x \in \mathbb{X}, \quad (2.9)$$

where$\{X_n, n \geq 0\}$ is the Markov chain generated by *π*, then the average cost of πis finite (cf. Meyn(2001)Thm. 3]).

PROPOSITION 2.3: If $\lambda\left(\frac{1}{\mu_1}+\frac{1}{\mu_2}\right) < 2$there exists a solution (g, v) oftheaverage cost optimality equations defined in (2.3)-(2. 6).

PROOF: We prove the result for the collaborative case because the noncollaborativecase is similar. Consider the policy that is exhaustive in queue 1. Let

$$V(i,j) = \frac{h_1}{2}\left(\frac{i^2}{2\mu_1-\lambda}\right) + \frac{h_2}{2}\left(\frac{2\mu_1 i}{2\mu_1-\lambda}+2j\right)\frac{i}{2\mu_1-\lambda} + \frac{\frac{h_2}{2}\left(\frac{2\mu_1 i}{2\mu_1-\lambda}+j\right)^2}{2\mu_2\left(1-\frac{\lambda}{2\mu_1}\right)-\lambda}$$

Intuitively, this is the total cost to empty the system starting in state $(i,j)$when the arrival and service processes are replaced by deterministic continuous flows with rates λ, μ_1,and , μ_1, respectively: a fluid model. A little algebra yields

$$\mathbb{E}_\pi[V(X_{n+1})-V(x)|X_n = x] = \begin{cases} -ih_1 - jh_2 + C_1 \text{if } i \geq 2 \\ -ih_1 - jh_2 + C_2 \text{if } i = 0 \text{ and } j \geq 2 \\ -ih_1 - jh_2 + C_3 \text{if } i = 1 \text{ and } j \geq 2 \end{cases} \quad (2.10)$$

whereC_1, C_2 and $C_3$are finite constants. Let $C = max\{C_1, C_2, C_3\}$. One may note that there are four states that are not covered by the cases in (2.10): $(0,0), (0,1), (1,0)$, and $(1,1)$. Except in those states, $\mathbb{E}_\pi[V(X_{n+1})-V(x)|X_n = x] + c(x, d(x))$is bounded above by C, but since those states are finite, we can clearly choose a value of$J$ large enough to satisfy (2.9) in all cases. Applying (2.9) yields that the cost is finite.

The result is now a direct application of Theorems 3.1, 5.1, and 5.2 of (Meyn, 1997(.Because we have that the exhaustive policy in queue 1 has finite average cost, if weinitialize policy iteration by this policy, the results of Meyn (1997) state that policyiteration is guaranteed to converge to a solution to the ACOE.

The importance of the results of Propositions 2.2 and 2.3 are twofold. When the two servers are fixed, it is easy to establish that stability of the Markov chain requires $\lambda/\mu_1 < 1$and $\lambda/\mu_2 < 1$.Proposition 2.2 shows that, in fact, the stability region is extended considerably by the use of a flexible workforce. Figure 1 illustrates this fact. Proposition 2.3 allows us to use the ACOE to find the structure of optimalpolicies and eliminate from consideration (in many states) the action that assigns one server to each station (see Proposition 3.1).

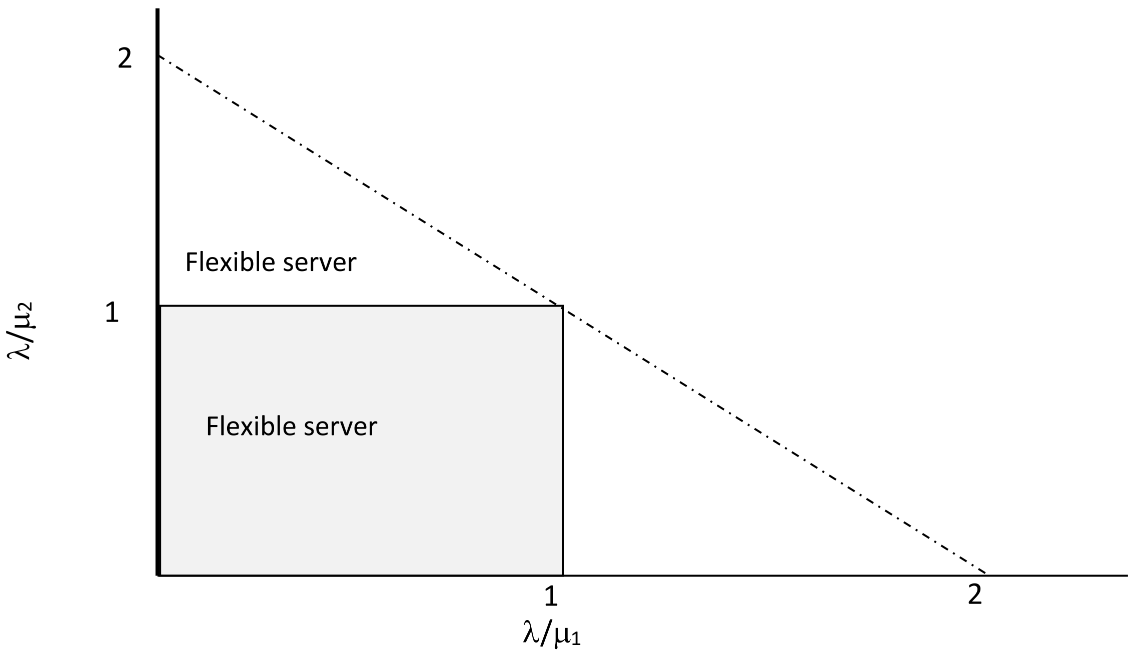

FIGURE 1: Stability condition in the system with fixed servers and the system with flexible servers.

3. AVERAGE COST OPTIMAL POLICIES

In this section, we discuss the structure of an optimal policy and give conditions under which the policy is quite simple. To do so, consider the inequality

$$\mu_1(h_1 - h_2) \leq \mu_2 h_2. \qquad (3.1)$$

Recall that we have discretized the model so that the holding cost rates can be interpreted as lump sum costs. Assume that there are customers to be served at each station and suppose we decide to assign server 1 to station 1. The left-hand side of (3.1) is then the rate of reduction in the holding cost due to making this assignment. Similarly, when we assign server 1 to station 2, the right-hand side of (3.1) denotes the rate of reduction in the holding cost. Note that this intuition is independent of the number of customers currently at each station so long as there are customers available to serve. Furthermore, it may be applied to either server (because they are identical). Analogous to steepest-descent algorithms in deterministic optimization settings, based on intuition, one might conjecture that we should choose to assign the servers so as to maximize the rate of reducing the holding costs. This leads to a policy that is exhaustive in one station or the other depending on the direction of (3.1). In the collaborative case, our intuition holds and this is, in fact, optimal. In the noncollaborative system, the intuition fails, but only when the inequality in (3.1) is reversed. In fact, the analysis of the noncollaborative system can lead to quite unintuitive optimal policies. This is illustrated by the policy depicted in Figure 2. In the figure, note that for a given number of jobs in station 2, the optimal policy assigns fewer servers to station 1 as the workload at station 1 increases.

We first show that in states when all three actions are feasible (both servers work at one of the stations, or each work at a separate station), we need not consider the action to assign one server to each station. Let $H_C \equiv \{(i,j) \in \mathbb{X} | i \geq \text{or } j \geq 1\}$ and $H_N \equiv \{(i,j) \in \mathbb{X} | i \geq 2 \text{ and } j \geq 2\}$ where $H_C \equiv (H_N)$denotes states where all three actions are feasible for the collaborative (noncollaborative) system.

PROPOSITION 3.1: For $x \in H_C(H_N)$ in the collaborative (noncollaborative) system, there exists an optimal policy that assigns both servers to a single station.

PROOF: In the collaborative and noncollaborative cases, the minimum in the optimality equations (2.3)-(2.6) for the sets $H_c$and $H_N$is

$$M = \min_{s=\{0,1,2\}} [s\mu_2 v(i-1, j+1) + 2 - s)\mu_2 v(i, j-1)$$
$$+(1 - \lambda - s\mu_1 - (2-s)\mu_2)v(i,j)].$$

TANDEM QUEUING SYSTEM

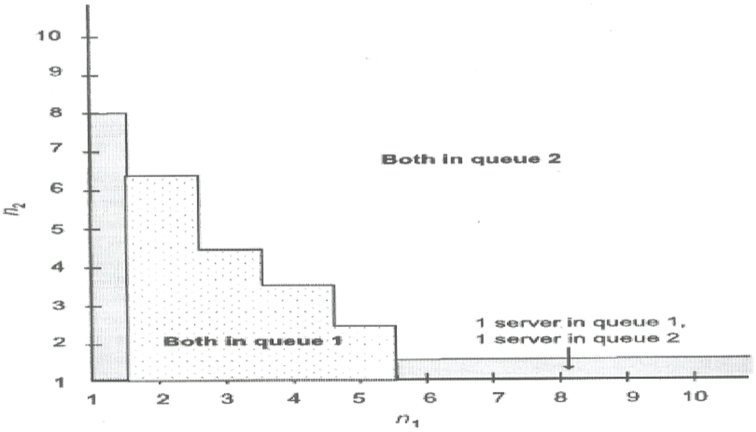

FIGURE 2: An example where the optimal policy is unintuitive: $\lambda = 0.2, \mu_1 = 0.4$, $\mu_1 = 0.4, h_1 = 1.999, \text{and} h_2 = 1$.

A little algebra yields

$$M = 2v(i,j) - 1 + (1 - \lambda - 2\mu_2)v(i,j) + \min_{s=\{0,1,2\}}\left[s\{\mu_1\left(v(i-1,j+1) + \mu_2\left(v(i,j-1)\right)\right)\}\right] \quad (3.2)$$

Because the term that is to be minimized in (3.2) is linear in s, the minimum must occur at the extreme points (i.e., s = 0 or 2).

The above result can be easily extended to the general case of an m-station tandem system with n servers. In this case, collaboration is optimal once again in all states in the collaborative system, and assigning all servers to a single station is optimal in all states where each station has at least n jobs in the non-collaborativesystem. Because the proof of this result is similar to that of Proposition 4.1 , we omit the details.

3.1. The Collaborative Case

In this subsection, we characterize an optimal policy for the collaborative case. Applying Proposition 3.1, we know that we need not consider policies that assign a server to each station. The question then becomes, given the number of customers in each queue, where should one assign both servers? We show that in the collaborative case, the answer to this question is quite simple and easy to implement. This subsection is dedicated to proving the following result.

Theorem 3.2: In the collaborative system, suppose $\lambda((1/\mu_1) + (1/\mu_2)) < 2 \text{and} \mu_1(h_1 - h_2) \geq (\leq)\mu_2 h_2$. Then, there exists an optimal policy with finite gain that is exhaustive in queue I (2).

In order to prove Theorem 3.2, we show that the result actually holds for the «-stage total expected cost problem defined by (2.1) for all n. We refer to this as the finite-horizon collaborative case. Let π^*be the exhaustive policy at queue 1. Under the assumption that$\mu_1(h_1 - h_2) \geq \mu_2 h_2$ we then have

$$V_n(\pi^*, x) \leq V_n(\pi, x) \quad (3.3)$$

for all $\pi \in \Pi$and for all n. Dividing both sides of (3.3) by $n$and taking limits as $n$approaches infinity, the average cost result follows. The finiteness of the gain follows from the results of Section 2.

To this end, it is well known that the optimal n-stage cost satisfies the finite-horizon optimality equations (FHOE). Set $v_0(i,j) = 0$and the n-stage optimality equation for states $(i \geq 1, j \geq 1)$is written as

$$\begin{aligned} v_0(i,j) &= h_1 i + h_2 j + \lambda v_{n-1}(i+1,j) \\ &+ \min_{s=\{0,1,2\}}\{s\mu_1 v_{n-1}(i-1,j+1) + (2-s)\mu_2 v_{n-1}(i,j-1) \\ &\quad + \left(1\lambda - s\mu_1 - (2-s)\mu_2\right)v_{n-1}(i,j)] \\ &\quad \text{for } i \geq \text{and } j \geq 1. \end{aligned} \quad (3.4)$$

It should be clear that the results of Proposition 3.1 extend to the finite-horizon collaborative case; that is, we need not consider policies that assign one server to each station. Thus, from the FHOE, it is optimal in the finite-horizon problem to allocate both servers to station 1 (2) when$\mu_1[v_n(i-1,j+1) - v_n(i,j)] \leq (\geq)\mu_2[v_n(i-1,j+1) - v_n(i,j)]$. The next lemma proves that this inequality holds for all n, under the same assumptions as Theorem 3.2.

LEMMA 3.3: In the finite-horizon collaborative case, suppose$\mu_1(h_1 - h_2) \geq (\leq)\mu_2 h_2$. Then, $\Delta_n(i,j) \equiv \mu_1[v_n(i-1,j+1) - v_n(i,j)] - \mu_2[v_n(i,j-1) - v_n(i,j)] \leq (\geq) 0$ for $i,j \geq 1$.

PROOF: Assume $\mu_1(h_1 - h_2) \geq \mu_2 h_2$. The proof is by induction. For$n = 0$, the result holds trivially. Assume that it holds for $n - 1$. For n, note that by the induction hypothesis, it is optimal

for both servers to serve at station 1 whenever possible.Hence, for $i \geq 1$,

$$v_1(i,j) = h_1 i + h_2 j + \lambda v_{n-1}(i+1,j) + 2\mu_1 v_{n-1}(i-1,j+1) + (1-\lambda-2\mu_1)v_{n-1}(i,j).$$

Consider the case when $i \geq 2$and$j \geq 1$. A little algebra yields

$$\mu_1[v_n i-1,j+1) - v_n(i,j)]$$
$$= \mu_1\{(h_2 - h_1) + \lambda[v_{n-1} i,j+1) - v_{n-1}(i+1,j)] + 2\mu_1[v_{n-1}(i-2,j+2) - v_{n-1}(i-1,j+1)] + (1-\lambda-2\mu_1)\,[v_{n-1}(i-1,j+1) - v_{n-1}(i,j)]\}$$
$$\mu_2[v_{n-1}(i,j-1) - v_n(i,j)]$$
$$= \mu_2\{-h_2 + \lambda[v_{n-1}(i+1,j-1) - v_{n-1}(i+1,j)] + 2\mu_1[v_{n-1}(i-1,j) - v_{n-1}(i-1,j+1)] + [1-(\lambda+2\mu_1)][v_{n-1}(i,j-1) - v_{n-1}(i,j)]\}.$$

To examine$\Delta_n(i,j)$, we note that the difference in the above expressions can beanalyzed term by term. The difference in the first terms is non-positive by the assumption that $\mu_1(h_1 - h_2) \geq \mu_2 h_2$. Each of the remaining differences is non-positive by the induction hypothesis. The algebra is similar when $i = 1$and$j \geq 1$. Thus, theresult is proven in one direction. Reversing the inequalities in the induction hypothesis yields the other direction. •

Recall that it is optimal in the finite-horizon collaborative case to allocate both servers to station 1 in state (i,j) in the «-stage problem when $\Delta_n(i,j) \leq 0$. Thus, Lemma 3.3 implies the following corollary directly.

COROLLARY 3.4: Under the assumption that $\mu_1(h_1 - h_2) \geq (\leq)\mu_2 h_2$,there exists an optimal policy to the n-stage collaborative problem that is exhaustive in station 1 (2) for all n.

PROOF OF THEOREM 3.2: Let π^*be the nonidling policy that allocates both servers to station 1 whenever possible. Then, by Corollary 3.4,

$$\frac{V_n(\pi^*,x)}{n} \leq \frac{V_n(\pi,x)}{n} \tag{3.5}$$

for all $\pi \in \Pi$.Taking the lim sum as *n*approaches infinity in (3.5) yields $\pi^*, x) \leq g(\pi,x)$for all $\pi \in \Pi$. (i.e., $\pi^*$is average cost optimal). The finiteness follows from the proof of Proposition 2.3. To show that under the assumption that $\mu_1(h_1 - h_2) \geq \mu_2 h_2$we have existence of an optimal policy that is exhaustive in queue 2, we apply the same argument. Because the gain of the policy that is exhaustive in queue 1 is finite, the finiteness must also hold for the optimal policy.

3.2. The Noncollaborative Case

The noncollaborative case is more subtle than the collaborative case. The difficulty arises because we do not allow both servers to work at the same station when there is not a separate job for the server to handle. Where possible, we have followed the same line of reasoning as the previous subsection for consistency.

Under the assumption that$\mu_1(h_1 - h_2) \geq \mu_2 h_2$, the results of the previous subsection continue to hold. However, as has been alluded to, when the inequality is reversed, an exhaustive policy is no longer guaranteed to be optimal. The main result of this section is stated in the following theorem.

THEOREM 3.5: In the noncollaborative system, suppose $\lambda((1/\mu_1) + (1/\mu_2)) < 2$and $\mu_1(h_1 - h_2) \geq \mu_2 h_2$.Then, there exists an optimal policy that is exhaustive in queue 1.

Thus, we have sufficient conditions for the optimality of an exhaustive policy in queue 1 in the non-collaborativecase that coincide with the collaborative case. This partially confirms a conjecture made in (Ahn et al, 1999). In Section 4, we discuss further the case when $\mu_1(h_1 - h_2) < \mu_2 h_2$.

Consider the finite-horizon optimality equations for the non-collaborative case for $i, j \geq 1$:

$$\begin{aligned} v_n(i,j) &= h_1 i + h_2 j + \lambda v_{n-1}(i+1,j) \\ &+ \min_{s \in \mathcal{A}} \{s\mu_1 v_{n-1}(i-1,j+1) + (2-s)\mu_2 v_{n-1}(i,j-1) \\ &\quad + \big(1\lambda - s\mu_1 - (2-s)\mu_2\big) v_{n-1}(i,j)\big] \end{aligned}$$

And$v_n(i,j) = 0$ for all $(i,j) \in \mathbb{X}$. The other states are defined in an analogous manner to the ACOE. Just as in the collaborative case, when it is possible, the FHOE imply that it is optimal to assign both servers to station 1 (2) at stage$n+1$when $\mu_1[v_n(i-1,j+1) - v_n(i,j)] - \mu_2[v_n(i,j-1) - v_n(i,j)] \leq (\geq) 0$. This leads to the following lemma.

LEMMA 3.6: In the finite-horizon noncollaborative case, suppose$\mu_1(h_1 - h_2) \geq \mu_2 h_2$.. For $n \geq 0$, we have the following:

(*a*) (i) $v_n(i,j+1) \geq v_n(i,j)$for $i,j \geq 0$

(*ii*) $v_n(i+1,j) \geq v_n(i,j+1)$for $i,j \geq 0$

(*b*)$\Delta_n(i,j) = \mu_1[v_n(i-1,j+1) - v_n(i,j)] - \mu_2[v_n(i,j-1) - (i,j)] \leq 0 for\ i,j \geq 1$.

PROOF: By induction on *n*. For *n* = 0, the claim clearly holds. Assume it holds for *n*-1. By the induction hypothesis (applied to part b), it is optimal to serve at station 1 at stage *n* whenever possible. For brevity, we will only show the case under which $i \geq 2$, although one needs to establish that the claim holds for all six cases:(i) $i \geq 2$, (ii)$i = 1, j \geq 1$,(iii) $i = 1, j = 0$,(iv)$i = 0, j \geq 2$, (v)$i = 0, j = 1, \geq$and (vi) $i = 0, j = 0. \geq$ Note that

$$\begin{aligned} v_n(i,j) = h_1 i + h_2 j + \lambda v_{n-1}(i+1,j) + 2\mu_1 v_{n-1}(i-1,j+1) \\ +(1 - \lambda - 2\mu_1) v_{n-1}(i,j) \text{if } i \geq 2. \end{aligned}$$

A little algebra yield

$$\begin{aligned} v_n(i,j) - v_n(i,j+1) = -h_2 + \lambda\,[v_{n-1}(i+1,j) - v_{n-1}(i+1,j+1)] \\ +2\mu_1[v_{n-1}(i-1,j+1) - v_{n-1}(i-1,j+2)] \\ +(1 - \lambda - 2\mu_1)\,[v_n(i,j) - v_n(i,j+1)] \\ \text{for } i \geq 2. \end{aligned}$$

Applying the induction hypothesis for each of the terms above yields part a (i). The other cases follow from similar algebra.

To show part a (ii), note that

$$\begin{aligned} v_n(i,j+1) - v_n(i+1,j) = h_2 - h_1 + \lambda\,[v_{n-1}(i+1,j+1) - v_{n-1}(i+2,j)] \\ +2\mu_1[v_{n-1}(i-1,j+2) - v_{n-1}(i,j+1)] \\ +(1 - \lambda - 2\mu_1)\,[v_{n-1}(i,j+1) - v_{n-1}(i+1,j)] \\ \text{for } i \geq 2. \end{aligned}$$

Applying similar algebra, one can show that the claim holds upon applying the induction hypothesis, the assumption that $h_2 - h_1 \leq 0$, and the previously proven part a (i) for all remaining cases. Therefore, claim a (ii) holds for all *n* as desired.

It remains to show that part (b) holds for *n*. As in the previous two parts, there are several cases to consider. However, the analysis requires a bit more detail. In the interest of brevity, we discuss only the state (1, 1). The others follow in precisely the same manner. We must show

$$\Delta_n(1,1) = \mu_1[v_n(0,2) - v_n(1,1)] - \mu_2[v_n(1,0) - v_n(1,1)] \leq 0. \qquad (3.6)$$

We have

$$\begin{aligned} \mu_1[v_n(0,2) - v_n(1,1)] - \mu_1\{(h_2 - h_1) + \lambda[v_{n-1}(1,2) - v_{n-1}(2,1)] \\ +\mu_2[v_{n-1}(0,1) - v_{n-1}(0,2,)] \\ +[1 - (\lambda + \mu_1 + \mu_2)\,][v_{n-1}(0,2) - v_{n-1}(1,1)] \end{aligned}$$

$$+\mu_2[v_{n-1}(0,1) - v_{n-1}(1,0)]\} \quad (3.7)$$

and

$$\mu_2[v_{n-1}(1,0) - v_{n-1}(1,1)] = \mu_1\{-h_2 + \lambda[v_{n-1}(2,0) - v_{n-1}(2,1)]$$
$$+\mu_1[v_{n-1}(0,1) - v_{n-1}(0,2,)]$$
$$+[1 - (\lambda + \mu_1 + \mu_2)\ ][v_{n-1}(1,0) - v_{n-1}(1,1)] \quad (3.8)$$

From the assumption that $\mu_1(h_1 - h_2) \geq \mu_2 h_2$and the induction hypothesis atstate (2, 1), the difference in each of the first three terms is non-positive (the second term cancels). The last term in (3.8) is non-positive by part a (ii). Thus, each of the claims holds for *n* and the result is proven.
The next corollary is analogous to Corollary 3.4 and follows directly from Lemma 3.6b.

COROLLARY 3.7: Under the assumption that$\mu_1(h_1 - h_2) \geq \mu_2 h_2$, there exists an optimal policy in the n-stage noncollaborative problem that is exhaustive in station 1 for all n.
Theorem 3.5 now follows just as Theorem 3.2. Because the proof is the same, it is omitted.

4. DISCUSSION AND EXTENSION

In the previous section, in the collaborative case, we proved the optimality of an exhaustive policy in either station 1 or station 2 depending on the direction of (3.1). Unfortunately, the same simple policy is not optimal for the noncollaborative case. Although the optimal policy is still exhaustive under the inequality as it appears in (3.1), when the inequality is reversed, the optimal policy in the non-collaborativecase may be quite different. This was demonstrated in Figure 2, On the other hand, our numerical study of the noncollaborative system supports the conjecture made in Ahn et al. [1] that a sufficient condition for the optimality of an exhaustive policy in queue 2 is $h_1 \geq (1 + (\mu_2/(\mu_1 + \mu_2)))h_2$.among the examples tested, and no case has been discovered to contradict this conjecture. However, we have not been able to 'prove this conjecture to date. Furthermore, upon rewriting the reverse inequality in (3.1) as $h_1 \geq (1 + (\mu_2/\mu_1))h_2$, IT is interesting to note that in each example tested when$(1 + (\mu_2/(\mu_1 + \mu_2)))h_2 < h_1 < (1 + (\mu_2/\mu_1))h_2$, a. switching curve policy was found to be optimal. That is to say, that once an optimal policy assigns both servers to station 2 in state (i, j), it will assign both to station 2 for all states (i', j')such that $i \leq i'$ and$j \leq j'$.

We have already seen in Section 2 that a flexible workforce allows for the stability region to be expanded considerably. One may wonder how important a flexible workforce can be to the overall average cost, as well as how important using an "optimal" allocation policy is to the effective use of this flexibility. We can explore the value of such flexibility by comparing the optimal policy to two candidate policies. The first is the obvious basis for comparison: where each server remains fixed at one station (i.e., no flexibility at all). We call this infixed tandem or static assignment policy. A second policy, variations of which have been recently analyzed, is called a push/putt policy. In the push/pull policy, one server is assigned to each station until such time that there is no work available at its respective station. It then moves to the other queue to aid the other server at its station. In other words, when there is no work to be done at station 2, both servers work to "pull" work to the second station. Similarly, when there is no work at station 1, both servers work to “push" work out of station 2.

Grassmann and Tavakoli (2001) studied a pull policy in which there is a finite buffer at the second queue, whereas Andradottir et al. (2001)discussed both push and pull policies (separately) and show that they perform well when average throughput is the objective. Thus, comparing the optimal policy to the static one shows the value of flexibility, whereas a comparison to the "push/pull" policy gives us a sense of the value of the "optimal" use of this flexibility. In the examples tested here, the static policy performed very poorly when compared to the optimal policy. The push/pull policy performed better than the static policy, but in some instances, it also performed poorly. Thus, investments in flexibility should be coordinated with sequencing policies that will optimally make use of such flexibility.

We created three sets of examples to test the optimal policy against these two candidate policies. In the first set (shown in Table 1), we fixed $(1+(\mu_2/(\mu_1+\mu_2)))h_2 < h_1 < (1+(\mu_2/\mu_1))h_2$.Thus, in these examples, the optimal policy was a switching curve policy in each case. For each combination of λ, μ_1, μ_2, and h_2, we chose values of $h_1$that were 20%, 50%, 80%, and 95%, respectively, of the spread between $(1+(\mu_2/(\mu_1+\mu_2)))h_2$ and $(1+(\mu_2/\mu_1))h_2$. In Table 1, we report the optimal cost as well as the cost achieved by the two polices (and the percentage difference from the optimal cost in parentheses). The benefit of a flexible workforce is substantial when compared to the fixed server system. In fact, the average sub optimality of the fixed server system is greater than 60% in the tested examples, and this is an average that excludes those instances where the. Stability condition is violated for the fixed tandem case, resulting in an average cost of infinity (as expected). By comparison, the average sub optimality of the push/pull policy is approximately 1.4%. The highest sub optimality for the push/pull was 11.47%. These results are detailed in Table 1.

TABLE 1: Performance of Optimal and Two Candidate Policies When $(1+(\mu_2/(\mu_1+\mu_2)))h_2 < h_1 < (1+(\mu_2/\mu_1))h_2$

λ	μ_1	μ_1	h_1	Cl	C2	h_1	Optimal Cost	Fixed Tandem (%)	Push/Pull (%)
0.2	**0.4**	**0.4**	**1**	**1.500**	**2.000**	**1.600**	**1.708**	**2.600(52.18)**	**1.728(1.17)**
0.2	**0.4**	**0.4**	**1**	**1.500**	**2.000**	**1.750**	**1.818**	**2.750(51.30)**	**1.829(0.58)**
0.2	**0.4**	**0.4**	**1**	**1.500**	**2.000**	**1.900**	**1.923**	**2.900 (50.80)**	**1.929(0.32)**
0.2	**0.4**	**0.4**	**1**	**1.500**	**2.000**	**1.975**	**1.973**	**2.975 (50.77)**	**1.979(0.31)**
0.2	**0.4**	**0.3**	**1**	**1.429**	**1.750**	**1.493**	**1.923**	**3.493 (81.63)**	**2.144(11.47)**
0.2	**0.4**	**0.3**	**1**	**1.429**	**1.750**	**1.589**	**2.190**	**3.589 (63.88)**	**2.214(1.11)**
0.2	**0.4**	**0.3**	**1**	**1.429**	**1.750**	**1.686**	**2.275**	**3.686 (62.03)**	**2.285 (0.45)**
0.2	**0.4**	**0.3**	**1**	**1.429**	**1.750**	**1.734**	**2.315**	**3.734(61.32)**	**2.321 (0.24)**
0.2'	**0.3**	**0.4**	**1**	**1.571**	**2.333**	**1.724**	**2.443**	**4.448 (82.06)**	**2.477(1.37)**
0.2	**0.3**	**0.4**	**1**	**1.571**	**2.333**	**1.952**	**2.695**	**4.905 (81.97)**	**2.714(0.71)**
0.2	**0.3**	**0.4**	**1**	**1.571**	**2.333**	**2.181**	**2.939**	**5.362 (82.41)**	**2.952 (0.43)**
0.2	**0.3**	**0.4**	**1**	**1.571**	**2.333**	**2.295**	**3.055**	**5.590 (83.02)**	**3.070 (0.50)**
0.2	**0.2**	**0.4**	**1**	**1.667**	**3.000**	**1.933**	**5.344**	**∞(N/A)**	**5.406(1.16)**
0.2	**0.2**	**0.4**	**1**	**1.667**	**3.000**	**2.333**	**6.337**	**∞ (N/A)**	**6.375 (0.60)**
0.2	**0.2**	**0.4**	**1**	**1.667**	**3.000**	**2.733**	**7.309**	**∞ (N/A)**	**7.344 (0.48)**
0.2	**0.2**	**0.4**	**1**	**1.667**	**3.000**	**2.933**	**7.779**	**∞ (N/A)**	**7.829 (0.64)**
0.2	**0.4**	**0.2**	**1**	**1.333**	**1.500**	**1.367**	**3.746**	**∞ (N/A)**	**3.881 (3.61)**
0.2	**0.4**	**0.2**	**1**	**1.333**	**1.500**	**1.417**	**3.841**	**∞ (N/A)**	**3.924(2.16)**
0.2	**0.4**	**0.2**	**1**	**1.333**	**1.500**	**1.467**	**3.934**	**∞ (N/A)**	**3.967 (0.83)**
0.2	**0.4**	**0.2**	**1**	**1.333**	**1.500**	**1.492**	**3.979**	**∞ (N/A)**	**3.988 (0.23)**

Note: The optimal policy is a switching curve policy.

We next considered examples when $(1+(\mu_2/\mu_1))h_2 \leq h_1$.As was shown in Theorem 3.5, an exhaustive policy in queue 1 is optimal. Values of $h_1$that are 100%, 150%, and 200% of $(1+(\mu_2/\mu_1))h_2$were considered. Once again, the system with flexible servers considerably outperforms the fixed assignment server system (on average, approximately 64.9%). In this case the push/pull policy underperformed on average by approximately 7.8%. Furthermore, there were several cases larger than 10%, with one case reaching 27.97%. Finally, we considered $(1+(\mu_2/(\mu_1+\mu_2)))h_2 \geq h_1$. This is the region where Ahn et al. (1999) conjectured that the optimal policy serves queue 2 with both servers exhaustively. Values of $h_1$were chosen that were 50%, 75%, and 100% of $(1+(\mu_2/(\mu_1+\mu_2)))h_2$.In every example tested, the optimal policy is, in fact, exhaustive in queue 2. The average sub optimality of the fixed tandem policy is approximately 73.1%, whereas that of the push/pull policy is approximately 10.73%, with the worst resulting in a significant sub optimality of 36.26%.

We have shown that the optimal policy in the collaborative case is quite simple and easy to implement. We were unable to completely characterize an optimal policy in the noncollaborative case; however, one direction is complete and the other direction is supported by the numerical study. Furthermore, this study suggests that the use of the flexible workforce can lead to significant improvements in the average cost if the flexible workforce is, in fact, allocated to tasks optimally. Even when the policy is simplified in the form of a push/pull policy, a workforce that has the freedom to work at either station can lead to significant improvements in certain cases, but not in all. We conjecture that a similar phenomenon will hold when the two-server model in this article is extended to several servers, but further research is required to characterize optimal policies in that more complicated case.

REFERENCES

Ahn, H.-S., Duenyas, I., & Zhang, R. (1999). Optimal stochastic scheduling of a two-stage tandem queue with parallel servers.*Advances in Applied Probability* 31: 1095-1117.

Andradottir, S., Ayhan, H., & Down, D.G. (2001). Server assignment policies for maximizing the steady-state throughput of finite queueing systems.*Management Science,* in press.

Andradottir, S., Ayhan, H., & Down, D.G. (2001). Dynamic server allocation for queueing networkswith flexible servers. Preprint.

Duenyas, L, Gupta, D., & Olsen, T. (1998). Control of a single server tandem queueing system withsetups. Technical Report No. 95-19, University of Michigan (to appear in *Operations Research}.*

Farrar. T. (1993).Optimal use of an extra server in a two station tandem queueing network.*IEEETransactions on Automatic Control* AC-38: 1296-1299.

Grassmann, W. &Tavakoli, J. (2001). A tandem queue with a movable server: An eigenvalue ap-proach. Preprint.

Iravani, S.M.R. &Buzacott, J.A. (1996). An TV-stage tandem queueing system attended by a movingserver with holding and switching costs. Technical Report 96-09, University of Toronto, Departmentof Industrial Engineering, Toronto, Canada.

Iravani, S.M.R., Posner, M.J.M., &Buzacott, J.A. (1997). Two-stage tandem queue attended by amoving server with holding and switching costs; static and semi-dynamic policy.*Queueing Systems*26: 203-228.

Javidi, T, Song, N., &Teneketzis, D. (2001). Expected make span minimization on identical ma-chines in two interconnected queues.*Probability in the Engineering and Informational Sciences* 15:409-443.

Kulkarni, V.G. (1995). *Modeling and analysis of stochastic systems.*London: Chapman & Hall.

Lippman, S. (1975). Applying a new device in the optimization of exponential queueing system.*Operations Research* 23: 687-710.

Meyn, S. (1997). The policy improvement algorithm for Markov decision processes. *IEEE Trans-actions on Automatic Control* AC-42(12): 1663-1680.

Meyn, S. (2001). Stability, performance evaluation, and optimization. In E. Feinberg& A. Schwartz(eds.), *Handbook of Markov decision processes: Methods and applications. Boston: Kluwer Academic.*

Meyn, S. &Tweedie, R. (1993). *Markov chains and stochastic stability.*New York: Springer-Verlag.

Pandelis, D. &Teneketzis, D. (1994). Optimal multiserver stochastic scheduling of two interconnected priority queues.*Advances in Applied Probability* 26: 258-279.
Sennott, L. (1999). *Stochastic dynamic programming and the control of queueing systems.*NewYork: Wiley.
Van Oyen, M.P., Senturk-Gel, E., &Hopp, W.J. Performance opportunity for workforce agility incollaborative and noncollaborative work systems.*HE Transactions*, in press.