

**Optimal Stochastic Scheduling of a 2-Stage Tandem Queue
with Parallel Servers**

Hyun-Soo Ahn

Department of Industrial and Operations Engineering
University of Michigan, Ann Arbor, Michigan

Izak Duenyas

Department of Industrial and Operations Engineering
University of Michigan, Ann Arbor, Michigan

Rachel Zhang

Department of Industrial and Operations Engineering
University of Michigan, Ann Arbor, Michigan

Technical Report 97-14

November 24, 1997

Optimal Stochastic Scheduling of a 2-Stage Tandem Queue with Parallel Servers

Hyun-Soo Ahn, Izak Duenyas¹, and Rachel Zhang

Abstract

We consider the optimal stochastic scheduling of a two-stage tandem queue with two parallel servers. The servers can serve either queue at any point in time and the objective is to minimize the total holding costs incurred until all jobs leave the system. We characterize sufficient and necessary conditions under which it is optimal to allocate both servers to the upstream or downstream queue. We then conduct a numerical study to investigate whether the results shown for the static case also hold for the dynamic case. Finally, we provide a numerical study that explores the benefits of having two flexible parallel servers which can work at either queue versus servers dedicated to each queue. We discuss the results' implications for cross-training workers to perform multiple tasks.

1. Introduction

In this paper, we consider stochastic scheduling of a two-stage tandem queueing system with two parallel servers. The objective is to minimize the sum of the holding costs incurred until all jobs in the system at time zero are processed and leave the system. Despite the fact that tandem queueing systems are prevalent in many manufacturing, and communications systems, there have been few papers focusing on the analysis and control of these systems. In manufacturing, these systems arise in a variety of practical situations including

¹ Corresponding author: Department of Industrial and Operations Engineering, The University of Michigan, 1205 Beal Avenue, Ann Arbor, MI 48109-2117, ducnyas@umich.edu.

worksharing systems with fewer servers than machines, (e.g., Bartholdi et al. 1995, McClain et al. 1997). However, very little is known about the optimal control of tandem queueing systems with multiple servers.

Among the few works which have considered optimal control of tandem queueing systems with multiple servers, Rosberg, Varaiya and Walrand (1982) and Hajek (1984) considered a two station tandem queueing system with one flexible server (rate μ) and one fixed server dedicated to queue 2 (rate μ_2). Farrar (1993) considered the case of a tandem queue with one fixed server for each stage and one flexible extra server which can be turned on and off. He showed that the control of the extra server can be characterized by a switching curve. He also showed that the optimal allocation policy of an extra server is transition monotone when $c_1 < c_2$. Pandelis and Teneketzis (1994a) analyzed optimal multi server stochastic scheduling of two interconnected queues and considered conditions under which exhaustive policy at the first queue is optimal. They showed that there is a sufficient condition described in terms of holding costs and the stochastic ordering of the service time distributions. This paper is very similar to ours; however it differs in several key areas: Our analysis assumes exponential processing times and also we assume that all jobs leaving queue 1 go to queue 2 and therefore our assumptions are less general than theirs; however under our more restrictive assumptions, we are able to derive sufficient **and** necessary conditions under which exhaustive policy at either queue by both servers is optimal, whereas they only derive sufficient conditions under which it is optimal to allocate both servers to the upstream queue. We further discuss the difference between the results obtained here and in the Pandelis and Teneketzis paper in the beginning of Section 5.

In other work on tandem queues, which allows for only a single server, Van Oyen and Teneketzis (1994) considered the control problem of forest networks attended by a single server with switching costs. Iravani et al. (1995) provided a performance analysis of two stage tandem queueing systems attended by a moving server with holding and switching costs, and Iravani et al. (1996) extended the analysis to N-stage tandem queues attended by a moving server. Duenyas et al. (1995) also focused on single flexible server tandem queues and provided a partial characterization of the optimal control policy and some heuristic control rules. In a different context, Weber and Stidham (1987) focused on problems where each queue in a tandem queueing system is allocated a single server but the decision maker can set the optimal service rate (where there is an operating cost associated with higher service rates).

Although there have been a few studies in optimal control analysis of systems of queues attended by multiple servers, these studies in general do not focus on tandem queues. Ross (1983) considered the minimization of makespan in a 2-stage tandem queue with 2 servers. (In contrast, we focus on the total expected cost

objective). Chang et al. (1991) focused on minimizing the expected weighted total cost of two classes of exponential jobs on multiple processors. They proved that in the optimal policy, there exists a switching curve that changes from the $c\mu$ -rule to the LEPT rule based upon the number of jobs in the system. Kampke (1987, 1989) derived conditions under which priority rules such as the $c\mu$ rule and LEPT rule are optimal in multiserver problems. Other recent references on stochastic scheduling problems (none of which focus on the scheduling of multiple servers on tandem queues) include Agrawal, Hedge and Teneketzis (1990), Gittins (1989), Hofri and Ross (1987), Pandelis and Teneketzis (1994b), Pinedo(1995), Stidham and Weber (1993), Van Oyen and Teneketzis (1992,1996).

The rest of this paper is organized as follows. In Section 2, we provide the problem formulation and show that the optimal policy can be complex. Section 3 contains our main result; mainly, the necessary and sufficient conditions under which it would be optimal to allocate both servers to queue 1 or queue 2 under the assumption of exponential processing times. The analysis in Section 3 is performed under the assumption that the two servers are never allowed to idle. In section 4, we prove that only non-idling policies can be optimal for the original problem. Therefore, the results derived in Section 3 also hold for policies where the server is actually allowed to idle. Section 5 is devoted to a numerical study which provides empirical justification for a conjecture that the results derived in Section 3 (for the static problem where the objective is to minimize the total expected cost until all jobs clear the system) also hold for the dynamic version of the problem (where the objective is the minimization of the average cost per unit time). In Section 5, we also focus on the actual benefits of having two flexible servers rather than one server dedicated to each queue, and show that this benefit can be large.

2. Problem Formulation

We consider a 2-stage tandem queue where all jobs leaving queue 1 immediately join queue 2. Jobs whose processing is completed at queue 2 leave the system. Initially, there are n_1 and n_2 jobs in the upstream and downstream queues respectively, and we assume that no external arrivals to the system will occur at any time. The system has two identical servers and the servers can process jobs at either queue at any point in time. Processing times at queue 1 (2) by either server are assumed to be exponentially distributed with mean μ_1^{-1} (μ_2^{-1}). Once a service begins, it can not be preempted. At any point in time, each one of the servers is either

processing a job from queue 1 (P_1), processing a job from queue 2 (P_2) or has completed processing a job from either queue and is ready to process a new job (which, we denote by state R). There is a positive holding cost h_1 per job per unit time incurred for holding jobs in queue 1 and a holding cost h_2 per job per unit time incurred for holding jobs at queue 2. The objective is to minimize the expected total holding costs incurred until the system clears all of the jobs present at time 0, i.e., to minimize the expression

$$E \int_0^{\infty} (h_1 n_1(t) + h_2 n_2(t)) dt$$

where $n_i(t)$, $i = 1$ and 2 , are the number of stage i jobs in the system at time t , and $n_i(0) = n_i$, $i = 1$ and 2 .

We will focus on both the non-idling version of the problem where the servers are never allowed to idle unless they have no jobs to work on and also the version where the servers are allowed to idle. We will first focus on the non-idling version of the problem, and then in Section 4, we will show that even when the servers are allowed to idle, it is in fact never optimal to idle unless the servers have no more jobs to work on.

We first focus on the non-idling version of the problem. Using the fact that neither server will idle unless it has no more jobs to process, we can formulate the problem of minimizing total expected costs until all jobs clear the system as a Markovian Decision Process. We let (i, j, X_1, X_2) be the state of the system, where i denotes the number of jobs in queue 1 and j denotes the number of jobs in queue 2. The vector (X_1, X_2) denotes the status of each machine where $X_1, X_2 \in \{P_1, P_2, R\}$. We note that since the machines are identical, we do not have to distinguish between the machines (i.e., $(X_1, X_2) = (X_2, X_1)$).

We define $V(i, j, X_1, X_2)$ as the minimal expected total cost incurred until the system is emptied, and let $V(0, 0, X_1, X_2) = 0$. Since only nonpreemptive policies are allowed, at any point in time (except time zero), we only need to make allocation decisions (of allocating a machine to queue 1 or queue 2) for at most one machine. Thus for $0 \leq i \leq n_1, 0 \leq j \leq n_2 + n_1 - i$, the value function, $V(i, j, X_1, X_2)$ satisfies the following dynamic programming equation.

$$V(i, j, P_1, R) = V(i, j, R, P_1) = \min \begin{cases} V(i, j, P_1, P_1) \\ V(i, j, P_1, P_2) \end{cases} \quad \text{for } i \geq 2, j \geq 1$$

$$V(i, j, P_2, R) = V(i, j, R, P_2) = \min \begin{cases} V(i, j, P_1, P_2) \\ V(i, j, P_2, P_2) \end{cases} \quad \text{for } i \geq 1, j \geq 2$$

$$V(i, 0, P_1, R) = V(i, 0, R, P_1) = V(i, 0, P_1, P_1) \quad \text{for } i \geq 2, j = 0$$

$$V(i, 1, P_2, R) = V(i, 1, R, P_2) = V(i, 1, P_1, P_2) \quad \text{for } i \geq 1, j = 1$$

$$V(1, j, P_1, R) = V(1, j, P_2, R) = V(1, j, P_1, P_2) \quad \text{for } i = 1, j \geq 1$$

$$V(0, j, P_2, R) = V(0, j, R, P_2) = V(0, j, P_2, P_2) \quad \text{for } i = 0, j \geq 2$$

$$V(i, j, P_1, P_1) = \frac{ih_1 + jh_2}{2\mu_1} + V(i-1, j+1, P_1, R)$$

$$V(i, j, P_1, P_2) = \frac{ih_1 + jh_2}{\mu_1 + \mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} V(i-1, j+1, P_2, R) + \frac{\mu_2}{\mu_1 + \mu_2} V(i, j-1, P_1, R)$$

$$V(i, j, P_2, P_2) = \frac{ih_1 + jh_2}{2\mu_2} + V(i, j-1, P_2, R)$$

$$V(1, 0, P_1, R) = \frac{h_1}{\mu_1} + V(0, 1, P_2, R), \quad V(0, 1, P_2, R) = \frac{h_2}{\mu_2}$$

At time zero, we assume that we can choose the allocation for both machines, and we have

$$V(n_1, n_2, R, R) = \min \begin{cases} V(n_1, n_2, P_1, R) \\ V(n_1, n_2, P_2, R) \end{cases}$$

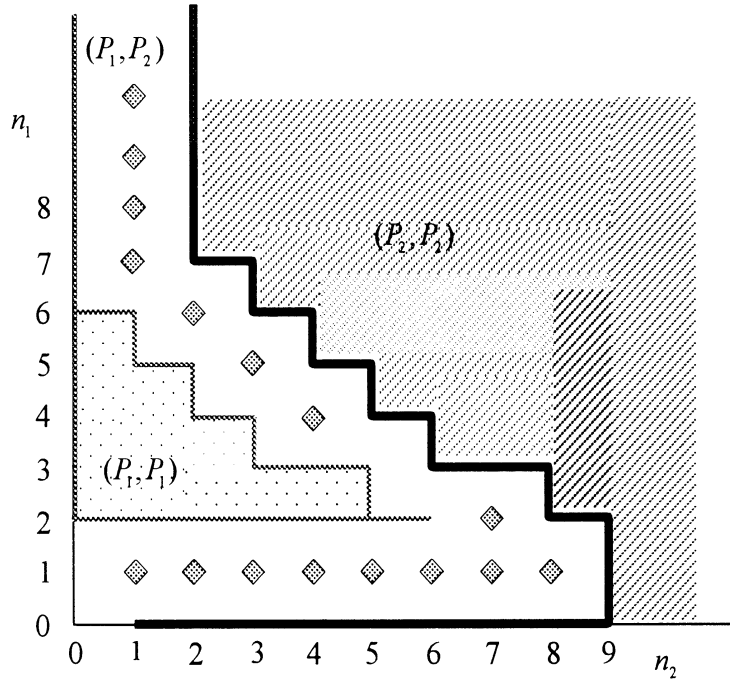


Figure 1. An example of the optimal policy.

In general, the optimal policy for this problem does not have a simple form. In particular, the allocation of machines to queues depends on the actual number of jobs at each queue. To illustrate how complicated the optimal policy can be, consider the following example with $\mu_1 = \mu_2 = 1$, $h_2 = 1$ and $h_1 = 1.995$. The optimal policy when there are two servers available is shown in Figure 1. As the figure indicates, the policy is rather complicated and can be unintuitive. For example, in state (6,2), it is optimal to allocate one server to each queue; however, when the number of jobs in queue 1 increases to 7; (i.e., in state (7,2)), it in fact becomes optimal to allocate both servers to queue 2. However, intuition suggests that if the holding cost for holding jobs

in queue 1 is sufficiently higher than the holding cost for holding jobs in queue 2, it would be optimal to allocate both servers to queue 1 and vice versa. In the next section, we present the main result of the paper which state the necessary and sufficient conditions for the optimality of allocating both servers to queue 1 or queue 2.

3. The Main Result

The main result of the paper is expressed in the following Theorem which states the necessary and sufficient conditions for the optimality of exhaustive service, by both servers at either the upstream or downstream queue.

Theorem 1:

(A) *The optimal policy always allocates both servers to queue 2 as long as there exist unserved jobs at*

$$\text{queue 2 at all states } (i,j) \text{ if and only if } h_1 \leq \left(1 + \frac{\mu_2}{\mu_1 + \mu_2}\right) h_2.$$

(B) *The optimal policy always allocates both servers to queue 1 as long as there exist unserved jobs at*

$$\text{queue 1 at all states } (i,j) \text{ if and only if } h_1 \geq \left(1 + \frac{\mu_2}{\mu_1}\right) h_2.$$

Proof: We give the detailed proofs for the sufficiency and necessity conditions of part (A). For part (B), only the necessity part is shown in detail, since the proof for the sufficiency condition is similar to the proof used in part (A). We provide the proof for part (A) in Section 3.1 and the proof for part (B) in Section 3.2.

3.1 Proof of Theorem 1 part (A).

In this subsection, we will prove that if and only if the condition, $h_1 \leq (1 + \mu_2/(\mu_1 + \mu_2))h_2$, which we will refer to as condition (A), holds then it is optimal to allocate both servers to the downstream queue (queue 2). We will use induction and direct evaluation of the value function at the boundaries to prove sufficiency. We will first prove two technical propositions and then use these propositions to prove sufficiency. Necessity will be shown by contradiction.

Proposition 1. Under condition (A), $V(1, j, P_2, R) = \min\{V(1, j, P_1, P_2), V(1, j, P_2, P_2)\} = V(1, j, P_2, P_2)$ and $Z_1(1, j) = V(1, j, P_1, P_2) - V(1, j, P_2, P_2) - V(1, j-1, P_1, R) + V(1, j-1, P_2, R) \geq 0$ for all $j \geq 2$.

Proof: The result will be shown by induction. First let $j=2$. For the induction basis of the first claim, it suffices to show that $V(1, 2, P_1, P_2) \geq V(1, 2, P_2, P_2)$ since $V(1, 2, P_2, R) = \min\{V(1, 2, P_1, P_2), V(1, 2, P_2, P_2)\}$.

Using the fact that $V(0, 3, P_2, R) = 3h_2/(2\mu_2) + V(0, 2, P_2, R)$ and $V(1, 1, P_1, R) = V(1, 1, P_2, R) = V(1, 1, P_1, P_2)$, we get

$$\begin{aligned} V(1, 2, P_1, P_2) - V(1, 2, P_2, P_2) &= \frac{h_1 + 2h_2}{\mu_1 + \mu_2} - \frac{h_1 + 2h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(0, 3, P_2, R) - V(1, 1, P_2, R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(1, 1, P_1, R) - V(1, 1, P_2, R)] \\ &= \frac{h_1 + 2h_2}{\mu_1 + \mu_2} - \frac{h_1 + 2h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} \left[\frac{3h_2}{2\mu_2} - \frac{h_1}{\mu_1} \right] = \frac{h_2(2\mu_2 + \mu_1) - h_1(\mu_1 + \mu_2)}{2\mu_2(\mu_1 + \mu_2)} \geq 0. \end{aligned}$$

where the last inequality follows directly from condition (A).

For the induction basis for the second claim, using the fact that $V(1, 1, P_1, R) = V(1, 1, P_2, R) = V(1, 1, P_1, P_2)$ and the fact (shown above) that $V(1, 2, P_1, P_2) - V(1, 2, P_2, P_2)$ is nonnegative, we have

$$Z_1(1, 2) = V(1, 2, P_1, P_2) - V(1, 2, P_2, P_2) - V(1, 1, P_1, R) + V(1, 1, P_2, R) \geq 0.$$

Now to continue the induction, suppose that $V(1, j', P_2, R) = V(1, j', P_2, P_2)$ and $Z_1(1, j') \geq 0$ for all $j' \leq j$.

Then at $(1, j+1)$, using the fact that $V(0, j+2, P_2, R) = (j+2)h_2/2\mu_2 + V(0, j+1, P_2, R)$ and using the induction hypothesis $V(1, j, P_1, P_2) - V(1, j, P_2, P_2) \geq 0$ give us,

$$\begin{aligned} &V(1, j+1, P_1, P_2) - V(1, j+1, P_2, P_2) \\ &= \frac{h_1 + (j+1)h_2}{\mu_1 + \mu_2} - \frac{h_1 + (j+1)h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(0, j+2, P_2, R) - V(1, j, P_2, R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(1, j, P_1, R) - V(1, j, P_2, R)] \\ &= \frac{h_1 + (j+1)h_2}{\mu_1 + \mu_2} - \frac{h_1 + (j+1)h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} \left[\frac{(j+2)h_2}{2\mu_2} - \frac{h_1 + jh_2}{2\mu_2} \right] \\ &\quad + \frac{\mu_1}{\mu_1 + \mu_2} [V(0, j+1, P_2, R) - V(1, j-1, P_2, R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(1, j, P_1, R) - V(1, j, P_2, R)]. \end{aligned} \tag{3.1.1}$$

It also follows from the induction hypothesis that

$$\begin{aligned} &\frac{\mu_1}{\mu_1 + \mu_2} [V(0, j+1, P_2, R) - V(1, j-1, P_2, R)] \\ &= -\frac{h_1 + jh_2}{\mu_1 + \mu_2} + \frac{h_1 + jh_2}{2\mu_2} - \frac{\mu_2}{\mu_1 + \mu_2} [V(1, j-1, P_1, R) - V(1, j-1, P_2, R)] + V(1, j, P_1, P_2) - V(1, j, P_2, P_2) \end{aligned} \tag{3.1.2}$$

After substituting (3.1.2) into (3.1.1), we have

$$\begin{aligned}
& V(1, j+1, P_1, P_2) - V(1, j+1, P_2, P_2) \\
&= V(1, j, P_1, R) - V(1, j, P_2, R) + \frac{\mu_2}{\mu_1 + \mu_2} [V(1, j, P_1, P_2) - V(1, j, P_2, P_2) + V(1, j-1, P_2, R) - V(1, j-1, P_1, R)] + \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_2 (\mu_1 + \mu_2)} \\
&= V(1, j, P_1, R) - V(1, j, P_2, R) + \frac{\mu_2}{\mu_1 + \mu_2} Z_1(1, j) + \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_2 (\mu_1 + \mu_2)} \\
&> V(1, j, P_1, R) - V(1, j, P_2, R) \\
&= V(1, j, P_1, P_2) - V(1, j, P_2, P_2) \geq 0
\end{aligned}$$

where the last inequality follows from condition (A) and the induction hypothesis that $Z_1(1, j)$ is nonnegative.

This completes the proof of the first part of the proposition.

To show $Z_1(1, j+1) \geq 0$, note that in the last argument we also showed (in the last inequality above) that $V(1, j+1, P_1, P_2) - V(1, j+1, P_2, P_2) > V(1, j, P_1, P_2) - V(1, j, P_2, P_2)$. It therefore follows that $Z_1(1, j+1)$ is nonnegative as well.

Proposition 2: Under Condition (A), for all $i \geq 2$, and all $j \geq 2$, the following are true:

i) a) $V(i, j-1, P_1, R) = V(i, j-1, P_1, P_2)$ and

b) $Z_2(i, j-1) = V(i, j-1, P_1, P_1) - V(i, j-1, P_1, P_2) + V(i-1, j, P_2, R) - V(i-1, j, P_1, R) \geq 0$.

ii) a) $V(i, j, P_2, R) = V(i, j, P_2, P_2)$ and

b) $Z_1(i, j) = V(i, j, P_1, P_2) - V(i, j, P_2, P_2) - V(i, j-1, P_1, R) + V(i, j-1, P_2, R) \geq 0$.

Proof: We will show these results by the induction on both i and j . We separate the proof into four parts. We first prove the results for $(i, j) = (2, 2)$. Then we prove that the results hold for $i = 2$ and all $j \geq 2$. Then, we prove that the results hold at state $(i, 2)$ for all $i \geq 2$. And finally we prove that the results hold for all (i, j) .

Part 1 ($i = 2, j = 2$):

To provide the induction basis, let us start with $(i, j) = (2, 2)$. To show i.a), we note $V(2, 1, P_1, R) = \min\{V(2, 1, P_1, P_1), V(2, 1, P_1, P_2)\}$ and we have,

$$\begin{aligned}
V(2, 1, P_1, P_1) - V(2, 1, P_1, P_2) &= \frac{2h_1 + h_2}{2\mu_1} - \frac{2h_1 + h_2}{\mu_1 + \mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(1, 2, P_1, R) - V(1, 2, P_2, R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(1, 2, P_1, R) - V(2, 0, P_1, R)] \\
&= \frac{2h_1 + h_2}{2\mu_1} - \frac{2h_1 + h_2}{\mu_1 + \mu_2} + [V(1, 2, P_1, R) - V(1, 2, P_2, R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(1, 2, P_2, R) - V(2, 0, P_1, R)] \quad (3.1.3)
\end{aligned}$$

Noting the fact that $V(2,0, P_1, R) = 2h_1/2\mu_1 + V(1,1, P_1, R)$, and $V(1,2, P_2, R) = (h_1 + 2h_2)/(2\mu_2) + V(1,1, P_2, R)$

(from Proposition 1), after substitutions, we get

$$\begin{aligned} & V(2,1, P_1, P_1) - V(2,1, P_1, P_2) \\ &= \frac{2h_1 + h_2}{2\mu_1} - \frac{2h_1 + h_2}{\mu_1 + \mu_2} + \frac{\mu_2}{\mu_1 + \mu_2} \left[\frac{h_1 + 2h_2}{2\mu_2} - \frac{2h_1}{2\mu_1} \right] \\ &+ \frac{\mu_2}{\mu_1 + \mu_2} [V(1,1, P_2, R) - V(1,1, P_1, R)] + [V(1,2, P_1, R) - V(1,2, P_2, R)] \end{aligned}$$

Note that $V(1,1, P_2, R) = V(1,1, P_1, R) = V(1,1, P_1, P_2)$. Thus, we have the following:

$$\begin{aligned} & V(2,1, P_1, P_1) - V(2,1, P_1, P_2) \\ &= \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_1 (\mu_1 + \mu_2)} + [V(1,2, P_1, R) - V(1,2, P_2, R)] \\ &> [V(1,2, P_1, R) - V(1,2, P_2, R)] = V(1,2, P_1, P_2) - V(1,2, P_2, P_2) \geq 0. \end{aligned} \tag{3.1.4}$$

From (3.1.4), we also get

$$V(2,1, P_1, P_1) - V(2,1, P_1, P_2) + V(1,2, P_2, R) - V(1,2, P_1, R) = Z_2(2,1) \geq 0.$$

which also proves *i.b*).

To prove *ii.a*), we first note that

$$\begin{aligned} & V(2,2, P_1, P_2) - V(2,2, P_2, P_2) \\ &= \frac{2h_1 + 2h_2}{\mu_1 + \mu_2} - \frac{2h_1 + 2h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(1,3, P_2, R) - V(2,1, P_2, R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(2,1, P_1, R) - V(2,1, P_2, R)] \\ &= \frac{2h_1 + 2h_2}{\mu_1 + \mu_2} - \frac{2h_1 + 2h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(1,3, P_2, R) - V(2,1, P_1, R)] + [V(2,1, P_1, R) - V(2,1, P_2, R)] \end{aligned} \tag{3.1.5}$$

From condition *i.a*), we have

$$V(2,1, P_1, R) = V(2,1, P_1, P_2) = \frac{[2h_1 + h_2]}{\mu_1 + \mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} V(1,2, P_2, R) + \frac{\mu_2}{\mu_1 + \mu_2} V(2,0, P_1, R). \tag{3.1.6}$$

Also from Proposition 1,

$$V(1,3, P_2, R) = V(1,3, P_2, P_2) = [h_1 + 3h_2]/(2\mu_2) + V(1,2, P_2, R). \tag{3.1.7}$$

Substituting (3.1.6) and (3.1.7) into (3.1.5), we get

$$\begin{aligned} & V(2,2, P_1, P_2) - V(2,2, P_2, P_2) \\ &= \frac{2h_1 + 2h_2}{\mu_1 + \mu_2} - \frac{2h_1 + 2h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} \left[\frac{h_1 + 3h_2}{2\mu_2} - \frac{2h_1 + h_2}{\mu_1 + \mu_2} \right] \\ &+ \frac{\mu_1}{\mu_1 + \mu_2} \frac{\mu_2}{\mu_1 + \mu_2} [V(1,2, P_2, R) - V(2,0, P_1, R)] + [V(2,1, P_1, R) - V(2,1, P_2, R)]. \end{aligned}$$

Then by using the fact from (3.1.3) that

$$\begin{aligned} & \frac{\mu_2}{\mu_1 + \mu_2} [V(1,2, P_2, R) - V(2,0, P_1, R)] - \frac{2h_1 + h_2}{\mu_1 + \mu_2} + \frac{2h_1 + h_2}{2\mu_1} \\ &= [V(2,1, P_1, P_1) - V(2,1, P_1, P_2) + V(1,2, P_2, R) - V(1,2, P_1, R)] = Z_2(2,1) \geq 0. \end{aligned} \tag{3.1.8}$$

and substituting this identity into (3.1.5), we get

$$\begin{aligned} & V(2,2,P_1,P_2) - V(2,2,P_2,P_2) \\ &= \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_2(\mu_1 + \mu_2)} + \frac{\mu_1}{\mu_1 + \mu_2} Z_2(2,1) + [V(2,1,P_1,R) - V(2,1,P_2,R)] \\ &> [V(2,1,P_1,R) - V(2,1,P_2,R)] = V(2,1,P_1,P_2) - V(2,1,P_1,P_2) = 0. \end{aligned}$$

To prove that $Z_1(2,2) \geq 0$, from the last inequality, we have

$$V(2,2,P_1,P_2) - V(2,2,P_2,P_2) - V(2,1,P_1,R) + V(2,1,P_2,R) = Z_1(2,2) \geq 0$$

which is the desired result.

Part 2. ($i = 2$, all j)

In Part 1, we showed that the results in Proposition 2 hold for ($i = 2, j = 2$). We now use the induction to show that Proposition 2 holds for $i = 2$ and all values of j . Assume that the proposition holds for $i = 2$ and for all $j_1 \leq j$, we now show that it must also hold for $i = 2$ and $j_1 = j + 1$. (We only show the proof for part *i*) as the proof for part *ii*) is identical). We first note that

$$\begin{aligned} & V(2,j,P_1,P_1) - V(2,j,P_1,P_2) \\ &= \frac{2h_1 + jh_2}{2\mu_1} - \frac{2h_1 + jh_2}{\mu_1 + \mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(1,j+1,P_1,R) - V(1,j+1,P_2,R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(1,j+1,P_1,R) - V(2,j-1,P_1,R)] \\ &= \frac{2h_1 + jh_2}{2\mu_1} - \frac{2h_1 + jh_2}{\mu_1 + \mu_2} + \frac{\mu_2}{\mu_1 + \mu_2} [V(1,j+1,P_2,R) - V(2,j-1,P_1,R)] + [V(1,j+1,P_1,R) - V(1,j+1,P_2,R)] \quad (3.1.9) \end{aligned}$$

From the induction hypothesis, we have

$$V(2,j-1,P_1,R) = (2h_1 + (j-1)h_2)/(\mu_1 + \mu_2) + \mu_1/(\mu_1 + \mu_2)V(1,j,P_2,R) + \mu_2/(\mu_1 + \mu_2)V(2,j-2,P_1,R). \quad (3.1.10)$$

and from Proposition 1, we have

$$V(1,j+1,P_2,R) = [h_1 + (j+1)h_2]/(2\mu_2) + V(1,j,P_2,R) \quad (3.1.11)$$

After substituting (3.1.10) and (3.1.11) into (3.1.9), we get

$$\begin{aligned} & V(2,j,P_1,P_1) - V(2,j,P_1,P_2) \\ &= \frac{2h_1 + jh_2}{2\mu_1} - \frac{2h_1 + jh_2}{\mu_1 + \mu_2} + \frac{\mu_2}{\mu_1 + \mu_2} \left[\frac{h_1 + (j+1)h_2}{2\mu_2} - \frac{2h_1 + (j-1)h_2}{\mu_1 + \mu_2} \right] \\ &\quad + \left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^2 [V(1,j,P_2,R) - V(2,j-2,P_1,R)] + [V(1,j+1,P_1,R) - V(1,j+1,P_2,R)] \end{aligned}$$

Noting from (3.1.9) for $j-1$ that

$$\begin{aligned}
& \frac{\mu_2}{\mu_1 + \mu_2} [V(1, j, P_2, R) - V(2, j - 2, P_1, R)] \\
&= \frac{2h_1 + (j - 1)h_2}{\mu_1 + \mu_2} - \frac{2h_1 + (j - 1)h_2}{2\mu_1} + V(2, j - 1, P_1, P_1) - V(2, j - 1, P_1, P_2) + V(1, j, P_2, R) - V(1, j, P_1, R) \\
&= \frac{2h_1 + (j - 1)h_2}{\mu_1 + \mu_2} - \frac{2h_1 + (j - 1)h_2}{2\mu_1} + Z_2(2, j - 1)
\end{aligned} \tag{3.1.12}$$

we get,

$$\begin{aligned}
& V(2, j, P_1, P_1) - V(2, j, P_1, P_2) \\
&= \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_1 (\mu_1 + \mu_2)} + \frac{\mu_2}{\mu_1 + \mu_2} Z_2(2, j - 1) + [V(1, j + 1, P_1, R) - V(1, j + 1, P_2, R)] \\
&> V(1, j + 1, P_1, R) - V(1, j + 1, P_2, R) \geq 0.
\end{aligned}$$

where the first inequality follows from the induction assumption and the second inequality follows from Proposition 1. We have thus shown *i.a)* and we note that *i.b)* also follows since the equation above implies

$$V(2, j, P_1, P_1) - V(2, j, P_1, P_2) - V(1, j + 1, P_1, R) + V(1, j + 1, P_2, R) = Z_2(2, j) \geq 0.$$

The proof for part *ii)* is similar and is omitted.

Part 3. (All $i, j = 2$)

We showed in Part 1 that Proposition 2 holds for $i = 2, j = 2$. Suppose that it is in fact optimal to allocate both servers to queue 2 (when queue 2 has 2 or more jobs) for all $i_1 \leq i, j = 2$. Here we prove that the result still holds at $(i + 1, 2)$. We again only show the proof for part *i)*, as the proof for *ii)* is identical.

For part *i)*, we need to evaluate $V(i + 1, 1, P_1, P_1) - V(i + 1, 1, P_1, P_2)$,

$$\begin{aligned}
& V(i + 1, 1, P_1, P_1) - V(i + 1, 1, P_1, P_2) \\
&= \frac{(i + 1)h_1 + h_2}{2\mu_1} - \frac{(i + 1)h_1 + h_2}{\mu_1 + \mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(i, 2, P_1, R) - V(i, 2, P_2, R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(i, 2, P_1, R) - V(i + 1, 0, P_1, R)] \\
&= \frac{(i + 1)h_1 + h_2}{2\mu_1} - \frac{(i + 1)h_1 + h_2}{\mu_1 + \mu_2} + [V(i, 2, P_1, R) - V(i, 2, P_2, R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(i, 2, P_2, R) - V(i + 1, 0, P_1, R)]
\end{aligned}$$

First, note that $V(i + 1, 0, P_1, R) = (i + 1)h_1 / 2\mu_1 + V(i, 1, P_1, R)$. Also, we have $V(i, 2, P_2, R) = \frac{ih_1 + 2h_2}{2\mu_2} + V(i, 1, P_2, R)$

from the induction hypothesis on i . Thus,

$$\begin{aligned}
& V(i+1,1,P_1,P_1) - V(i+1,1,P_1,P_2) \\
&= \frac{(i+1)h_1 + h_2}{2\mu_1} - \frac{(i+1)h_1 + h_2}{\mu_1 + \mu_2} + \frac{\mu_2}{\mu_1 + \mu_2} \left[\frac{ih_1 + 2h_2}{2\mu_2} - \frac{(i+1)h_1}{2\mu_1} \right] \\
&+ \frac{\mu_2}{\mu_1 + \mu_2} [V(i,1,P_2,R) - V(i,1,P_1,R)] + [V(i,2,P_1,R) - V(i,2,P_2,R)]
\end{aligned}$$

Note that $V(i,1,P_2,R) = V(i,1,P_1,R) = V(i,1,P_1,P_2)$, because of the induction hypothesis on i . Then,

$$\begin{aligned}
& V(i+1,1,P_1,P_1) - V(i+1,1,P_1,P_2) \\
&= \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_1 (\mu_1 + \mu_2)} + [V(i,2,P_1,R) - V(i,2,P_2,R)] > [V(i,2,P_1,R) - V(i,2,P_2,R)] \geq 0.
\end{aligned}$$

To prove $Z_2(i+1,1) \geq 0$, using the inequality in the last equation;

$$V(i+1,1,P_1,P_1) - V(i+1,1,P_1,P_2) + V(i,2,P_2,R) - V(i,2,P_1,R) = Z_2(i+1,1) \geq 0.$$

Part 4. (All i and j)

We showed in parts 1, 2 and 3 that the result holds for ($i=2$ and all j) and ($j=2$ and all i). To complete the induction argument, now suppose that the proposition holds for ($i_1 \leq i$, all $j \geq 2$), and also for ($i_1 = i+1$ and $j_1 \leq j$), we now show that it also holds for $(i+1, j+1)$, and therefore by induction holds for all i and j . We once again omit the argument for ii) as it is similar to that of i). To show i) we again note that

$$\begin{aligned}
& V(i+1,j,P_1,P_1) - V(i+1,j,P_1,P_2) \\
&= \frac{(i+1)h_1 + jh_2}{2\mu_1} - \frac{(i+1)h_1 + jh_2}{\mu_1 + \mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(i,j+1,P_1,R) - V(i,j+1,P_2,R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(i,j+1,P_1,R) - V(i+1,j-1,P_1,R)] \\
&= \frac{(i+1)h_1 + jh_2}{2\mu_1} - \frac{(i+1)h_1 + jh_2}{\mu_1 + \mu_2} + [V(i,j+1,P_1,R) - V(i,j+1,P_2,R)] + \frac{\mu_2}{\mu_1 + \mu_2} [V(i,j+1,P_2,R) - V(i+1,j-1,P_1,R)] \quad (3.1.13)
\end{aligned}$$

From the induction hypothesis, we have

$$V(i,j+1,P_2,R) = [ih_1 + (j+1)h_2] / (2\mu_2) + V(i,j,P_2,R), \quad (3.1.14)$$

$$V(i+1,j-1,P_1,R) = ((i+1)h_1 + (j-1)h_2) / (\mu_1 + \mu_2) + \mu_1 / (\mu_1 + \mu_2) V(i,j,P_2,R) + \mu_2 / (\mu_1 + \mu_2) V(i+1,j-2,P_1,R). \quad (3.1.15)$$

After substituting (3.1.14) and (3.1.15) into (3.1.13), we get

$$\begin{aligned}
& V(i+1,j,P_1,P_1) - V(i+1,j,P_1,P_2) \\
&= \frac{(i+1)h_1 + jh_2}{2\mu_1} - \frac{(i+1)h_1 + jh_2}{\mu_1 + \mu_2} + \frac{\mu_2}{\mu_1 + \mu_2} \left[\frac{ih_1 + (j+1)h_2}{2\mu_2} - \frac{(i+1)h_1 + (j-1)h_2}{\mu_1 + \mu_2} \right] \\
&+ \left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^2 [V(i,j,P_2,R) - V(i+1,j-2,P_1,R)] + [V(i,j+1,P_1,R) - V(i,j+1,P_2,R)]
\end{aligned}$$

Then using the fact (from (3.13) by replacing j by $j-1$) that,

$$\begin{aligned}
& \frac{\mu_2}{\mu_1 + \mu_2} [V(i, j, P_2, R) - V(i+1, j-2, P_1, R)] \\
&= \frac{(i+1)h_1 + (j-1)h_2}{\mu_1 + \mu_2} - \frac{(i+1)h_1 + (j-1)h_2}{2\mu_1} + V(i+1, j-1, P_1, P_1) - V(i+1, j-1, P_1, P_2) + V(i, j, P_2, R) - V(i, j, P_1, R) \\
&= \frac{(i+1)h_1 + (j-1)h_2}{\mu_1 + \mu_2} - \frac{(i+1)h_1 + (j-1)h_2}{2\mu_1} + Z_2(i+1, j-1)
\end{aligned} \tag{3.1.16}$$

we get

$$\begin{aligned}
& V(i+1, j, P_1, P_1) - V(i+1, j, P_1, P_2) \\
&= \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_1 (\mu_1 + \mu_2)} + \frac{\mu_2}{\mu_1 + \mu_2} [V(i+1, j-1, P_1, P_1) - V(i+1, j-1, P_1, P_2) + V(i, j, P_2, R) - V(i, j, P_1, R)] \\
&\quad + [V(i, j+1, P_1, R) - V(i, j+1, P_2, R)] \\
&= \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_1 (\mu_1 + \mu_2)} + \frac{\mu_2}{\mu_1 + \mu_2} Z_2(i+1, j-1) + [V(i, j+1, P_1, R) - V(i, j+1, P_2, R)] \\
&> V(i, j+1, P_1, R) - V(i, j+1, P_2, R) \geq 0.
\end{aligned}$$

To show $Z_2(i+1, j) \geq 0$, from the last inequality,

$$V(i+1, j, P_1, P_1) - V(i+1, j, P_1, P_2) + V(i, j+1, P_2, R) - V(i, j+1, P_1, R) = Z_2(i+1, j) \geq 0,$$

and therefore we have shown Proposition 2.

To complete the proof of Theorem 1 part (A) we note that the sufficiency directly follows from Propositions 1 and 2. We prove necessity by contradiction. Suppose that it is in fact optimal to allocate both servers to queue 2 but that $h_1 > (1 + \frac{\mu_2}{\mu_1 + \mu_2})h_2$. Consider a problem instance where $i=1$, $j=2$, one server is working on a job

from queue 2 and the other server is ready for service. At this point, if we compare the value of assigning the second server to queue 2 as well versus assigning it to queue 1, we get (by simple algebra),

$$V(1, 2, P_1, P_2) - V(1, 2, P_2, P_2) = \frac{h_2(2\mu_2 + \mu_1) - h_1(\mu_1 + \mu_2)}{2\mu_2(\mu_1 + \mu_2)} < 0. \text{ However, this contradicts the fact that assigning}$$

both servers to queue 2 is optimal and therefore the necessity is shown as well. \blacklozenge

3.2 Proof of Theorem 1 part (B).

In this section, we provide the proof for part (B) of Theorem 1. Since the proof of sufficiency is very similar to that of part (A), we provide a summary of the proof. We give the full details for necessity. The sufficiency proof hinges on the following three propositions which are entirely analogous to Propositions 1 and 2.

Proposition 3: Under condition (B), the following are true:

$$i) V(1, j, P_2, R) = \min \begin{cases} V(1, j, P_1, P_2) \\ V(1, j, P_2, P_2) \end{cases} = V(1, j, P_1, P_2) \text{ for all } j \geq 2 .$$

$$ii) V(2, j, P_1, R) = \min \{V(2, j, P_1, P_1), V(2, j, P_1, P_2)\} = V(2, j, P_1, P_1) \text{ and}$$

$$Z_2(2, j) = V(2, j, P_1, P_1) - V(2, j, P_1, P_2) + V(1, j+1, P_2, R) - V(1, j+1, P_1, R) \leq 0 \text{ for all } j \geq 1.$$

Proposition 4: Under condition (B), the following are true for $i, j \geq 2$:

$$i) V(i, j, P_2, R) = \min \{V(i, j, P_1, P_2), V(i, j, P_2, P_2)\} = V(i, j, P_1, P_2)$$

$$ii) a) V(i+1, j-1, P_1, R) = V(i+1, j-1, P_1, P_1)$$

$$b) Z_2(i+1, j-1) = V(i+1, j-1, P_1, P_1) - V(i+1, j-1, P_1, P_2) + V(i, j, P_2, R) - V(i, j, P_1, R) \leq 0$$

Proof: The proofs of Propositions 3 and 4 are virtually identical to those of Propositions 1 and 2 and are omitted in the interest of space. ♦

The sufficiency in part (B) of Theorem 1 follows directly from Propositions 3 and 4. We now provide the proof of necessity in detail. We will again prove this by contradiction. That is, suppose that an optimal policy allocates servers to queue 1 as long as there are unserved jobs at queue 1 but that $h_1 < (1 + \mu_2 / \mu_1)h_2$. Now, from Theorem 1 part (A) which was shown above, we already know that when $h_1 \leq (1 + \mu_2 / (\mu_1 + \mu_2))h_2$, it is in fact optimal to allocate both servers to queue 2 and therefore in that case the result follows by contradiction immediately. We therefore need to focus only in the region $(1 + \mu_2 / (\mu_1 + \mu_2)) < h_1 < (1 + \mu_2 / \mu_1)h_2$. Now, consider the case with $n_1 = 1$. The optimality of the exhaustive policy at queue 1 implies $V(1, j, P_2, R) = V(1, j, P_1, P_2) \leq V(1, j, P_2, P_2)$ for all $j \geq 2$.

Then, pick n_2 such that

$$n_2 = \min \left\{ j \in \mathbb{N}; \frac{h_2 - h_1}{2\mu_2} + \frac{h_2}{2\mu_1} \left[1 - \left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^{j-1} \right] > 0 \right\}.$$

The existence of a finite $n_2 > 2$ is shown easily. We note that n_2 can not be 1 since for $j=1$, the expression above yields $(h_2 - h_1) / 2\mu_2$ which is negative since $h_1 > (1 + \mu_2 / (\mu_1 + \mu_2))h_2 > h_2$. Similarly, when $j=2$, the expression above yields $\frac{(h_2 - h_1)(\mu_1 + \mu_2) + h_2\mu_2}{2\mu_2(\mu_1 + \mu_2)}$ which is again negative because $h_1 > (1 + \mu_2 / (\mu_1 + \mu_2))h_2$. On

the other hand, since $\mu_2 / (\mu_1 + \mu_2) < 1$, $\lim_{j \rightarrow \infty} (\mu_2 / (\mu_1 + \mu_2))^j = 0$. We note that $(h_2 - h_1) / (2\mu_2) + h_2 / (2\mu_1) [1 - (\mu_2 / (\mu_1 + \mu_2))^{j-1}]$ is increasing in j and converges to

$[\mu_2 h_2 + \mu_1 (h_2 - h_1)] / (2\mu_1 \mu_2) > 0$ which is strictly positive because of $h_1 < (1 + \mu_2 / \mu_1) h_2$. Therefore, a finite value of $n_2 > 2$ exists.

We now focus on the difference of $V(1, n_2, P_1, P_2)$ and $V(1, n_2, P_2, P_2)$ and we have

$$\begin{aligned} & V(1, n_2, P_1, P_2) - V(1, n_2, P_2, P_2) \\ &= \frac{h_1 + n_2 h_2}{\mu_1 + \mu_2} - \frac{h_1 + n_2 h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} V(0, n_2 + 1, P_2, R) + \frac{\mu_2}{\mu_1 + \mu_2} V(1, n_2 - 1, P_1, R) - V(1, n_2 - 1, P_2, R). \end{aligned} \quad (3.2.1)$$

We also note that $V(1, n_2 - 1, P_1, R) = V(1, n_2 - 1, P_1, P_2)$, and $V(0, n_2 + 1, P_2, R) = \frac{(n_2 + 1)h_2}{2\mu_2} + V(0, n_2, P_2, R)$.

Also, since allocating servers to unserved jobs at queue 1 is optimal, we have

$$V(1, n_2 - 1, P_2, R) = V(1, n_2 - 1, P_1, P_2) = \frac{h_1 + (n_2 - 1)h_2}{\mu_1 + \mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} V(0, n_2, P_2, R) + \frac{\mu_2}{\mu_1 + \mu_2} V(1, n_2 - 2, P_1, R).$$

which after substitutions yields;

$$\begin{aligned} & V(1, n_2, P_1, P_2) - V(1, n_2, P_2, P_2) \\ &= \frac{h_1 + n_2 h_2}{\mu_1 + \mu_2} - \frac{h_1 + n_2 h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} [V(0, n_2 + 1, P_2, R) - V(1, n_2 - 1, P_2, R)] \\ &= \frac{h_1 + n_2 h_2}{\mu_1 + \mu_2} - \frac{h_1 + n_2 h_2}{2\mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} \left[\frac{(n_2 + 1)h_2}{2\mu_2} - \frac{h_1 + (n_2 - 1)h_2}{\mu_1 + \mu_2} \right] + \frac{\mu_1}{\mu_1 + \mu_2} \frac{\mu_2}{\mu_1 + \mu_2} [V(0, n_2, P_2, R) - V(1, n_2 - 2, P_1, R)] \end{aligned} \quad (3.2.2)$$

Substituting the expression

$$\frac{\mu_1}{\mu_1 + \mu_2} [V(0, n_2, P_2, R) - V(1, n_2 - 2, P_2, R)] = V(1, n_2 - 1, P_1, P_2) - V(1, n_2 - 1, P_2, P_2) - \frac{h_1 + (n_2 - 1)h_2}{\mu_1 + \mu_2} + \frac{h_1 + (n_2 - 1)h_2}{2\mu_2}$$

(which follows from the optimality of assigning servers to queue 1 as long as there exist unserved jobs in queue 1) into (3.2.2), we get

$$V(1, n_2, P_1, P_2) - V(1, n_2, P_2, P_2) = \frac{\mu_2}{\mu_1 + \mu_2} [V(1, n_2 - 1, P_1, P_2) - V(1, n_2 - 1, P_2, P_2)] + \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_2 (\mu_1 + \mu_2)} \quad (3.2.3)$$

By recursively applying (3.2.3) to $V(1, n_2 - 1, P_1, P_2) - V(1, n_2 - 1, P_2, P_2)$, $V(1, n_2 - 2, P_1, P_2) -$

$V(1, n_2 - 2, P_2, P_2)$, and so on, we get

$$\begin{aligned} & V(1, n_2, P_1, P_2) - V(1, n_2, P_2, P_2) \\ &= \frac{\mu_2}{\mu_1 + \mu_2} \left[\left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^{n_2 - 3} [V(1, 2, P_1, P_2) - V(1, 2, P_2, P_2)] + \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_2 (\mu_1 + \mu_2)} \sum_{k=1}^{n_2 - 3} \left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^{k-1} \right] + \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_2 (\mu_1 + \mu_2)} \\ &= \frac{\mu_2}{\mu_1 + \mu_2} \left[\left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^{n_2 - 3} \frac{\mu_2 (h_2 - h_1)}{2\mu_2 (\mu_1 + \mu_2)} + \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_2 (\mu_1 + \mu_2)} \sum_{k=1}^{n_2 - 2} \left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^{k-1} \right] + \frac{\mu_2 h_2 + \mu_1 (h_2 - h_1)}{2\mu_2 (\mu_1 + \mu_2)}. \end{aligned}$$

After a lot of algebra, this simplifies to

$$V(1, n_2, P_1, P_2) - V(1, n_2, P_2, P_2) = \frac{h_2 - h_1}{2\mu_2} + \frac{h_2}{2\mu_1} \left(1 - \left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^{n_2 - 1} \right). \quad (3.2.4)$$

However, we note that the right hand side of (3.2.4) is strictly greater than zero from the definition of n_2 . However, this implies that $V(1, n_2, P_1, P_2) > V(1, n_2, P_2, P_2)$, which contradicts the fact that it is optimal to allocate servers to queue 1 when there exist unserved jobs at queue 1. We therefore have also shown the necessity result. ♦

4. Optimality of Non-Idling Policies

In the previous section, we assumed that the servers are never allowed to idle. The only way a server can idle is if there are no more jobs for the server to serve. Unenforced idling is not permitted. In this section, we prove that even when unenforced idling is permitted, it is in fact never optimal for the servers to idle, and thus the optimality results in Theorem 1 extend to policies where idling is permitted. This is stated in the next theorem.

Theorem 2: *The optimal policy for the problem where idling is permitted is a non-idling policy and the results in Theorem 1 hold when idling is permitted.*

Proof: It is very straightforward (through a very simple interchange argument) to show that it can never be optimal to have both servers idling at the same time so we only focus on the case where one server is working on a job from either queue and the other one is either ready to start a new job or to idle. We divide the proof into two cases ($h_1 \geq h_2$ and $h_1 < h_2$):

Case i) ($h_1 \geq h_2$)

We will prove the result in this case through an interchange argument. In order to set up the interchange argument, we let $J_{l,m}$ be the type (i.e., job from queue 1 or queue 2) of the m -th job processed by server l . For example, if $J_{l,m} = 1$, this means that m -th job server l processes is in queue 1. The processing time of the job $J_{l,m}$ is denoted as $P_{l,m}$. Also, we let $C^\Pi(s)$, $s \geq 0$, be the holding cost rate at time s under the policy Π and $V^\Pi = \int_0^\infty C^\Pi(s) ds$ be the expected total cost under the policy Π . Consider a policy $\tilde{\Pi}$ where one of the servers is working on a job, and the other server is ready to process a job. Under this policy, the second server

idles. We will assume, without loss of generality, that the second server idles for a duration of time I , which is no longer than the remaining processing time of the job the first server is processing which we will denote by $P_{1,1}(0)$. (We note that the second server could idle for a longer duration, in which case, we could make the same argument by dividing the idling periods into two periods I_2 and I_1 , and repeating the same argument for both periods (see Figure 2.) We therefore assume without loss of generality that $I \leq P_{1,1}(0)$. Our proof shows that $\tilde{\Pi}$ can not be optimal by constructing a policy Π where the idling is postponed to the future. But, since the same argument can be repeated at all times, this means that it is never optimal to idle except when there are no jobs to serve.

Now, we show that policy $\tilde{\Pi}$ can not be optimal. We divide the proof into 4 cases depending on which jobs the first server serves at time zero and the second server serves after the completion of its idling period under policy $\tilde{\Pi}$.

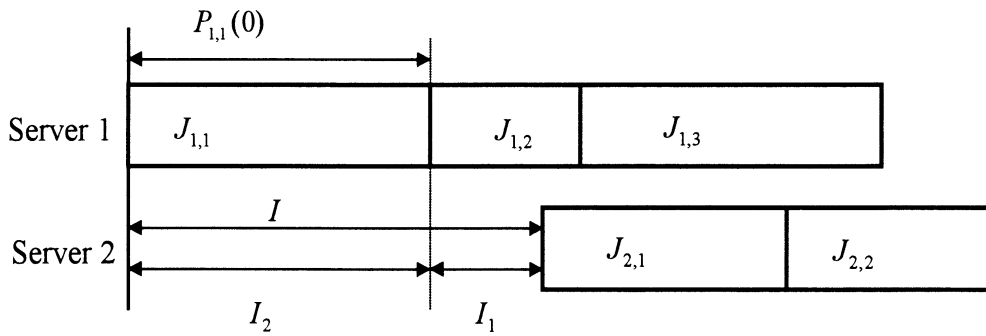


Figure 2. $I > P_{1,1}(0)$

Case 1. ($J_{1,1} = 2$)

Suppose that at time zero, the first server was busy serving a job from queue 2 (i.e., $J_{1,1} = 2$) and server 2 idles as in Figure 3.a) under policy $\tilde{\Pi}$ even though there exist other jobs it can process. We compare this policy with another policy Π where server 2 first process $J_{2,1}$ (where this job can be from either queue) and then idles as in Figure 3.b) (i.e., the idling period is shifted). The two policies are completely identical except for this shift of the idling period and the serving of job $J_{2,1}$ by server 2. We denote the time at which the two situations become identical by time t where $t = I + P_{1,1}$.

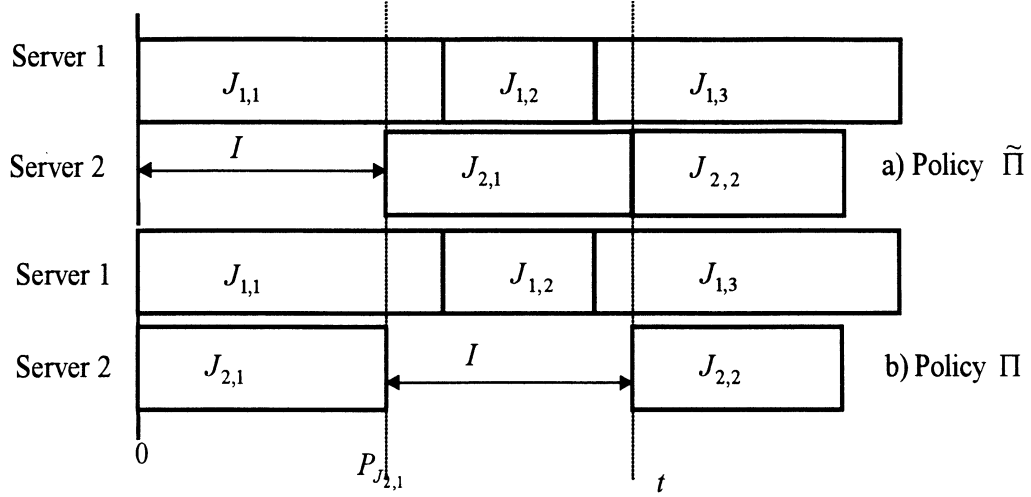


Figure 3. Case 1,2 and 3

Since the two policies are identical after time t , we have, $C^\Pi(s) = C^{\tilde{\Pi}}(s)$, $s \geq t$. And also cost rates are identical before time $P_{2,1}$ ($C^\Pi(s) = C^{\tilde{\Pi}}(s)$, $s < P_{2,1}$). But, in the time between $P_{2,1}$ and t , clearly $C^\Pi(s) < C^{\tilde{\Pi}}(s)$, $P_{2,1} \leq s < t$. So we have

If $J_{2,1} = 2$ then,

$$V^{\tilde{\Pi}} - V^\Pi = \int_0^\infty C^{\tilde{\Pi}}(s) ds - \int_0^\infty C^\Pi(s) ds = \int_{P_{2,1}}^t (C^{\tilde{\Pi}}(s) - C^\Pi(s)) ds = \int_{P_{2,1}}^t h_2 ds > 0$$

If $J_{2,1} = 1$ then, since $h_1 \geq h_2$ we have

$$V^{\tilde{\Pi}} - V^\Pi = \int_0^\infty C^{\tilde{\Pi}}(s) ds - \int_0^\infty C^\Pi(s) ds = \int_{P_{2,1}}^t (h_1 - h_2) ds \geq 0$$

So the policy Π is better than the policy $\tilde{\Pi}$, but this is a contradiction since $\tilde{\Pi}$ was presumed optimal.

Case 2. ($J_{1,1} = 1, J_{2,1} = 1$): The analysis in this case is identical to Case 1 and is omitted.

Case 3. ($J_{1,1} = 1, J_{2,1} = 2$) and $J_{2,1}$ is available for processing at time 0 (i.e. $n_2(0) \geq 1$). The analysis in this case is identical to Case 1 and is therefore omitted.

Case 4. $J_{1,1} = 1, J_{2,1} = 2$, and $J_{2,1} = 2$ is not available at time 0. (i.e. $n_2(0) = 0$, and the job that server 1 is processing in queue 1 will be the first job that will be processed by server 2 in queue 2 under policy $\tilde{\Pi}$. The choice at time 0 for server 2 here was between idling and serving a job from queue 1 since jobs from queue 2 were not available). In this case, $I = P_{1,1}$ since Server 2 is idling to serve the job that Server 1 is going to finish processing. Now consider an alternative policy Π where the second server processes $J_{1,2}$ (which would have

been processed by server 1 after it finished $J_{1,1}$ under $\tilde{\Pi}$), and under this revised policy, the server 1 processes the job at queue 2 that will be available when it finishes processing the job at queue 1 (instead of handing it over to server 2 as in $\tilde{\Pi}$). Server 2 idles after finishing processing $J_{1,2}$. After time $t = P_{1,2} + I$, the two policies are identical (except that server 2 becomes server 1 and vice versa, but that does not change anything since the servers are identical). The system under policy $\tilde{\Pi}$, and under policy Π is shown in Figure 4.

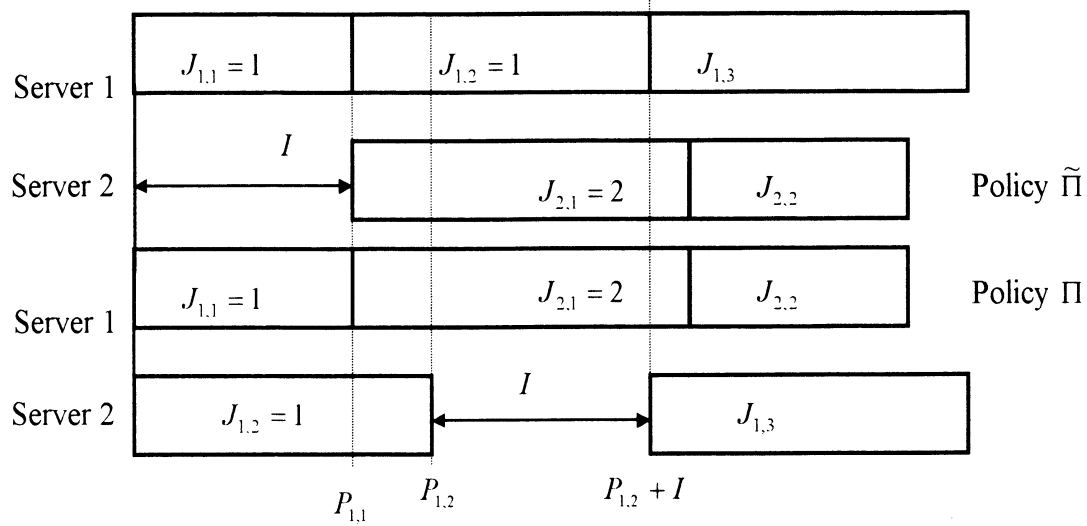


Figure 4. Interchange argument for Case 4

Then the difference of costs under the two policies will be

$$V^{\tilde{\Pi}} - V^{\Pi} = \int_0^{\infty} C^{\tilde{\Pi}}(s) ds - \int_0^{\infty} C^{\Pi}(s) ds = \int_{P_{1,2}}^{P_{1,2}+I} (h_1 - h_2) ds \geq 0 \text{ from } h_1 \geq h_2.$$

Thus, we have shown in each case that a policy $\tilde{\Pi}$ where one of the servers idles at a given time can not be optimal by shifting the idling period to the future. The same argument can be repeated so that the only time idling can occur is when one of the servers has no more jobs to process.

ii) $h_2 > h_1$

In this case, the argument is more complicated. We first note that it is obvious that when $n_1 + n_2 \leq 2$, (i.e. there are only two or fewer jobs left in the system), it can not be optimal to idle. So we only need to focus on cases with more than 2 jobs in the system. (It is also obvious that in states $(0, j)$, it can not be optimal to idle). Also, we can use the fact that the optimal policy is Markovian (which directly follows from finite state and action spaces), and the obvious fact that it can not be optimal for both servers to be idle at any point in time. We will prove the result by induction on all of the states to show that it can not be optimal to idle in any state. We divide the proof into three parts:

Part 1.

We first note that it is not optimal to idle in states $(0, j, P_2, R)$. We will show that it is not optimal to idle in states $(1, j, P_1, R)$. First, we note that it is easy to check that the non-idling policy is optimal in state, $(1, 1, P_1, R)$. To proceed the induction on j , we suppose that the claim above is true for $j \leq j_1, j_1 \geq 1$. In the next step, we show that the claim must also hold at $(1, j_1 + 1, P_1, R)$. Let $\tilde{\Pi}$ be a policy that idles in the state $(1, j_1 + 1, P_1, R)$ and let $\tilde{V}(1, j_1 + 1, P_1, R)$ be the value function associated with this policy. The policy Π assigns the server to queue 2 in that state and both policies follow the optimal policy in all other states. Note that because of the induction assumption and Theorem 1, this implies that there will be no idling in all other states under both policies and the servers will be assigned to queue 2 when possible. Thus, we have,

$$\begin{aligned}\tilde{V}(1, j_1 + 1, P_1, R) &= [h_1 + h_2(j_1 + 1)]/\mu_1 + \tilde{V}(0, j_1 + 2, P_2, R) = [h_1 + h_2(j_1 + 1)]/\mu_1 + V(0, j_1 + 2, P_2, R), \\ V(1, j_1 + 1, P_1, R) &= \frac{h_1 + h_2 j_1 + h_2}{\mu_1 + \mu_2} + \frac{\mu_1}{\mu_1 + \mu_2} V(0, j_1 + 2, P_2, R) + \frac{\mu_2}{\mu_1 + \mu_2} V(1, j_1, P_1, R).\end{aligned}$$

Then we have

$$\tilde{V}(1, j_1 + 1, P_1, R) - V(1, j_1 + 1, P_1, R) = \frac{\mu_2 [h_1 + h_2(j_1 + 1)]}{\mu_1(\mu_1 + \mu_2)} + \frac{\mu_2}{\mu_1 + \mu_2} [V(0, j_1 + 2, P_2, R) - V(1, j_1, P_1, R)] \quad (4.1.1)$$

Since a non-idling policy is optimal in state $(0, j_1 + 2, P_2, R)$ and in state $(1, j_1, P_1, R)$, we have

$$V(0, j_1 + 2, P_2, R) = h_2(j_1 + 2)/(2\mu_2) + V(0, j_1 + 1, P_2, R) \quad (4.1.2)$$

$$V(1, j_1, P_1, R) \leq \tilde{V}(1, j_1, P_1, R) = \frac{h_1 + j_1 h_2}{\mu_1} + V(0, j_1 + 1, P_2, R) = \frac{h_1 + j_1 h_2}{\mu_1} + V(0, j_1 + 1, P_2, R) \quad (4.1.3)$$

Substituting (4.1.2) and (4.1.3) into (4.1.1), we have

$$\begin{aligned}\tilde{V}(1, j_1 + 1, P_1, R) - V(1, j_1 + 1, P_1, R) &= \frac{\mu_2 [h_1 + h_2(j_1 + 1)]}{\mu_1(\mu_1 + \mu_2)} + \frac{\mu_2}{\mu_1 + \mu_2} [V(0, j_1 + 2, P_2, R) - V(1, j_1, P_1, R)] \\ &\geq \frac{\mu_2 [h_1 + h_2(j_1 + 1)]}{\mu_1(\mu_1 + \mu_2)} + \frac{\mu_2}{\mu_1 + \mu_2} \left[\frac{h_2(j_1 + 2)}{2\mu_2} + V(0, j_1 + 1, P_2, R) - \frac{h_1 + j_1 h_2}{\mu_1} - V(0, j_1 + 1, P_2, R) \right] \\ &= \frac{\mu_2 [h_1 + h_2(j_1 + 1)]}{\mu_1(\mu_1 + \mu_2)} - \frac{\mu_2}{\mu_1 + \mu_2} \frac{h_1 + h_2 j_1}{\mu_1} + \frac{\mu_2}{\mu_1 + \mu_2} \frac{h_2(j_1 + 2)}{2\mu_2} > 0.\end{aligned}$$

From the last inequality, we have shown that idling can not be optimal in state $(1, j_1 + 1, P_1, R)$ as well.

Part 2.

We now show that if a nonidling policy is optimal for all states (i_1, j_1, P_1, R) and $(i_1 - 1, j_1, P_2, R)$ for all states $((i_1 \leq i, \forall j_1) \text{ and } (i \geq 1, j_1 \leq j))$, then nonidling is also optimal in states $(i, j + 1, P_1, R)$ and $(i - 1, j + 1, P_2, R)$. We let $\tilde{\Pi}$ be a policy under which idles at the state $(i - 1, j + 1, P_2, R)$ and let $\tilde{V}(i - 1, j + 1, P_2, R)$ be the value function associated with this policy. Once again, V denotes the value function for a policy which does not idle but assign the server to queue 2. We then have

$$\tilde{V}(i - 1, j + 1, P_2, R) = \frac{h_1(i - 1) + h_2(j + 1)}{\mu_2} + \tilde{V}(i - 1, j, R, R) = \frac{h_1(i - 1) + h_2(j + 1)}{\mu_2} + V(i - 1, j, P_2, R).$$

The last equality directly follows from the induction hypothesis. (i.e., the optimal policy is a non-idling policy at $(i - 1, j, P_2, R), j \leq j_1$.) We compare this with the value function obtained by the non-idling policy Π , which is

$$V(i - 1, j + 1, P_2, R) = V(i - 1, j + 1, P_2, P_2) = [h_1(i - 1) + h_2(j + 1)] / (2\mu_2) + V(i - 1, j, P_2, R).$$

Then we have

$$\tilde{V}(i - 1, j + 1, P_2, R) - V(i - 1, j + 1, P_2, R) = \frac{h_1(i - 1) + h_2(j + 1)}{\mu_2} - \frac{h_1(i - 1) + h_2(j + 1)}{2\mu_2} > 0.$$

Thus, we have shown that idling can not be optimal in state $(i - 1, j + 1, P_2, R)$ as well. The argument for $(i, j + 1, P_1, R)$ is identical and is omitted.

Part 3.

Finally, to complete the induction argument, assuming that the result holds for $((i_1 \leq i, \forall j_1, P_1, R) \text{ and } (i_1 \leq i - 1, \forall j_1, P_2, R))$, we show that it also holds for $(i + 1, 0, P_1, R)$ and $(i, 1, P_2, R)$. The argument for this part is exactly the same as parts 1 and 2 and we omit the details. We have thus shown that idling can not be optimal in any state. ♦

5. Discussion and Extensions

In this section, we discuss the differences between the results obtained here and in Pandelis and Teneketzis, and then discuss some possible extensions that can be considered by future research. As described in the introduction, the paper by Pandelis and Teneketzis considers a similar problem. However, they assume general and not exponential processing times, and in their model jobs leaving queue 1 go to queue 2 with probability p and leave the system with probability $1 - p$. Denote the processing time at queue 1 by the

random variable X_1 , and the processing time at queue 2 by the random variable X_2 . Then their main result is stated in the following

Theorem 3 (Pandelis and Teneketzis, 1994) Suppose that $X_1 \leq_{st} X_2$, X_1 has nondecreasing likelihood ratio and $h_1 - ph_2 \geq h_2$, then the policy which allocates both servers to queue 1 (as long as there are jobs to be served in queue 1) is optimal.

We now note that when the processing times are exponential and $p = 1$, as in this paper, this result implies the following

Corollary 1 (Pandelis and Teneketzis, 1994) If $\mu_1 \geq \mu_2$, and $h_1 \geq 2h_2$ then the strategy which allocates both servers to queue 1 is optimal.

It is straightforward to note that our result, using the exponential assumption, provides a more complete characterization of when it is optimal to allocate both servers to queue 1, and provides the sufficient as well as necessary conditions under which allocating servers to queue 1 or queue 2 is optimal, and is therefore significantly different than the result by Pandelis and Teneketzis.

Having described the difference between the results in this paper and in previous literature, we now discuss possible extensions and conjectures. In section 3, we showed the conditions under which it is optimal to allocate both servers to queue 1 or queue 2. However, the results also implied that when $(1 + \mu_2 / (\mu_1 + \mu_2))h_2 < h_1 < (1 + \mu_2 / \mu_1)h_2$, the optimal policy does not necessarily allocate both servers to one of the queues. An interesting question is whether allocating one server to queue 1 and one server to queue 2 under these conditions (at least until one of the queues gets exhausted after which both servers would serve the other queue) is a policy that would perform well. In general, our numerical studies indicate that in these middle cases, the optimal allocation policy actually depends on how many jobs are in queue 1 and queue 2. However, to get a sense of how well simpler policies would perform, we tested two simple heuristic policies in the situations where $(1 + \mu_2 / (\mu_1 + \mu_2))h_2 < h_1 < (1 + \mu_2 / \mu_1)h_2$. The first heuristic (which we denote by $H(P_1, P_2)$) allocates one server to each queue so long as both queues have jobs to serve. In cases, where one of the queues has zero jobs, then both servers serve the other queue. We compare this policy against the policy of allocating both servers to queue 2 so long as queue 2 has jobs and only serving queue 1 when one of the servers runs out of jobs to serve at queue 2. (We denote this policy by $H(P_2, P_2)$).

We tested these two heuristics against the optimal policy. Since we know the exact policy except when $(1 + \mu_2 / (\mu_1 + \mu_2))h_2 < h_1 < (1 + \mu_2 / \mu_1)h_2$, we are only interested in values of h_1 which fall in this interval for given values of μ_1, μ_2, h_2 . Therefore, for each combination of μ_1, μ_2, h_2 , we chose values of h_1 which were 5%, 30%, 50%, 70% and 95% of the spread between $(1 + \mu_2 / \mu_1)h_2$ and $(1 + \mu_2 / (\mu_1 + \mu_2))h_2$, plus $(1 + \mu_2 / (\mu_1 + \mu_2))h_2$. For any given choice of h_1 , we picked 16 values of (n_1, n_2) , $n_1, n_2 \in \{3, 15, 3, 60\}$ to calculate the average sub-optimality of each heuristic.

Case 1. $\mu_1 = 1, \mu_2 = 2, h_2 = 1$

h_1	Avg. % Suboptimality (Max) of $H(P_1, P_2)$	Avg. % Suboptimality (Max) of $H(P_2, P_2)$
1.73333 (5%)	10.565 (23.934)	0.004 (0.040)
2.06667 (30%)	6.859 (15.966)	0.021 (0.211)
2.33333 (50%)	4.443 (10.590)	0.033 (0.320)
2.6 (70%)	2.435 (5.920)	0.092 (0.895)
2.93333 (95%)	0.456 (1.711)	0.314 (3.079)

Case 2. $\mu_1 = 2, \mu_2 = 1, h_2 = 1$

h_1	Avg. Suboptimality (Max) of $H(P_1, P_2)$ (%)	Avg. Suboptimality (Max) of $H(P_2, P_2)$ (%)
1.341667 (5%)	4.758 (6.765)	0.002 (0.018)
1.38333 (30%)	3.418 (4.862)	0.010 (0.107)
1.41666 (50%)	2.388 (3.406)	0.016 (0.176)
1.45 (70%)	1.396 (2.004)	0.025 (0.266)
1.49166 (95%)	0.221 (0.326)	0.054 (0.576)

Case 3. $\mu_1 = 1, \mu_2 = 2, h_2 = 1$

h_1	Avg. Suboptimality (Max) of $H(P_1, P_2)$ (%)	Avg. Suboptimality (Max) of $H(P_2, P_2)$ (%)
1.525 (5%)	8.369 (13.913)	0.003 (0.032)
1.65 (30%)	5.797 (9.704)	0.017 (0.183)
1.75 (50%)	3.935 (6.646)	0.028 (0.293)
1.85 (70%)	2.242 (3.828)	0.057 (0.594)
1.975 (95%)	0.357 (0.606)	0.143 (1.505)

Table 1: Results for performance of heuristics when $(1 + \mu_2 / (\mu_1 + \mu_2))h_2 < h_1 < (1 + \mu_2 / \mu_1)h_2$

In Table 1, we present the results comparing heuristics $H(P_1, P_2)$, $H(P_2, P_2)$ to the optimal solution. For each value of μ_1, μ_2, h_2 and h_1 , we give the average percentage suboptimality of each heuristic over the 16 cases considered. (We also provide the maximum suboptimality over the 16 problems in parentheses). We note

that it is very interesting that in these cases the heuristic $H(P_2, P_2)$, (i.e. exhaustive service at queue 2 when possible) performs much better than the heuristic $H(P_1, P_2)$, and is in fact very close to optimal. We note that despite the fact that in the region of values considered in this experiment, h_1 is always greater than h_2 , it is still much better to allocate both servers to queue 2 rather than queue 1. The average suboptimality of $H(P_2, P_2)$ over all the cases considered was 0.05% and a maximum suboptimality was 0.31% versus an average of 3.84% and a maximum of 10.57% of $H(P_1, P_2)$. Especially when h_1 is not significantly greater than $(1 + \mu_2 / \mu_1)h_2$, the performance of heuristic $H(P_1, P_2)$ is very close to the optimal policy, as one would expect.

Therefore, we can conclude that in general the simple policy of allocating both servers to queue 2 exhaustively whenever $h_1 < (1 + \mu_2 / \mu_1)h_2$ and allocating both servers to queue 1 exhaustively otherwise will work extremely well. Notice that one way to rewrite this inequality is to rewrite it as $\mu_2 h_2 > \mu_1 (h_1 - h_2)$. One way to interpret the left side is that this is the rate with which rewards are earned (or costs are reduced) when a server is working in queue 2, similarly the right side is the rate with which costs are reduced (so long as $h_1 > h_2$) when the server is working in queue 1. We would like to note that as Theorem 1 indicates this simple intuition does **not** lead to this policy being optimal in this whole region. However, as Table 1 indicates, the policy performs extremely well.

A second interesting question is whether Theorem 1 remains true if arrivals to queue 1 are introduced and the objective changed to minimizing that of the average cost per unit time. We conjecture that this result carries over to the case with dynamic Poisson arrivals. However, we have not to date been able to prove this result. Our extensive numerical tests yielded no cases where Theorem 1 did not hold in the dynamic case with the average cost minimization objective. In Tables 2 and 3, we provide some of the empirical evidence that we have observed. Table 2 includes the cases where we would expect exhaustive allocation of servers to queue 2 would be optimal since we have $h_1 = \alpha(1 + \mu_2 / (\mu_1 + \mu_2))h_2$, where $\alpha = 1, 0.75$ and 0.5 , and therefore Theorem 1 (if it held under these assumptions) would indicate optimality of this policy. (Note that λ denotes the arrival rate of jobs to queue 1. Similarly, in Table 3, we would expect optimality of exhaustive allocation of servers to queue 1 (since we have $h_1 = \alpha(1 + \mu_2 / \mu_2)h_2$, $\alpha = 1, 1.5$, and 2) and in fact this is the case.

The final column in Tables 2 and 3 illustrates the benefits of having flexible servers rather than dedicated servers. We note that even when a policy as in Theorem 1 which requires the servers to serve queue 1 or 2

exhaustively is implemented, the servers still need to be flexible (or cross trained) since when one queue is exhausted, the server switches to the other queue. Compare this to the case where each server or worker is trained for only one task, which we call the fixed tandem policy. In this case, the server only serves the queue that it is assigned to and idles if it has no jobs to work on. The results in Tables 2 and 3 indicate the benefits of a flexible work force as the fixed tandem policy results in costs which are on average 50% higher.

λ	μ_1	μ_2	h_2	$(1 + \frac{\mu_2}{\mu_1 + \mu_2})h_2$	h_1	Optimal policy	Optimal cost per unit time	Fixed tandem (suboptimality in %)
.45	1	2	1	5/3	5/3	Exhaustive at 2.	1.091	1.654 (51.632)
.45	1	2	1	5/3	5/4	Exhaustive at 2.	0.875	1.313 (50.090)
.45	1	2	1	5/3	5/6	Exhaustive at 2.	0.659	0.972 (47.627)
.45	2	1	1	4/3	4/3	Exhaustive at 2.	0.843	1.205 (42.964)
.45	2	1	1	4/3	1	Exhaustive at 2.	0.745	1.109 (48.735)
.45	2	1	1	4/3	2/3	Exhaustive at 2.	0.647	1.012 (56.380)
.45	1	1	1	3/2	3/2	Exhaustive at 2.	1.388	2.045 (47.336)
.45	1	1	1	3/2	9/8	Exhaustive at 2.	1.154	1.739 (50.607)
.45	1	1	1	3/2	3/4	Exhaustive at 2.	0.919	1.432 (55.718)

Table 2. Results for dynamic cases when $h_1 \leq (1 + \frac{\mu_2}{\mu_1 + \mu_2})h_2$.

λ	μ_1	μ_2	h_2	$(1 + \mu_2 / \mu_1)h_2$	h_1	Optimal policy	Optimal cost per unit time	Fixed tandem (suboptimality in %)
.45	1	2	1	3	3	Exhaustive at 1.	1.772	2.745 (54.906)
.45	1	2	1	3	4.5	Exhaustive at 1.	2.505	3.972 (58.584)
.45	1	2	1	3	6.0	Exhaustive at 1.	3.239	4.381 (60.525)
.45	2	1	1	1.5	1.5	Exhaustive at 1.	0.890	1.254 (40.896)
.45	2	1	1	1.5	2.25	Exhaustive at 1.	1.084	1.471 (35.707)
.45	2	1	1	1.5	3.0	Exhaustive at 1.	1.280	1.689 (31.984)
.45	1	1	1	2	2	Exhaustive at 1.	1.691	2.455 (45.157)
.45	1	1	1	2	3	Exhaustive at 1.	2.223	3.273 (47.210)
.45	1	1	1	2	4	Exhaustive at 1.	2.754	4.091 (48.519)

Table 3. Results for dynamic cases when $h_1 \geq (1 + \mu_2 / \mu_1)h_2$

Further research is needed to prove the conjectures presented in this section and to develop results for tandem queues with a larger number of jobs and servers.

References:

- Agrawal, R., Hedge, M., Teneketzis, D., (1990), "Multi-armed bandit problems with multiple plays and switching costs," *Stochastics and Stochastic Reports* **29**, 437-359.
- Bartholdi, J.J., Bunimovich, L.A., and Eistentein, D., (1995), "Dynamics of 2 and 3-worker bucket brigade production lines," Technical Report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0160.
- Chang, C., Nelson, R. and Pinedo, M., (1991), "Scheduling two classes of exponential jobs on parallel processors: Structural results and worst-case analysis," *Advances in Applied Probability*, **23**, 925-944.
- Duenyas, I., Gupta, D. and Olsen, T. (1995), "Control of a single server tandem queueing system with setups," 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," (1993), *IEEE Transactions on Automatic Control*, **38**, 1296-1299.
- Gittins, J.C., (1989) *Multi-armed Bandit Allocation Indices*, Wiley, New York.
- Hajek, B., (1984), "Optimal control of two interacting service stations," *IEEE Transactions on Automatic Control*, **AC29**, no. 6, 491-499.
- Hofri, M. and Ross, K., (1987), "On the optimal control of two queues with server setup times and its analysis," *SIAM Journal of Computing*, **16**, 399-420.
- Iravani, S.M.R., Posner, M.J.M., and J.A. Buzacott, (1995), "Two-stage tandem queue attended by a moving server with holding and switching costs; static and semi-dynamic policy," Technical Report. 95-18, University of Toronto, Department of Industrial engineering, Toronto, Canada.
- Iravani, S.M.R., Posner, M.J.M., and J.A. Buzacott, (1996), "An N-stage tandem queueing system attended by a moving server with holding and switching costs," Technical Report., 96-09, University of Toronto, Department of Industrial Engineering, Toronto, Canada.
- Kampke, T., (1987), "On the optimality of static priority policies in stochastic scheduling on parallel machines," *Journal of Applied Probability*, **24**, 430-448.
- Kampke, T., (1989), "Optimal scheduling of jobs with exponential service times on identical parallel processors," *Operations Research*, **37**, 126-133.
- McClain, J.O., L.J. Thomas, and K. Schultz, "Management of Worksharing Systems," Technical Report, S.C. Johnson Graduate School of Management, Cornell University, Ithaca, NY, 1997.
- Pandelis, D. and Teneketzis, D., (1994a), "Optimal multiserver stochastic scheduling of two interconnected priority queues," *Advances in Applied Probability*, **26**, 258-279.

- Pandelis, D. and Teneketzis, D., (1994b), "Optimal stochastic dynamic scheduling in multi-class queues with tardiness and/or earliness penalties," *Probability in the Engineering and Informational Sciences*, **8**, 491-509.
- Pinedo, M., (1995), *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, New Jersey.
- Rosberg, Z., Varaiya, P, and Walrand, J., (1982), "Optimal control of service in tandem queues," *IEEE Transactions in Automatic Control*, **AC-27**, 600-609.
- Ross, S. (1983) *Introduction to Stochastic Dynamic Programming*, Academic Press, New York.
- Stidham, S. and Weber, R., (1993), "A survey of Markov decision models for control of networks of queues," *Queueing Systems: Theory and Applications*, **13**, 291-314.
- Van Oyen, M and Teneketzis, D., (1992) Optimal scheduling of a finite capacity shuttle under delayed information, *Probability in the Engineering and Informational Sciences*, **6**, 29-61.
- Van Oyen, M., and Tenektzis, D., (1994), "Optimal Stochastic Scheduling of Forest Networks with Switching Penalties," *Advances in Applied Probability*, **26**, 474-497.
- Van Oyen, M. and Teneketzis, D., (1996), "Optimal batch services for a polling system under partial information," *ZOR: Methods and Models of Operations Research*, **44**, 401--419.
- Weber, R. and Stidham, S. (1987), "Optimal control of service rates in networks of queues," *Advances in Applied Probability*, **19**, 202-217.