A Robust Policy for Serial Agile Production Systems

S.M.R. Iravani,¹ J.A. Buzacott,² M.J.M. Posner³

¹ Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208
² Schulich School of Business, York University, Toronto, Ontario, M3J 1P3, Canada
³ Department of Industrial Engineering and Management, Jerusalem College of Technology, Jerusalem 91160, Israel

Received 28 September 2001; revised 13 August 2004; accepted 18 August 2004
DOI 10.1002/nav.20051
Published online 12 November 2004 in Wiley InterScience (www.interscience.wiley.com).

Abstract: One of the major problems in modeling production systems is how to treat the job arrival process. Restrictive assumptions such as Markovian arrivals do not represent real world systems, especially if the arrival process is generated by job departures from upstream workstations. Under these circumstances, cost-effective policies that are robust with respect to the nature of the arrival process become of interest. In this paper, we focus on minimizing the expected total holding and setup costs in a two-stage produce-to-order production system operated by a cross-trained worker. We will show that if setup times are insignificant in comparison with processing times, then near-optimal policies can be generated with very robust performances with respect to the arrival process. We also present conditions under which these near-optimal policies can be obtained by using only the arrival and service rates. © 2004 Wiley Periodicals, Inc. Naval Research Logistics 52: 58 –73, 2005.

Keywords: tandem queues; moving servers; single-machine stochastic scheduling; setup costs

1. INTRODUCTION

Agile manufacturing, a natural successor to lean manufacturing, emphasizes highly customized products and quick response to customer demands. Unlike lean manufacturing, which emphasizes mass production in large-scale corporate structures, agile manufacturing is better suited to smaller scale and modular flexible production facilities. In this environment, multifunctional machinery and cross-trained (agile) workforce will become increasingly essential for enabling agile manufacturers to produce highly customized products and to respond rapidly to changes in customer demand. An agile or cross-trained workforce consists of multiskilled workers that can be dynamically allocated to different work stations in the system. Sekine [23] describes how several companies in Japan redesigned their production systems and used cross-trained workers to increase their productivity while reducing the number of workers, sometimes even by half.

In this paper, we consider a two-station serial agile produce-to-order system in which a fully cross-trained worker is in charge of both stations. Jobs arrive according to a general arrival process and each arriving job requires processing by the worker in station 1 and then in station 2. Holding costs are charged for each unit of Work-In-Process (WIP) in stations 1 and 2, and also a switching (setup) cost is incurred whenever the worker switches from one station to the other. We assume that switching (setup) times are negligible relative to job processing times, and we therefore consider these setup times to be zero in our model. However, a switching (setup) cost is incurred whenever the worker switches from one station to the other.

Our simple model in this paper also presents systems which employ multifunctional machines. An example is a workshop where all the parts being produced require the same sequence of two operations on the workshop’s only Computer Numerical Control (CNC) machine. In this case, the cross-trained worker is actually the machine and each operation is analogous to a station. The switching costs in this version of the problem are the machine setup costs and the holding cost is charged for the WIP at a machine.

Correspondence to: S.M.R. Iravani (s-iravani@northwestern.edu)

© 2004 Wiley Periodicals, Inc.
Setup time reduction was one of the key components of the success of Toyota’s production system. In Toyota’s system, setup operations are divided into offline and online operations. Offline setup operations can be performed while the system still processes jobs; however, in order to perform online setup operations, production must be stopped. The main goal in this approach is to minimize the online operation times as much as possible, so the system is stopped only for a short period of time for setup. The best example of this approach is SMED or Single-Minute Exchange of Dies. Our model in this paper represents situations where the server’s switching times (online setup times) have been reduced to insignificance in comparison with processing times; however, we consider switching costs to represent costs associated with offline setup operations.

The optimal schedule of the cross-trained worker with setup times and costs under a homogeneous Poisson job arrival process has been studied in Iravani, Posner, and Buzacott [13], and it was shown to have a complex structure. However, when the arrival process is not homogeneous Poisson, the optimization or performance analysis of these systems has not been studied in the literature. This is not unusual, since under a non-Markovian arrival process the analysis becomes very complex. In this paper, we will show that if setup times are negligible compared to the job processing times, then cost-effective policies can be obtained without prior knowledge of job arrival process and rate.

In the literature, serial production systems with cross-trained workers appear in different contexts such work sharing (McClain et al. [19–21,26], Bischack [7]), Bucket Brigade (Bartholdi and Eisenstein [5] and Bartholdi, Eisenstein, and Foley [6]), workforce agility (Van Oyen, Gel, and Hopp [24], Gel, Hopp, and Van Oyen [10]), tandem queues attended by moving servers with no arrivals (Farrar [9], Hopp and Van Oyen [12]), and Iravani, Posner, and Buzacott [13]. For a review of queueing models for systems with a general arrival process is not Poisson, especially if the arrival distribution is not Poisson, especially if the queueing system is a subsystem of a manufacturing environment. In a manufacturing setting, interruptions such as machine failures, blocking, or lack of raw material in upstream work stations make it almost impossible to characterize the arrival process at a work station. On the other hand, even if the arrival process is characterized, analyzing queueing models for systems with a general arrival process is an easy task. Thus, studies on these systems have been limited to the simulation analysis of some special cases.

Another way of dealing with these complicated systems is to look for policies which perform reasonably well and are robust with respect to the arrival process. This will be the focus of this paper. In Section 2 we first introduce a class of policies, namely, Limited Policies, and in Sections 3 and 4 we show how the optimal limited policies can be obtained or very accurately approximated without having any information regarding the arrival process and service time distributions. Then, in Section 5 we revise the structure of the limited policies to construct a new policy, which we call Limited Policy with Startup Batch (LPSB). We examine how close the optimal LPSB is to the global optimal policy and describe how it can be approximated by using only the job arrival and service rates. Section 6 is devoted to a numerical study we performed to demonstrate the robustness and cost effectiveness of the optimal limited LPSB policies. Finally, Section 7 indicates how our results can be used in production systems with multiple stages and multiple workers.

2. A TWO-STAGE TANDEM QUEUE WITH LIMITED POLICY

In this section we model our two-station production system with a cross-trained worker as a two-stage tandem
queue attended by a moving server. Consider a two-stage tandem queue in which only one server is assigned the responsibility of serving customers in both queues, and the buffers at both stages are of infinite size (see Fig. 1). Customers arrive at stage 1 according to a general arrival process with rate $\lambda$ and enter stage 2 immediately after their service completion at stage 1. The service times at stage $i$ are independent and identically distributed with mean $E[S_i]$. A holding cost $h_i$ per unit time is defined for each customer during its wait and service in stage $i$ ($i = 1, 2$). We assume that $h_2 \geq h_1$. This is certainly a reasonable assumption in manufacturing environments, since the value of a job increases after each completion of an operation in a stage. Also, a switching cost $K_i$ is incurred whenever the server switches to stage $i$ ($i = 1, 2$).

We assume that the server applies a limited policy in stage 1 and an exhaustive policy in stage 2. According to a limited policy, when the server switches to stage 1, s/he continues serving and does not switch to stage 2 until stage 1 becomes either empty or at most a predetermined number of customers, $M$, are served. If the server finds stage 1 empty when s/he switches to that stage, s/he waits there until the next customer arrives. When the server switches to stage 2, s/he serves all customers waiting there and then switches back to stage 1 (exhaustive policy in stage 2). Defining a cycle as the time elapsed between two consecutive switches to stage 1, the number of customers served in each cycle under a limited policy is a discrete random variable between 1 and $M$. (Note that although our model assumes that the arrival rate is known, we will later show that the optimal limited policy can be approximated independent of the arrival rate.)

We must also emphasize that nonidling policies, such as our limited policy, in which the server never idles as long as there are customers waiting in the system (stage 1 or 2), become more attractive when server idling is expensive. For example, in systems with multifunctional expensive CNC machines, the low utilization due to machine idling is a waste of investment. The limited policy is an attractive policy under these circumstances since it not only controls the average holding and switching cost through the control limit $M$, but it also minimizes the idling cost since it never idles as long as there is work to be done in the system.

3. **OPTIMAL LIMITED POLICY**

It can be concluded from Johri and Katehakis [14] that if switching costs in our model are zero, then the optimal policy which minimizes the average holding costs per unit time is a Sequential Policy (i.e., a limited policy with $M = 1$). In other words, when applying a sequential policy, the server starts serving each customer at stage 2 immediately after her service completion in stage 1. However, when switching costs are not zero, the sequential policy results in a high average switching cost per unit time. In fact, under a sequential policy only one customer is served in each cycle; as a result, the switching cost $K_1 + K_2$ is charged for each served customer.

Clearly, policies which on average serve more customers in a cycle result in a lower average switching cost per unit time. On the other hand, a larger number of served customers in a cycle results in more customers waiting in the second stage while the server is busy in the first stage. This increases the average holding cost per unit time, since $h_2 \geq h_1$. The limited policy applies a control limit $M$ on the maximum number of customers served in a cycle in order to control the trade off between the average holding costs and the average switching costs. To find the optimal limit $M$ which minimizes the expected total holding and switching cost per unit time, $E[TC(M)]$, we must solve the following optimization problem:

$$
\text{Min } E[TC(M)] = h_1L_1^{(M)} + h_2L_2^{(M)} + (K_1 + K_2) \frac{1}{E[C^{(M)}]} 
$$

subject to:

$$
M \in \{1, 2, 3, \ldots \},
$$

where $L_1^{(M)}$ and $L_2^{(M)}$ are the average number of customers in steady-state in stages 1 and 2, respectively, and $E[C^{(M)}]$ is the average cycle time in steady-state under a limited policy with limit $M$. Even under the assumption of a homogeneous Poisson arrival process and deterministic service times, $L_1^{(M)}$, $L_2^{(M)}$, and $E[C^{(M)}]$ do not have a simple form that allows one to obtain the optimal limited policy through analytical methods (see Katayama [15]). Thus,
finding the optimal $M$ for systems under a limited policy may require an extensive search,\(^1\) where each value of $M$ must be examined in the objective function (1).

This problem becomes more tedious when the arrival process is not homogeneous Poisson. In that case, the expressions derived by Katayama [15] for the average number of customers in each stage and the average cycle time are not valid. Therefore, in regard to problem (1) with its messy objective function tightly dependent on the arrival process and service time distributions, two interesting questions arise: (i) are there any circumstances under which the optimal limited policy can be obtained regardless of the arrival process? If not, (ii) is it possible to at least approximate the optimal limit $M^\ast$ without using any information regarding the arrival process?

We address the above two questions as follows: In Proposition 1 we will show that when service times are deterministic, under some condition, the optimal limited policy can be obtained without using any information about the arrival process.\(^2\) Then in Section 4 we develop a heuristic approach for systems with stochastic processing times that yields a good approximation for the limit of the optimal limited policy independent of the arrival process. Before we present Proposition 1, we need to define $N(M)$ and $K(M)$ as follows:

\[
\begin{align*}
N(M) &= \text{number of customers served in a cycle under a limited policy with limit } M, \\
K(M) &= \text{number of customers present in stage 1 at the beginning of a cycle under a limited policy with limit } M.
\end{align*}
\]

**LEMMA 1:** Under a limited policy with limit $M$ in a system with deterministic service times, consider a given cycle of length $T_{cy}$ in which exactly $N$ ($N \leq M$) customers are served, and let $E[T_{cy}|N(M) = N]$ denote the expected total cost incurred during $T_{cy}$. If $E[T_{cy}|N(M) = N]$ denotes the expected total cost incurred during $T_{cy}$ when these $N$ customers are served in two consecutive cycles serving $n_1$ and $n_2$ customers ($n_1 + n_2 = N$), respectively, then

\[
E[T_{cy}|N(M) = N] = E[T_{cy}^{n_1,n_2}|N(M) = N] = n_1n_2\mathcal{H}_1 + \mathcal{H}_2, \quad (2)
\]

\(^1\)It has not been analytically proven that objective function (1) is convex. However, in our extensive numerical study we did not find an example in which (1) was not convex.

\(^2\)Note that deterministic service times represent situations in which variability in service times is negligible. For example, in manufacturing systems, this often happens when the job processing consists of simple and routine operations, or cases where job processing is performed automatically by an advanced multifunctional machine such as CNC.

where $\mathcal{H}_1 = (h_2 - h_1)S_2 + h_2S_1$, $\mathcal{H}_2 = K_1 + K_2$, and $S_1$ is the service time at stage $i$ ($i = 1, 2$).

**PROOF:** Suppose that the given cycle starts at time $t$. In other words, the server switches to stage 1 at time $t$ to start a cycle which ends up serving $N$ customers ($N = 2, 3, \ldots, M$). Without losing generality, we set $t = 0$, and assume that there are $k$ customers present in stage 1 at time $t = 0$. We consider two cases, namely $k < N$, and $k \geq N$.

When $K < N$, then customers $k + 1, k + 2, \ldots, N$ arrive during the cycle. Thus, the total holding cost in stage 1 associated with customers $1, 2, \ldots, k$ during the cycle is $h_1\sum_{i=1}^{k}iS_i$. On the other hand, the holding cost associated with customer $i$ ($i = k + 1, k + 2, \ldots, N$), who arrives at time $\tau_i > t = 0$ and waits in the queue of stage 1 for $W_q(i)$ units of time, is

\[
h_1E[W_q(i)|N(M) = N, K(M) = k] = h_1E[(i - 1)S_i - \tau_i + S_i|N(M) = N, K(M) = k] = h_1S_i - h_iE[\tau_i|N(M) = N, K(M) = k].
\]

The expected total cost associated with customers $k + 1, k + 2, \ldots, N$ in stage 1 is therefore

\[
h_1\sum_{i=k+1}^{N}iS_i - h_1\sum_{i=k+1}^{N}E[\tau_i|N(M) = N, K(M) = k], \quad (3)
\]

and the expected total cost of serving these $N$ customers in stages 1 and 2 during $T_{cy}$ is

\[
E[T_{cy}|N(M) = N, K(M) = k] = h_1\sum_{i=1}^{N}iS_i + h_2\sum_{i=k+1}^{N}iS_i + h_2\sum_{i=k+1}^{N}E[\tau_i|N(M) = N, K(M) = k] + E[T_{cy}^{(N)}] + \mathcal{H}_1 + \mathcal{H}_2, \quad (4)
\]

where $E[T_{cy}^{(N)}]$ corresponds to the expected total costs associated with all customers in excess of the $N$ processed ones.

Now, suppose that the same $N$ customers are served in two consecutive cycles serving $n_1$ and $n_2$ customers, where
$n_1 + n_2 = N$. Note that $n_1$ and $n_2$ can be any two positive integer numbers that satisfy $n_1 + n_2 = N$. Also, observe that any cycle generated under a limited policy can be broken into two consecutive cycles in which the server never idles. The reason is that under a limited policy, in a cycle that serves $N$ customers, the arrival epoch of the $(i + 1)$th served customer in that cycle is always before the service completion epoch of the $i$th served customer at stage 1, $i = 1, 2, \ldots, N - 1$. Thus, after serving any number $n_1$ customers in stages 1 and 2 (in the first cycle), the server can serve the remaining $N - n_1$ customers in the second cycle without waiting (idling) for any of the $(N - n_1)$ customers to arrive.

Using the same argument as in (4), it can be shown that

$$E[T^c_{cy}|N^{(M)} = N, K^{(M)} = k] = h_1 \frac{n_1(n_1 + 1)}{2} S_1 + h_2 \frac{n_1(n_1 + 1)}{2} S_2 + h_1 n_1 n_2[S_1 + S_2]$$

$$+ h_1 \frac{n_2(n_2 + 1)}{2} S_1 + h_2 \frac{n_2(n_2 - 1)}{2} S_1 + h_2 \frac{n_2(n_2 + 1)}{2} S_2$$

$$- h_1 \sum_{i=k+1}^{N} E[\tau_i|N^{(M)} = N, K^{(M)} = k] + E[T^c_{cy}|N^{(n_1, n_2)}] + 2 \mathcal{K}_{12}. \quad (5)$$

The term $h_1 n_1 n_2(S_1 + S_2)$ in (5) is the cost of holding $n_2$ customers in stage 1 during the service of $n_1$ customers in stages 1 and 2. Note that $E[\tau_i|N^{(M)} = N, K^{(M)} = k]$ also appears in (5), because these $N$ customers who are served in two consecutive cycles are the same $N$ customers who were served in one cycle [see (4)].

Since serving $N$ customers in one or two consecutive cycles takes the same amount of time (i.e., $T^c_{cy}$), then $E[T^c_{cy}|N^{(n_1, n_2)}] = E[T^c_{cy}|N^{(M)}]$. Subtracting (5) from (4) after some algebra for $k < N$, we get

$$E[T^c_{cy}|N^{(M)} = N, K^{(M)} = k] - E[T^c_{cy}|N^{(n_1, n_2)}|N^{(M)} = N, K^{(M)} = k] = n_1 n_2(2h_2 - h_1)S_2 + h_2 S_2 - (K_1 + K_2) = n_1 n_2 \mathcal{K}_{12} - \mathcal{K}_{12}. \quad (6)$$

For cases where $k \geq N$, the terms in (3) are zero, and therefore it can be easily shown that (6) also holds for $k \geq N$. Finally, since (6) is the same for all $k$ [i.e., (6) is independent of the number of customers present at the beginning of a cycle], therefore we have

$$E[T^c_{cy}|N^{(M)} = N] - E[T^c_{cy}|N^{(n_1, n_2)}|N^{(M)} = N] = n_1 n_2 \mathcal{K}_{12} - \mathcal{K}_{12}. \quad \Box$$

**COROLLARY 1:** In Lemma 1, serving $n_1$ and $n_2$ customers in two consecutive cycles and serving $n_2$ and $n_1$ customers in two consecutive cycles have the same expected costs, since according to (2), we have $E[T^c_{cy}(n_1, n_2)] = E[T^c_{cy}(n_2, n_1)] = E[T^c_{cy}(N)] - n_1 n_2 \mathcal{K}_{12} + \mathcal{K}_{12}$. 

**COROLLARY 2:** Define

$$R_{12} = \frac{\mathcal{K}_{12}}{\mathcal{K}_{12}}. \quad (7)$$

Based on (2), it is clear that if $R_{12} > n_1 n_2$ ($R_{12} < n_1 n_2$), then breaking a cycle which serves $N$ customers into two consecutive cycles which serve $n_1$ and $n_2$ customers increases (decreases) the total cost during that cycle.

**LEMMA 2:** In systems with deterministic service times, if the sequential policy is optimal ($M^*_1 = 1$), then $R_{12} \leq 1$.

**PROOF:** To prove this lemma, we use contradiction. Suppose that the limited policy with $M^*_1 = 1$ is optimal, but we have $R_{12} > 1$. Now, if the sample path of a limited policy with limit $M = 2$ is compared with the sample path of a limited policy with limit $M = 1$, it is clear that the sample path of the latter follows the former except for cycles in which two customers are served. In those cases, the limited policy with $M = 1$ breaks those cycles into two cycles each serving one customer. However, since $R_{12} > 1$, then according to Corollary 2 this increases the expected total cost. Therefore, limited policy with $M = 2$ will have a lower expected total cost than the limited policy with $M^*_1 = 1$, and the limited policy with limit $M = 1$ cannot be optimal, a contradiction! \qquad \Box

In Lemma 3 we will show that, regardless of interarrival and service time distributions, limited policies with different limits generate the same busy and idle periods. In fact, this is true for all nonidling policies where the server never idles in any stage as long as there is at least one customer in the system (stage 1 or 2).

**LEMMA 3:** All classes of nonidling policies generate the same busy and idle periods in a two-stage tandem queue attended by a moving server.

**PROOF:** A two-stage tandem queueing system attended by a moving server can be viewed as a single stage queueing system in which the service of a customer has two parts, which take random times $S_1$ and $S_2$, respectively. In this system, the server is allowed to start the service of another customer after finishing part one service of a customer. From this perspective, it becomes clear that different non-idling policies only dictate different orders of serving cus-
tomers and do not prolong or shorten the length of a busy period.

We now introduce a condition under which the optimal limited policy in systems with deterministic service times can be obtained without using any information regarding the arrival process. The proof of Proposition 1 is rather long and therefore is presented in the Appendix.

**PROPOSITION 1:** In a two-stage tandem queue attended by a moving server with holding costs $h_2 \geq h_1$, switching costs $K_1$ and $K_2$, and deterministic service times, the optimal limited policy has parameter $M^* = 2$ if

$$1 \leq \mathcal{R}_{12} \leq 2. \quad (8)$$

Proposition 1 introduces conditions under which, in systems with deterministic service times, the optimal limited policy [i.e., the solution to optimization problem (1)] is independent of the arrival process. The question now is whether in systems that violate condition (8), or in systems with stochastic processing times, the optimal limited policy can be obtained (or approximated) independent of the arrival process. To further investigate this possible independence, we will now look at the behavior of the optimal limited policy in systems with stochastic processing times. First, we start by analyzing the system under heavy traffic (i.e., systems under high utilization).

### 4. AN APPROXIMATION METHOD FOR THE OPTIMAL LIMITED POLICY

In order to further explore the behavior of the optimal limited policy, we carried out a simulation study examining the behavior of the objective function of the optimization problem (1). Several problems with different arrival processes and service time distributions were considered, and for each we explored the effects of changes in the value of $M$ of the limited policy as well as changes in the arrival rate. Figure 2 shows a typical behavior of this function.

Figure 2 was obtained through simulation of a two-stage tandem queue with normally distributed interarrival times with mean $1/\lambda$ and variance $(0.3/\lambda)^2$, hyperexponential service times in stage 1 with parameters $(p = 0.7, \mu_1 = 14, \mu_2 = 10)$ and mean $E[S_1] = 0.08$, and deterministic service time in stage 2 which takes 0.05 time units. The switching costs were $K_1 + K_2 = 10$, and the holding costs were $h_2 = 20$, and $h_1 = 16$. We increased the arrival rate from 1 to 7 which created traffic intensities from $\rho = 0.13$ to $\rho = 0.91$; however, in all cases the optimal limit remained at $M^* = 3$.

As Figure 2 shows, the bottom of curve $E[TC(M)]$ becomes flatter as $\lambda$, and therefore traffic intensity $\rho = \lambda(E[S_1] + E[S_2])$ decreases. This implies that, as $\rho$ decreases, the limited policy with limit $M + 1$ acts more like the limited policy with limit $M$. Thus, in systems with low utilization (i.e., light traffic, see Fig. 2 for $\lambda = 1$), all limited policies with limits $M \geq 2$ behave like the limited policy with limit $M = 1$. The reason for this behavior is that in light traffic, whenever the server switches to stage 1 and finishes the service of a customer, with a large probability,

---

3In our simulation program the negative random numbers generated for normal distributions are ignored. Since Normal distributions used in this paper have means at least three times larger than their standard deviations, the probability of having a negative generated number is insignificant (i.e., less than 0.001).
stage 1 becomes empty and the server must switch to stage 2 regardless of the limit $M$.

Under heavy traffic (when $\rho \to 1$), on the other hand, there are significant differences in expected total costs under different limited policies (see Fig. 2 for $\lambda = 7$). This is so because, in almost all cycles, when $\rho \to 1$, limited policy with limit $M$ serves exactly $M$ customers, while the limited policy with limit $M + 1$ serves $M + 1$ customers. The interesting point is that as $\rho$ approaches but does not equate to 1, the optimal limited policy can be accurately approximated independent of the arrival process. This is due to the fact that when $\rho \to 1$, the number of customers in stage 1 is very large. Therefore, the search for the optimal limited policy in these systems under a stochastic arrival process becomes equivalent to the search for the optimal limited policy in the same system with no arrivals but with a large number of customers in stage 1. This motivated us to develop a heuristic method that finds a good approximation for the optimal limit of the limited policy in systems with heavy traffic without using any information regarding the arrival process. We call our heuristic the Heavy-Traffic Heuristic or the H-Heuristic.

If we define

$$\mathcal{H}_1 = (h_2 - h_1)E[S_2] + h_2E[S_1] \quad \text{and} \quad \mathcal{R}_1 = \frac{\mathcal{H}_1}{h_2}$$

then our heuristic method suggests that integer $m^*$ which satisfies

$$\frac{m^*(m^* - 1)}{2} \leq \mathcal{R}_1 \leq \frac{m^*(m^* + 1)}{2}$$

(10)
can be used as a good approximation for the optimal limit of the limited policy for systems under heavy traffic.

To illustrate the basis of our heuristic method, consider a system in heavy traffic where $\rho$ is very close to 1. In that system there will almost always be a large number of customers waiting at stage 1. Our heuristic therefore assumes that in heavy traffic, there are always more than $m(m + 1)$ customers available at stage 1, i.e., it assumes $Pr(K^{(m)} \geq m(m + 1)) = 1$, and $Pr(K^{(m)} < m(m + 1)) = 0$. Hence, under limited policy $\omega_0$ with limit $m$ for $i = 1, 2$, we have

$$E[S_i] = E[S_i|K^{(m)} \geq m(m + 1)] Pr(K^{(m)} \geq m(m + 1)) + E[S_i|K^{(m)} < m(m + 1)]$$

(11)

where $Pr(K^{(m)} \geq m(m + 1))$ can be approximated as a good approximation for the optimal limit of the limited policy in systems under heavy traffic without using any information regarding the arrival process. We call our heuristic the Heavy-Traffic Heuristic or the H-Heuristic.

Let $E[TC_{\omega_0}^{(m + 1)}]$ be the expected total holding and switching costs associated with serving $m(m + 1)$ customers under policy $\omega_0$ during the time that the server serves these customers. Assuming that under heavy traffic, there are always more than $m(m + 1)$ customers available at stage 1 at the beginning of a cycle, policy $\omega_0$ will serve $m(m + 1)$ customers in $(m + 1)$ consecutive cycles. $E[TC_{\omega_0}^{(m + 1)}]$ can thus be approximated by

$$E[TC_{\omega_0}^{(m + 1)}] = (m + 1)\left(h_1 \frac{m(m + 1)}{2} E[S_1|K^{(m)} \geq m(m + 1)]ight)$$

$$+ h_2 \frac{m(m + 1)}{2} E[S_2|K^{(m)} \geq m(m + 1)]$$

$$+ (m + 1)(K_1 + K_2 + mh_1(mE[S|K^{(m)} \geq m(m + 1)])$$

$$\times \left(1 - \frac{m(m + 1)}{2} + E[TC_{\omega_0}^{(m + 1 + 1)}] + E[TC_{\omega_0}^{(m + 1)}]\right),$$

(12)

where $[S|K^{(m)} \geq m(m + 1)] = [S_1|K^{(m)} \geq m(m + 1)] + [S_2|K^{(m)} \geq m(m + 1)], E[TC_{\omega_0}^{(m + 1 + 1)}]$ is the expected total cost associated with all other customers [excluding the $m(m + 1)$ customers] under policy $\omega_0$, and $E[TC_{\omega_0}^{(m + 1)}]$ is the expected total cost associated with the $m(m + 1)$ customers up to the time that the service of the first of the $m(m + 1)$ customer starts. Considering (11) and $E[S] = E[S_1] + E[S_2]$, we will have

$$E[TC_{\omega_0}^{(m + 1)}] = (m + 1)\left(h_1 \frac{m(m + 1)}{2} E[S_1]ight)$$

$$+ h_2 \frac{m(m + 1)}{2} E[S_2] + (m + 1)$$

$$\times (K_1 + K_2 + mh_1(mE[S]) \left(1 - \frac{m(m + 1)}{2} + E[TC_{\omega_0}^{(m + 1 + 1)}] + E[TC_{\omega_0}^{(m + 1)}]\right).$$

(13)

Now consider limited policy $\omega_1$ with parameter $m + 1$. Under heavy traffic, this policy serves the same number of customers, $m(m + 1)$, but in $m$ consecutive cycles. In a way similar to the case for policy $\omega_0$, our heuristic method finds an approximation for $E[TC_{\omega_1}^{(m + 1)}]$, i.e., the expected total cost associated with serving the same $m(m + 1)$ customers under policy $\omega_1$:
$$E[TC^m_{\omega_0}] = m \left( h_1 \frac{(m+1)(m+2)}{2} E[S_1] + h_2 \frac{(m+1)(m+2)}{2} E[S_2] \right)$$

$$+ m(K_1 + K_2) + (m+1)h_1(m+1)E[S]\left( \frac{m(m-1)}{2} \right) + E[TC^m_{\omega_1}] + E[TC^m_{\omega_2}]_0. \quad (14)$$

Since $$E[TC^m_{\omega_0}] = E[TC^m_{\omega_1}]$$, and

$$E[TC^m_{\omega_0}] - E[TC^m_{\omega_1}] = E[TC^m_{\omega_2}]_a,$$

we get

$$E[TC^m_{\omega_0}] - E[TC^m_{\omega_1}] = \frac{m(m+1)}{2} [(h_2 - h_1)E[S_2] + h_2E[S_1]] - (K_1 + K_2).$$

If $$E[TC^m_{\omega_0}] - E[TC^m_{\omega_1}] \geq 0$$, or, in other words, if

$$\frac{K_1 + K_2}{(h_2 - h_1)E[S_2] + h_2E[S_1]} \leq \frac{m(m-1)}{2}, \quad (15)$$

then our H-Heuristic concludes that policy $$\omega_0$$ will be better than policy $$\omega_1$$. Note that using the same approach in comparing policies $$\omega_0$$ and $$\omega_{-1}$$ with parameter $$m = 1$$, our H-Heuristics concludes that if

$$\frac{K_1 + K_2}{(h_2 - h_1)E[S_2] + h_2E[S_1]} \geq \frac{m(m-1)}{2}, \quad (16)$$

then policy $$\omega_0$$ is better than policy $$\omega_{-1}$$. Finally, by combining (15) and (16), our H-Heuristic concludes that, in heavy traffic, policy $$\omega_0$$ with parameter $$m$$ is better than policies $$\omega_1$$ and $$\omega_{-1}$$ if

$$\frac{m(m-1)}{2} \leq \mathcal{R}_{12} \leq \frac{m(m+1)}{2}. \quad (17)$$

Using the same approach, it can be shown that if (17) holds, policy $$\omega_0$$ is also better than all limited policies with parameters greater than or equal to $$m + 2$$, and less than or equal to $$m - 2$$. Therefore, as we mentioned before, our H-Heuristic suggests that, in heavy traffic, the optimal limit of the limited policy can be approximated by $$m^*$$ that satisfies (10).

Note that our H-Heuristic uses (10) and obtains an approximation for the optimal limit of the limited policy independent of the arrival process. In Section 7 we perform an extensive numerical study and show that the optimal limits obtained by our H-Heuristic for heavy traffic systems are also good candidates for the optimal limits of the limited policy in systems under lower traffic intensities.

5. LIMITED POLICY WITH STARTUP BATCH

In this section we turn our focus from the robustness of the optimal limited policy (with respect to the arrival process) to its cost effectiveness. As we mentioned, the limited policy is an attractive and cost-effective policy among the class of nonidling policies. However, when its performance is compared with policies outside the class of nonidling policies, limited policies might have a larger expected total cost. This is mainly because the limited policy controls the maximum number of jobs processed in a cycle ($$M$$) and has no control over the minimum number of jobs processed in a cycle (since it does not allow server idling).

In this section we revise the limited policy by allowing idling in stage and adding a control limit $$M_s$$ on the minimum number of customers served in a cycle. We allow idling in stage 1 only after the server switches to stage 1 and before s/he starts to serve the first customer in that cycle. Our new revised limited policy, which we call “Limited Policy with Startup Batch” or LPSB, is as follows: The server applies an exhaustive policy in stage 2, and after switching to stage 1, s/he remains idle until the number of customers in stage 1 reaches to $$M_s$$ ($$M_s < M$$). S/he then starts serving customers until stage 1 becomes either empty or at most $$M_s$$ customers are served, whereupon s/he switches to stage 2. Applying this policy, the number of customers served in a cycle is between $$M_s$$ and $$M$$.

Note that allowing idling only after the server switches to stage 1 to wait for a batch of size $$M_s$$ to form, becomes critical in light traffic systems as $$h_2 - h_1$$ increases. When $$h_2 - h_1$$ is large, keeping customers waiting in stage 1 becomes less costly than keeping them waiting in stage 2. Thus, to guarantee a minimum number of $$M_s$$ customers served in a cycle, it makes sense that the server will idle in stage 1 waiting for $$M_s$$ customers to arrive, rather than serving each customer at stage 1 and then idling while waiting for more customers to arrive. In the latter case, when traffic intensity is low, the server will be highly likely to end up having an empty stage 1 and holding less than $$M_s$$ customers in stage 2.

The limited policy with startup batch is similar to the ($$s$$, $$S$$) policy in classical inventory models in the following senses: (i) Both policies tend to minimize holding and setup costs, (ii) both policies have two thresholds, (iii) threshold $$M$$ controls the maximum number of served customers in a cycle, while threshold $$S$$ controls the maximum inventory in a period, and (iv) threshold $$M_s$$ controls the minimum number of customers served in a cycle, while $$s$$ tends to control the minimum inventory in a period.
5.1. The Light-Traffic Heuristic (L-Heuristic)

The startup batch size $m_s$ of the LPSB plays the main role in minimizing the cost in systems with very low utilization, since it controls the minimum number of customers served in a cycle. Here we use an approach similar to that in Section 4 in order to obtain an approximation for the optimal startup batch size for LPSB in systems under light traffic. Our heuristic method, which we call the Light-Traffic Heuristic or the L-Heuristic, suggests that if we define $B_{12}$ as

$$B_{12} = \frac{K_1 + K_2}{h_2(E[S_1] + E[S_2]) + h_1(E[S_1] + 1/\lambda)},$$  \hspace{1cm} (18)

then integer $m_s^*$ which satisfies

$$\frac{m_s^*(m_s^* - 1)}{2} \leq B_{12} \leq \frac{m_s^*(m_s^* + 1)}{2},$$  \hspace{1cm} (19)

can be used as a good approximation for the optimal startup batch size of the LPSB in systems under light traffic.

To illustrate the basis for our heuristic approach, consider a two-stage system with a very low utilization $\rho$, i.e., where $\rho$ is very close to zero. Now consider the time at which a customer has just arrived while the server is idle at stage 1, and the number of customers at that stage then becomes $m_s$. If we define $A^{(m)}$ as the number of arrivals during service times of the $m_s$ customers in stages 1 and 2, then in light traffic systems, our L-Heuristic assumes that there will always be no arrivals during the service times of the $m_s$ customers; it therefore also assumes that $Pr[A^{(m)} = 0] \approx 1$, while $Pr[A^{(m)} \geq 1] \approx 0$. Let $E[S_i|A^{(m)} = 0]$ be the expected service time of the $m_s$ customers at stage $i$, given the fact that no customers arrived during the service times of the $m_s$ customers. Thus, in light traffic, our L-Heuristic finds an approximation for $E[S_i|A^{(m)} = 0]$ as follows:

$$E[S_i] = E[S_i|A^{(m)} = 0] Pr[A^{(m)} = 0] + E[S_i|A^{(m)} \geq 1] Pr[A^{(m)} \geq 1] \approx E[S_i|A^{(m)} = 0].$$  \hspace{1cm} (20)

Note that under the assumption of having no arrivals during the service time of $m_s$ customers in stages 1 and 2, (i) an LPSB with startup batch size $m_s$ will serve exactly $m_s$ customers in every cycle, and (ii) when the server switches back to the first stage, s/he will find that stage empty. A good approximation for the optimal size of the startup batch can therefore be obtained by first assuming that the server always serves $m_s$ customers in each busy period (i.e., in each cycle), and then comparing the expected total costs of these busy periods under different LPSB policies. To do this, consider LPSB1 with batch size $m_s$, and let $E[B_{m_s}]$ be the expected total holding and switching costs of each busy period generated under LPSB1 with batch size $m_s$. Our L-Heuristic approximates $E[B_{m_s}]$ as

$$E[B_{m_s}] = \frac{m_s(m_s - 1)}{2} \left( \frac{1}{\lambda} \right) + h_1 \frac{m_s + 1}{2} E[S_1|A^{(m)} = 0] + h_2 \frac{m_s(m_s - 1)}{2} E[S_2|A^{(m)} = 0] + h_2 \frac{m_s + 1}{2} E[S_2|A^{(m)} = 0] + K_1,$$  \hspace{1cm} (21)

or, in light of (20),

$$E[B_{m_s+1}] \approx \frac{m_s(m_s + 1)}{2} \left( \frac{1}{\lambda} \right) + h_1 \frac{m_s + 1(m_s + 2)}{2} E[S_1] + K_2 + h_2 \frac{m_s + 1}{2} E[S_1] + h_2 \frac{(m_s + 1)(m_s + 2)}{2} E[S_2] + K_1.$$  \hspace{1cm} (22)

The first term on the right-hand side of (23) is the expected holding cost for customers waiting in stage 1 for the arrival of the $m_s$th customer in that busy period. The remaining terms are switching costs and the expected total holding costs of serving $m_s$ customers in stages 1 and 2.

Similarly, in light traffic, our L-Heuristic approximates $E[B_{m_s+1}]$, the expected total holding and switching cost in each busy period under policy LPSB2 with batch size $m_s + 1$ by

$$E[B_{m_s+1}] = \frac{m_s(m_s + 1)}{2} \left( \frac{1}{\lambda} \right) + h_1 \frac{m_s + 1(m_s + 2)}{2} E[S_1] + K_2 + h_2 \frac{m_s + 1}{2} E[S_1] + h_2 \frac{(m_s + 1)(m_s + 2)}{2} E[S_2] + K_1.$$  \hspace{1cm} (23)

Since under our heuristic assumption, policy LPSB1 serves $m_s(m_s + 1)$ customers in $m_s + 1$ busy periods, while policy LPSB2 serves the same number of customers in $m_s$ busy periods, then

$$m_s E[B_{m_s+1}] - (m_s + 1) E[B_{m_s}] = \frac{m_s(m_s + 1)}{2} [h_2(E[S_1] + E[S_2]) + h_1(E[S_1] + 1/\lambda)] - (K_1 + K_2)$$  \hspace{1cm} (24)
represents an approximation for the difference between the expected total costs under policies LPSB1 and LPSB2 in serving \( m_s(m_s + 1) \) customers. If (24) is positive, meaning that

\[
B_{12} = \frac{K_1 + K_2}{h_s(E[S_1] + E[S_2]) + h_s(E[S_1] + 1/\lambda)} < \frac{m_s(m_s + 1)}{2}, \tag{25}
\]

then our L-Heuristic concludes that serving \( m_s(m_s + 1) \) customers under policy LPSB1 will generate an expected total holding and switching cost which is less than that under policy LPSB2.

Using the same line of argumentation as in our H-Heuristic, and comparing LPSB1 with other LPSB policies (i.e., policies with startup batch sizes greater than \( m_s + 1 \), and policies with startup batch sizes smaller than \( m_s \)), our L-Heuristic concludes that the integer \( m_s^* \) which satisfies (19) can be considered to be a good approximation for the startup batch size of the LPSB in light traffic.

The startup batch size \( m_s^* \) obtained by our L-Heuristic is for systems in light traffic and thus may not be a good approximation for systems under medium or heavy traffic. Nevertheless, we show in the next section how (19) can also be used to construct a method to obtain a good approximation for the optimal startup batch size in systems that are not under light traffic.

5.2. The Light and Heavy Traffic (L&H) Heuristic

In systems under light traffic, there is a very small probability (almost zero) that an LPSB with startup batch size \( m_s^* = 1 \) generates a cycle which serves \( m_s^* \) customers. However, as traffic intensity increases, this probability becomes larger and an LPSB with startup batch size \( m_s^* - 1 \) will generate more cycles with at least \( m_s^* \) served customers in each cycle. For example, consider an LPSB with startup batch of size \( m_s = j \). Under this policy the server starts serving customers at stage 1 when the number of customers in that stage reaches \( j \). However, while \( j \) customers are being served at stage 1, on average approximately \( jE[S_1]/(1/\lambda) \) new customers arrive at that stage. In systems with medium or high utilization, the number of new arriving customers (i.e., \( jE[S_1]/(1/\lambda) \)) can be a large number. Some of these customers will be served in the same cycle that the \( j \) customers are served. Thus, when an LPSB with startup batch size \( j \) is used, one would expect to see a large number of cycles in which at least \( Y(j) = j + \lfloor jE[S_1]/(1/\lambda) \rfloor \) customers are served.

Note in the above example that we did not discuss control limit \( M^* \) of the optimal LPSB. As we will show in the next section, this limit can be very closely approximated by our H-Heuristic. Based on the above example, as well as our L-Heuristic and H-Heuristic, we develop the following heuristic, which we call the L&H-Heuristic. Our L&H-Heuristic approximates the optimal parameters \( (M_s^*, M^*) \) of the LPSB policy by \( (\tilde{M}_s^*, \tilde{M}^*) \).

Observe in step 3 that the \( \tilde{M}_s^* = m_s^* \) obtained in step 2 (by our L-Heuristic method) for systems with light traffic is revised (similar to our example) to incorporate the system’s traffic intensity. The interesting character of the L&H-Heuristic is that it approximates the optimal parameter \( M^* \) without using any information regarding the arrival process, and approximates \( M_s^* \) only by using the arrival rate. In the next section we evaluate the performances of policies generated by the L&H-Heuristic by comparing them with the global optimal policy.

6. NUMERICAL STUDY

In this section, we report the results of a numerical study we conducted. The purpose of the study was: (i) to investigate the robustness of the optimal limited policy with respect to the arrival process and (ii) to compare the performance of the LPSB policy generated by the L&H-Heuristic with the global optimal policy.
Table 1. Comparison of optimal limits $M^*$ and $m^*$ in systems with $\rho = 0.3, 0.75$ and 0.95.

<table>
<thead>
<tr>
<th>$R_{12}$</th>
<th>$m^*$</th>
<th>$TC_{m^*}$</th>
<th>$M^*$</th>
<th>$TC_{M^*}$</th>
<th>$\delta$</th>
<th>$m^*$</th>
<th>$TC_{m^*}$</th>
<th>$M^*$</th>
<th>$TC_{M^*}$</th>
<th>$\delta$</th>
<th>$m^*$</th>
<th>$TC_{m^*}$</th>
<th>$M^*$</th>
<th>$TC_{M^*}$</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>2</td>
<td>22.69</td>
<td>2</td>
<td>22.69</td>
<td>0.000</td>
<td>2</td>
<td>68.75</td>
<td>2</td>
<td>68.75</td>
<td>0.000</td>
<td>2</td>
<td>82.50</td>
<td>2</td>
<td>82.50</td>
<td>0.000</td>
</tr>
<tr>
<td>2.5</td>
<td>2</td>
<td>25.58</td>
<td>3</td>
<td>25.56</td>
<td>0.078</td>
<td>2</td>
<td>75.54</td>
<td>2</td>
<td>75.54</td>
<td>0.000</td>
<td>2</td>
<td>85.30</td>
<td>2</td>
<td>85.30</td>
<td>0.000</td>
</tr>
<tr>
<td>2.9</td>
<td>3</td>
<td>28.47</td>
<td>3</td>
<td>28.44</td>
<td>0.105</td>
<td>2</td>
<td>82.32</td>
<td>3</td>
<td>82.21</td>
<td>0.134</td>
<td>2</td>
<td>88.10</td>
<td>2</td>
<td>88.10</td>
<td>0.000</td>
</tr>
<tr>
<td>3.1</td>
<td>3</td>
<td>29.88</td>
<td>3</td>
<td>29.47</td>
<td>0.000</td>
<td>3</td>
<td>85.55</td>
<td>3</td>
<td>85.55</td>
<td>0.000</td>
<td>3</td>
<td>89.82</td>
<td>3</td>
<td>89.82</td>
<td>0.000</td>
</tr>
<tr>
<td>4.0</td>
<td>3</td>
<td>36.36</td>
<td>3</td>
<td>36.36</td>
<td>0.000</td>
<td>3</td>
<td>100.55</td>
<td>3</td>
<td>100.55</td>
<td>0.000</td>
<td>3</td>
<td>94.73</td>
<td>3</td>
<td>94.73</td>
<td>0.000</td>
</tr>
<tr>
<td>5.0</td>
<td>3</td>
<td>43.57</td>
<td>3</td>
<td>43.57</td>
<td>0.000</td>
<td>3</td>
<td>117.21</td>
<td>3</td>
<td>117.21</td>
<td>0.000</td>
<td>3</td>
<td>105.10</td>
<td>3</td>
<td>105.10</td>
<td>0.000</td>
</tr>
<tr>
<td>5.9</td>
<td>3</td>
<td>50.05</td>
<td>3</td>
<td>50.05</td>
<td>0.000</td>
<td>3</td>
<td>132.21</td>
<td>3</td>
<td>132.21</td>
<td>0.000</td>
<td>3</td>
<td>124.02</td>
<td>3</td>
<td>124.02</td>
<td>0.000</td>
</tr>
<tr>
<td>6.1</td>
<td>4</td>
<td>51.52</td>
<td>3</td>
<td>51.47</td>
<td>0.097</td>
<td>4</td>
<td>135.70</td>
<td>3</td>
<td>134.55</td>
<td>0.855</td>
<td>4</td>
<td>134.67</td>
<td>3</td>
<td>134.67</td>
<td>0.000</td>
</tr>
<tr>
<td>8.0</td>
<td>4</td>
<td>65.20</td>
<td>4</td>
<td>65.20</td>
<td>0.000</td>
<td>4</td>
<td>167.31</td>
<td>4</td>
<td>167.31</td>
<td>0.000</td>
<td>4</td>
<td>114.63</td>
<td>4</td>
<td>114.63</td>
<td>0.000</td>
</tr>
<tr>
<td>9.9</td>
<td>4</td>
<td>78.89</td>
<td>5</td>
<td>78.79</td>
<td>0.127</td>
<td>4</td>
<td>198.91</td>
<td>5</td>
<td>198.84</td>
<td>0.035</td>
<td>4</td>
<td>123.74</td>
<td>4</td>
<td>123.74</td>
<td>0.000</td>
</tr>
<tr>
<td>10.1</td>
<td>5</td>
<td>80.23</td>
<td>5</td>
<td>80.23</td>
<td>0.000</td>
<td>5</td>
<td>202.16</td>
<td>5</td>
<td>202.16</td>
<td>0.000</td>
<td>5</td>
<td>124.02</td>
<td>5</td>
<td>124.02</td>
<td>0.000</td>
</tr>
<tr>
<td>12.5</td>
<td>5</td>
<td>97.48</td>
<td>5</td>
<td>97.48</td>
<td>0.000</td>
<td>5</td>
<td>242.00</td>
<td>5</td>
<td>242.00</td>
<td>0.000</td>
<td>5</td>
<td>134.67</td>
<td>5</td>
<td>134.67</td>
<td>0.000</td>
</tr>
<tr>
<td>14.9</td>
<td>5</td>
<td>114.70</td>
<td>5</td>
<td>114.70</td>
<td>0.000</td>
<td>5</td>
<td>281.84</td>
<td>5</td>
<td>281.84</td>
<td>0.000</td>
<td>5</td>
<td>145.32</td>
<td>5</td>
<td>145.32</td>
<td>0.000</td>
</tr>
</tbody>
</table>

6.1. Robustness of the Optimal Limited Policy

In order to investigate the robustness of the optimal limited policy with respect to the arrival process, we compared the performance of limited policies obtained (independent of the arrival process) by our H-Heuristic with optimal limited policies in systems with different interarrival distributions and traffic intensities. We examined more than 1000 problems that we created based on: (i) four distributions for inter-arrival times (i.e., Exponential, Normal, Erlang, and Hyperexponential), (ii) seven distributions for service times (i.e., Exponential, Normal, Erlang, Hyperexponential, Triangular, Uniform, and deterministic), (iii) five different traffic intensities (i.e., $\rho = 0.30, 0.60, 0.75, 0.90, 0.95$), (iv) three cases for $E[S_1]/E[S_2]$ (i.e., $E[S_1]/E[S_2] < 1$, or $= 1$, or $> 1$), and (v) the following set of values for $R_{12}$:

\[ R_{12} \in \{2.1, 2.5, 2.9, 3.1, 4, 5, 5.9, 6.1, 8, 9.9, 10.1, 12.5, 14.9\}. \]

Note that the values for $R_{12}$ is chosen to be in the middle and close to the boundaries of intervals $[m(m - 1)/2, m(m + 1)/2], m = 3, 4, 5$. These intervals are used by our H-Heuristic in order to determine the approximation for the parameter of the optimal limited policy, independent of the arrival process.

For each problem we used simulation to evaluate the objective function (1) for different values of $M$ and obtained the optimal limit $M^*$ that minimizes that function. Then we compared $M^*$ with $m^*$ obtained by our H-Heuristic. If $M^*$ was different from $m^*$, then their expected total costs per unit time were compared to evaluate the following error:

\[
\delta = \frac{E[TC(m^*)] - E[TC(M^*)]}{E[TC(M^*)]} \times 100.
\]

Table 1 presents three different sets of problems with different arrival distributions and traffic intensities. Problems with $\rho = 0.3$ are for a system with normally distributed interarrival times with parameters ($\mu_a = 20$, $\sigma_a = 6$), normally distributed service times in stage 1 with ($\mu_1 = 4$, $\sigma_1 = 1$), and service times in stage 2 distributed uniformly on (1, 3). In problems with $\rho = 0.75$, the system has the same arrival process (as for cases with $\rho = 0.3$) but different service time distributions. The service time distributions in stages 1 and 2 are triangular with parameters (4, 5, 6) and Normal with ($\mu_2 = 10$, $\sigma_2 = 3$), respectively. Finally, cases with $\rho = 0.95$ refers to a system in which the interarrival distribution is Erlang with 2 phases and average 20, and service time distributions in stages 1 and 2 follow a Uniform distribution on (4, 8) and Normal distribution with ($\mu_2 = 13$, $\sigma_2 = 4$), respectively. (Note in Table 1 that $TC_{m^*}$ and $TC_{M^*}$ are used in place of $E[TC(m^*)]$ and $E[TC(M^*)]$, respectively, to simplify the notation in the table.)

Based on our simulation study, we found that in almost 70% of the cases, our H-Heuristic yielded the exact optimal limit $M^*$, and when it did not, the average and maximum errors were below 0.4% and 5%, respectively. In other words, in 70% of our cases the limit $m^*$, obtained independent of the arrival process, is the exact optimal limit, and if it is not, the expected total cost per unit time of using a limited policy with limit $m^*$ is very close to the cost of the optimal limited policy. These results show that one can accurately approximate the optimal limit of the limited policy without having any information regarding the arrival process.

6.2. LPSB Policy vs. The Optimal Policy

As mentioned earlier, the global optimal policy which minimizes the expected total cost per unit time in a two-stage tandem queue has a complex structure (see Iravani,
For each traffic intensity $h$, we would like to emphasize that a major part of the difference between the expected total cost of the LPSB policy and the global optimal policy is due to the fact that the LPSB is a policy generated by the L&H-Heuristic and the global optimal policy is the complex structure global optimal policy. Note that the average gap of 1.5% between the cost of LPSB policy and the global optimal policy is large. Table 2 depicts one set of these problems for cases of the traffic intensities $h_1 = 10$ and $h_2 = 40$.

Based on our numerical study we have found that the L&H-Heuristic generates policies with the expected total cost, on average, 1.5% higher than the expected total cost of the global optimal policy. We also found that in 65% of the cases, this difference ($\zeta$) was less than 1%, and in only 15% of cases the difference was larger than 3%. The maximum difference we found was about 8%.

We would like to emphasize that a major part of the difference between the expected total cost of the LPSB policies generated by the L&H-Heuristic and the global optimal policy is due to the fact that the LPSB policy is a static policy that does not imitate the complex dynamic structure of the global optimal policy. In other words, the average gap of 1.5% between the cost of LPSB policy and the cost of the global optimal policy is mostly the price we pay in order to use an easily implementable policy in place of the complex structure global optimal policy. Note that we examined over 500 problems with traffic intensities $\rho \in \{0.3, 0.5, 0.6, 0.7, 0.8, 0.9\}$, holding costs $h_1 = 10$, and $h_2 \in \{10, 15, 20, 40\}$, under $\mathbb{R}_{12} \in \{2.5, 5, 8, 14, 25\}$. For each traffic intensity $\rho$ we created five cases with different $E[S_1]/E[S_2]$ ratios. We assumed exponential processing and interarrival times, so we were able to use a Markov decision process to obtain the global optimal policy.
other classes of policies with smaller gaps can probably be generated using approximation methods such as Brownian approximations. However, in order to have a smaller gap than 1.5%, those policies should behave more like the complicated global optimal policy than simple static policies such as LPSB. This makes them less implementable in practice.

7. OPTIMAL LIMITED POLICY IN MULTISERVER TANDEM QUEUES

The results obtained in previous sections for the two-stage serial production system with one worker can be applied in more complex systems. More specifically, our results can be used as heuristic policies in (i) \(N\)-station serial production systems with one worker and (ii) \(N\)-station serial production systems with multiple workers.

7.1. \(N\)-Station Serial Production System with One Worker

The robustness of the optimal limited policies makes them attractive heuristic policies in \(N\)-stage tandem queues if the server applies a limited policy with limit \(M\) in all stages and only switches back to upstream stages when downstream stages are empty. This is not an unrealistic assumption since the holding cost in downstream stages typically exceeds that in upstream stages, and, thus, it makes sense for the server to clear downstream stages before switching back to upstream stages. Applying a limited policy in this manner, the server actually implements the limited policy in stage 1 and an exhaustive policy in stages 2, 3, \ldots, \(N\), and always switches from stage \(j\) to \(j + 1\) (\(j = 2, 3, \ldots, N - 1\)), except when finishing the service of the last customer in stage \(N\), whereupon she switches back to stage 1.

The robustness of the optimal limited policy in \(N\)-stage systems becomes clear when the \(N\)-stage system is converted to an equivalent two-stage system with the same arrival process, switching costs \(K_1^{(E)}\) and \(K_2^{(E)}\), holding costs \(h_1^{(E)}\) and \(h_2^{(E)}\), and service times \(S_1^{(E)}\) and \(S_2^{(E)}\) in stages 1 and 2, respectively, where \(S_1^{(E)} = S_1\), \(h_1^{(E)} = h_1\), \(K_1^{(E)} = K_1\), and

\[
S_2^{(E)} = S_2 + S_3 + \cdots + S_N, \\
K_2^{(E)} = K_2 + K_3 + \cdots + K_N,
\]

and \(h_2^{(E)}\) will be a function of holding costs in stages 2 to \(N\). If the sample paths of the \(N\)-stage tandem queue and its equivalent two-stage tandem queue are compared it is found that: (i) Both have the same busy and idle periods; (ii) both switch to stage 2 at exactly the same time; (iii) both switch back to stage 1 at exactly the same time; (iv) both have cycles with the same lengths (recall that a cycle is the time elapsed between two consecutive switches to stage 1); (v) both process the same number of jobs in each cycle; and (vi) both have the same number of jobs in stage 1 at any time. Since in both systems the optimal limited policy dictates when the server must switch from stage 1 to stage 2, and since there are identical return switch times to stage 1, it follows that the robustness of the optimal limited policy in the equivalent two-stage system corroborates the robustness of the optimal limited policy in the \(N\)-stage system. If \(\Pi_{1N}\) and \(\Pi_{1N}^3\), \(R_{1N}\) and \(B_{1N}\) are defined as follows:

\[
\Pi_{1N} = \sum_{i=1}^{N} (h_i - h_1)E[S_i] + \sum_{i=2}^{N} h_iE[S_{i-1}], \\
\Pi_{1N}^3 = \sum_{i=1}^{N} K_i, \\
R_{1N} = \frac{\Pi_{1N}}{h_1(E[S_1] + \frac{1}{\lambda}) + \sum_{i=2}^{N} h_i(E[S_i] + E[S_{i-1}])}, \\
B_{1N} = \frac{\Pi_{1N}}{h_1(E[S_1] + \frac{1}{\lambda}) + \sum_{i=2}^{N} h_i(E[S_i] + E[S_{i-1}])},
\]

then it can be shown that all the results presented in this paper for the two-stage system (i.e., Proposition 1 for cases with deterministic service times, and L&H-Heuristic for cases with stochastic service times) can be used in \(N\)-stage systems when \(B_{12}, \Pi_{12}, \Pi_{12}^3\) are replaced with \(B_{1N}, \Pi_{1N}, \Pi_{1N}^3\), respectively. The limited policy is even more attractive in multistage systems since as the number of station increases, the structure of the global optimal policy becomes more difficult to implement in practice.

7.2. \(N\)-Station Serial Production Systems with Multiple Workers

Consider a system where a number of \(N\)-station serial systems, each with one moving server, are placed in a series configuration. This is the basic structure of a class of U-shaped lines or manufacturing cells which we call lines or cells with nonoverlapping sequential (or NS) working zones. In lines or cells with NS working zones, each worker, say worker \(k\), is in charge of operations (stations) \(k_1, k_2, \ldots, k_i\), where operation \(k_u\) is the immediate precedence for operation \(k_{u+1}\). Furthermore, in lines or cells with NS working zones each operation is assigned to one worker, and thus there is no overlap among working zones (see Fig. 3).
The major problem with developing models to measure the performance of these systems under any particular policy is the difficulty in characterizing the job arrival process to each working zone. This is due to the fact that the arrival process at each working zone is actually the job departure process from its upstream working zone. The optimization analysis of these systems is even more complicated due to the large dimension of the state space. Even if the global optimal policy could be obtained, its complex structure would make it almost impossible to implement in practice. Thus, the need for a class of simple structured suboptimal policies which perform reasonably well and also are easy to obtain becomes apparent. The class of limited policies with startup batch is a good candidate, since:

1. They are easy to implement. The global optimal policy in a working zone depends on the positions of the jobs in that working zone as well as the positions of the jobs and workers in all other working zones. If workers are supposed to follow the global optimal policy in their working zones, they may need to revise their actions every time a job is processed anywhere in the system or every time any worker switches to another station. This is not practical and it clearly shows the need for a class of policies such as LPSB which can be implemented in each working zone independent of the status of the other working zones.

2. They are easy to optimize. Finding the optimal LPSB in each working zone does not require the analysis of the departure process of the upstream working zone. If jobs arrive with rate $\lambda$ to station 1 (working zone 1 in Fig. 3), under steady state conditions, the job arrival rates at working zones 2 and 3 are also $\lambda$. Thus, our heuristic can be used to approximate the parameters of the optimal LPSB in each working zone.

**Figure 3.** A U-shaped line with 12 workstations, 3 workers, and nonoverlapping sequential working zones.

**Remark:** In some manufacturing cells, there is enough material in work station 1; so that the system does not operate in a produce-to-order fashion. Our results also hold in those cases since the first working zone can be viewed as a $N$-stage tandem queue with one moving server under heavy traffic. This is under the assumption that the objective is to minimize the holding and switching costs. However, if the goal is to maximize the throughput, idling policies such as LPSB might not perform well. Finding the optimal policies under these circumstances requires further research and is not in the scope of this paper.

**8. CONCLUSION AND FURTHER RESEARCH**

The emergence of agile manufacturing systems which use cross-trained workers and/or multifunctional machinery has opened a new chapter in modeling production systems. In these new systems, workers are in charge of more than one station and are allowed to move among those stations to process jobs. The analysis of these new production systems is far more complicated than traditional production systems, since in addition to the inherent complexity of queuing problems, they also carry the complications of scheduling problems. To simplify the analysis, almost all the literature on production systems with cross-trained workers or multifunctional machines assumes homogeneous Poisson job arrival process, since without the Markovian property the analysis of these systems is very complex. Under these circumstances information regarding policies whose performance is independent of or robust with respect to the arrival process becomes very crucial.

In this paper we focused on the properties of limited policies in a serial production system with a cross-trained worker. We showed that when setup times are insignificant, the optimal limited policy can be approximated, and in some cases obtained, independent of the arrival process or service time distributions. This nice property makes the limited
policy an attractive policy in systems in which the arrival process has a complicated structure or is even unknown. We also developed a heuristic method that generates reasonably cost-effective policies using minimum information regarding the arrival process and service time distributions. Further research should focus on finding other robust policies in agile production systems with different job routing schemes. Finding robust policies in serial configurations with the objective of maximizing throughput would also be of interest.

APPENDIX: PROOF OF PROPOSITION 1

It is clear from Lemma 2 that if $R_{12}^1 > 1$, then a limited policy with $M = 1$ is not optimal and $M^* \geq 2$. We must show that if $1 \leq R_{12}^1 \leq 2$, a limited policy with $M = 2$ has a lower expected total cost than a limited policy with $M \geq 3$.

Consider two policies: $\beta$ (a limited policy with $M = 2$) and $\pi$ (a limited policy with $M \geq 3$). A sample path of $\pi$ includes two types of busy periods: (i) type-I busy periods in which there is no cycle with more than two served customers; and (ii) type-II busy periods in which there is at least one cycle which serves more than two customers. Policies $\pi$ and $\beta$ generate the same expected total cost during type-I busy periods, since in those periods $\pi$ behaves exactly like $\beta$. Furthermore, according to Lemma 3, all busy periods start and end at the same time under both policies $\pi$ and $\beta$. Thus, comparing $\pi$ and $\beta$ reduces to comparing these two policies in a type-II busy period. In our proof we show that policy $\beta$ is always better than $\pi$ in the first type-II busy period. The proof for the second, third, and later type-II busy periods is exactly the same, and hence is not presented here.

The proof has two parts. PART ONE: Based on policy $\pi$, we construct policy $\beta_0$ which serves a maximum of two customers in each cycle, and we show that if $R_{12}^1 \leq 2$, then policy $\beta_0$ is always better than $\pi$. PART TWO: Based on policy $\beta_0$, we construct policy $\beta$ which is a limited policy with limit $M = 2$. We then show that if $1 \leq R_{12}^1 \leq 2$, then policy $\beta$ is always better than policy $\beta_0$.

PART ONE: Without loss of generality we assume that under policy $\pi$, the first type-II busy period that starts at time $t_1$ and ends at time $t_2$ consists of $k$ cycles in which $N_2^1$, $N_2^2$, $\ldots$, $N_2^k$ customers are served. Suppose that policy $\beta_0$ behaves exactly the same as $\pi$ except for cycles in which more than two customers are served (i.e., cycles with $N_2^j \geq 3$ for some $j \in \{1, 2, \ldots, k\}$). For these cycles, policy $\beta_0$ behaves as follows: (i) if $N_2^j$ is an even number, policy $\beta_0$ breaks the given cycle into $N_2^j/2$ consecutive cycles serving 2 customers in each cycle; (ii) if $N_2^j$ is an odd number, policy $\beta_0$ breaks the given cycle into $[(N_2^j - 1)/2] + 1$ cycles serving two customers in the first $(N_2^j - 1)/2$ cycles and one customer in the last cycle. It is easy to show that if $R_{12}^1 = 2$, then according to Corollary 2, policy $\beta_0$ will have a lower total average cost in all those cycles than policy $\pi$. For example, consider a given cycle in which $N_2^j \geq 3$ is an odd number. If $N_2^j$ is broken into two consecutive cycles serving $n_1 = 2$ and $n_2 = N_2^j - 2$ customers, then, using Lemma 1, we have

$$E[TC_{\pi}^{l_0}] - E[TC_{\beta_0}^{l_0}] = 2(N_2^j - 2)/R_{12}^1 - k_{12}.$$

On the other hand, when $R_{12}^1 \leq 2$, we will have $2R_{12}^1 - 3k_{12} \geq 0$. This guarantees that, for $N_2^j \geq 3$, we have $2(N_2^j - 2)/R_{12}^1 - k_{12} \geq 0$, which in turn implies that $E[TC_{\pi}^{l_0}] - E[TC_{\beta_0}^{l_0}] \geq E[TC_{\pi}^{l_0}] - E[TC_{\beta_0}^{l_0}]$. In other words, when $R_{12}^1 \leq 2$, then the policy that breaks a given cycle with $N_2^j \geq 3$ served customers will result in a lower expected total cost. Continuing in this manner, it can be shown that if $R_{12}^1 \leq 2$, then policy $\beta_0$ that breaks cycles with $N_2^j$ served customers into $[(N_2^j - 1)/2] + 1$ cycles will have a lower expected total cost than policy $\pi$.

Note that policy $\beta_0$ is not a limited policy with $M = 2$. It is a nonidling policy that serves a maximum of 2 customers in each of its cycles. In PART TWO below, based on policy $\beta_0$, we construct limited policy $\beta$ with limit $M = 2$, and we show that $\beta$ has lower total average holding and switching costs than $\beta_0$ when $R_{12}^1 = 1$.

PART TWO: Without loss of generality, we assume that the first type-II busy period under policy $\beta_0$ has $l$ cycles, where $l > k$. Also, let $N_0^j$ be the number of customers served under policy $\beta_0$ in its $j$th cycle ($j \leq l$) of that busy period, where the $j$th cycle starts at $E_0^j$. Note that $N_0^j \leq 2$ for all $j = 1, 2, \ldots, l$, and $E_0^j = t_0$.

We will show how, based on policy $\beta_0$, a limited policy with limit $M = 2$ can be constructed through an iterative process. We will also show that, at each iteration, the constructed policy will have a lower cost than policy $\beta_0$. Recall that the first type-II busy period starts and ends at $t_0$ and $t_1$, respectively. This means that policy $\beta_0$ behaves like a limited policy up to time $t_0$. Our iterative process is as follows:

STEP 1: Consider the first cycle under policy $\beta_0$ in which $N_0^1$ customers are served. This cycle starts at time $t_0$ and ends at time $E_0^1$. There are two possible scenarios: (i) Scenario 1 occurs when $N_0^1 = 2$, and (ii) Scenario 2 occurs when $N_0^1 = 1$.

Scenario 1: Under this scenario, policy $\beta_0$ indeed behaves like a limited policy with limit $M = 2$ in its first cycle in the first type-II busy period. This means that policy $\beta_0$ is a limited policy with limit $M = 2$ up to time $E_0^1$. We rename policy $\beta_0$ as $\beta_1$, and we can now focus on the next (i.e., the second) cycle of the busy period under policy $\beta_1$. This will be done in Step 2.

Scenario 2: Under this scenario there are two possible cases: Case 1, $N_0^1 = 1$, $N_0^2 = 1$, and Case 2, where $N_0^1 = 1$, $N_0^2 = 2$.

Scenario 2, Case 1: In this case $N_0^1 = 1$ and $N_0^2 = 1$, which implies that two customers are served in two consecutive cycles, namely, the first and second cycles. For this case, we have the following two possibilities:

- In the first cycle, if stage 1 becomes empty upon service completion there, then policy $\beta_1$ is indeed behaving like a limited policy with limit $M = 2$ in the first cycle. We rename policy $\beta_1$ as $\beta_1$, and we can now focus on the next (i.e., the second) cycle of the busy period under policy $\beta_1$. This will be done in Step 2.

- In the first cycle, if the server finds at least one customer present in stage 1 upon service completion there, then we construct policy $\beta_2$ which follows policy $\beta_0$ except in its first and second cycles, where $\beta_2$ behaves like a limited policy with limit $M = 2$ and serves two customers in one cycle. If $E[TC_{\beta_0, [a,b]}]$ denotes the expected total cost during time interval $[a, b]$ when policy $\xi$ is applied, then

$$E[TC_{\beta_0, [a,b]}] - E[TC_{\beta_1, [a,b]}] = E[TC_{\beta_2, [a,b]}] - E[TC_{\beta_0, [a,b]}].$$

Policy $\beta_0$ actually serves one and then another customer in interval $[t_0, E_0^1]$ in two consecutive cycles; however, policy $\beta_1$ serves these two customers in one cycle. It is clear from Corollary 2 that if $R_{12}^1 \geq 1$, then $E[TC_{\beta_0, [t_0,t_1]}] \geq E[TC_{\beta_1, [t_0,t_1]}]$, which means that policy $\beta_1$, which is a limited policy with limit $M = 2$ up to the end of its first cycle, is always better than $\beta_0$. Thus, we can now focus on the next (i.e., the second) cycle of the busy period under policy $\beta_1$. This will be done in Step 2.

Scenario 2, Case 2: In this case $N_0^1 = 1$ and $N_0^2 = 2$, which entails that policy $\beta_0$ serves 1 and 2 customers in its first and second cycles of the busy
period, respectively. For this case, we will have the following two possibilities:

- In the first cycle, if stage 1 becomes empty upon service completion there, then policy $\pi_1$ is indeed behaving like a limited policy with limit $M = 2$ in its first cycle. We rename policy $\pi_1$ as policy $\pi_2$, and we now can focus on the next (i.e., the second) cycle of the busy period under policy $\pi_2$. This will be done in Step 2.

- In the first cycle, if the server finds at least one customer present in stage 1 upon service completion there, then policy $\pi_1$ can be constructed as follows: $\pi_1$ follows $\pi_0$ except during interval $[t_0, t_1^{[\pi_0]}]$, where policy $\pi_2$ serves $N_1^{[\pi_2]} = 2$ and $N_2^{[\pi_2]} = 1$ customer(s) in two consecutive cycles. Note that during the same interval, policy $\pi_0$ serves $N_1^{[\pi_0]} = 1$ and $N_2^{[\pi_0]} = 2$ customers in two consecutive cycles. Therefore,

$$E[T_{C_0}^{\pi_0} - E[T_{C_0}^{\pi_0}] = E[T_{C_0}^{\pi_0}] - E[T_{C_0}^{\pi_0}].$$

However, according to Corollary 1, $E[T_{C_0}^{\pi_0}] = E[T_{C_0}^{\pi_0}]$. As a result, we see that policy $\pi_2$ is as good as policy $\pi_0$ in terms of the expected cost total. Thus, we can now focus on the next (i.e., the second) cycle of the busy period under policy $\pi_2$. This will be done in Step 2.

- STEP 2: All scenarios in Step 1 result in policy $\pi_2$, which is a limited policy with limit $M = 2$ up to the end of its first cycle. We can now focus on the second cycle under policy $\pi_2$ and construct policy $\pi_3$, which is a limited policy up to the end of its second cycle of the first type-II busy period. The analysis is similar to the way policy $\pi_2$ was constructed. In other words, we return to Step 1 and focus on the second cycle under policy $\pi_2$, in which $N_1^{[\pi_2]}$ customers are served. This process iterates until policy $\pi$ is constructed. Policy $\pi$ will be a limited policy with limit $M = 2$ during the entire first type-II busy period.

\[ \square \]

ACKNOWLEDGMENTS

The authors thank the anonymous Associate Editor and two referees for their valuable comments on this paper. The authors especially express their indebtedness to the Associate Editor for her/his insightful and very thorough comments on our earlier version of this paper. This research was supported by the National Science Foundation under Grant DMI-0000125.

REFERENCES


