THE STOCHASTIC ECONOMIC LOT SCHEDULING PROBLEM: HEAVY TRAFFIC ANALYSIS OF DYNAMIC CYCLIC POLICIES

DAVID M. MARKOWITZ, MARTIN I. REIMAN and LAWRENCE M. WEIN

Logistics Management Institute, McLean, Virginia 22102
Bell Laboratories, Lucent Technologies, Murray Hill, New Jersey 07974
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, lwein@mit.edu

(Received November 1995; revision received July 1997; accepted September 1998)

We consider two queueing control problems that are stochastic versions of the economic lot scheduling problem: A single server processes N customer classes, and completed units enter a finished goods inventory that services exogenous customer demand. Unsatisfied demand is backordered, and each class has its own general service time distribution, renewal demand process, and holding and backordering cost rates. In the first problem, a setup cost is incurred when the server switches class, and the setup cost is replaced by a setup time in the second problem. In both problems we employ a long-run average cost criterion and restrict ourselves to a class of dynamic cyclic policies, where idle periods and lot sizes are state-dependent, but the N classes must be served in a fixed sequence. Motivated by existing heavy traffic limit theorems, we make a time scale decomposition assumption that allows us to approximate these scheduling problems by diffusion control problems. Our analysis of the approximating setup cost problem yields a closed-form dynamic lot-sizing policy and a computational procedure for an idling threshold. We derive structural results and an algorithmic procedure for the setup time problem. A computational study compares the proposed policy and several alternative policies to the numerically computed optimal policy.

We consider a queueing system scheduling problem that is motivated by a situation commonly found in make-to-stock manufacturing. The system consists of a single server, or machine, and multiple customer classes, which will be referred to as *products*. Each product has its own general service time distribution, and completed units are placed into a finished goods inventory; we assume that an ample amount of (costless) raw material inventory is available. Each product has its own renewal demand process that depletes the inventory, and unsatisfied demand is backordered.

We analyze two variants of the scheduling problem. In the *setup cost problem*, a cost is incurred when the machine switches production from one product to another; in the *setup time problem*, a random setup time is incurred when the server switches product. We restrict ourselves to the class of *dynamic cyclic policies*, where each product is serviced once per cycle and the order of production does not change (such cycles are sometimes referred to as rotation cycles). Thus, the server has three scheduling options at each point in time: Produce a unit of the product that is currently set up, change over to the next product in the cycle (and initiate service in the setup cost problem), or remain idle. Although our restriction to this class of policies is motivated by analytical tractability, the regularity induced by these policies eases the task of raw material procurement (e.g., Gallego and Joneja 1994). Each product has its own costs per unit time for holding and backordering a unit in inventory. The objective in the setup time problem is to minimize the long-run expected average inventory costs (that is, holding and backorder costs); the objective in the setup cost problem is to minimize the average inventory and setup costs.

These problems are prevalent in many industries because facilities that operate in a make-to-stock mode typically produce standardized products that require setups. The setup time problem is more realistic than the setup cost problem in most situations, but is also less amenable to analysis. The setup cost problem, however, may be relevant for manufacturing systems that have internalized their setup times; that is, they incur significant material, labor, and/or capital costs to greatly reduce their switchover times.

This dynamic scheduling, or *lot-sizing*, problem is a stochastic version of the classic economic lot scheduling problem ELSP, which is NP-hard (Hsu 1983) and has not been solved in general. Despite the vast literature devoted to the ELSP (see the survey paper by Elmaghraby 1978 and Zipkin 1991 for a list of more recent references), its deterministic viewpoint has probably prevented its widespread industrial use: The solution to a deterministic problem in a make-to-stock setting will not hedge against uncertainty in future service times (e.g., machine failures) and demand, resulting in many costly backorders (see the numerical results in Federgruen and Katalan 1996).

Not surprisingly, the stochastic version of the ELSP appears to be analytically intractable. When the state space is taken to be discrete, the stochastic ELSP (or SELSP) can be viewed as a make-to-stock version of the dynamic scheduling problem for a *polling system*, which is a traditional (i.e., make-to-order) multiclass queue with setups. In fact, our paper can be viewed as a companion to Reiman and Wein

Subject classifications: Inventory/production : stochastic multi-item lot-sizing. Queues: diffusion approximation of scheduling problems.
Area of review: MANUFACTURING OPERATIONS.

0030-364X/00/4801-0136 $05.00
1526-5463 electronic ISSN

(1998), who analyze the dynamic scheduling problem for a two-class polling system. The SELSP is more challenging than the polling scheduling problem, which also appears to defy exact analysis, because of the nonlinear cost structure and the lack of an imposed boundary at the origin. Despite its difficulty, this problem has been the subject of a recent flurry of activity. Graves (1980) develops a Markov decision model for a one-product problem, and uses it to develop a heuristic for the SELSP in a periodic review setting. Leachman and Gascon (1988), Gallego (1990), and Bourland and Yano (1995) develop heuristic lot-sizing algorithms for the ELSP with stochastic demands that are rooted in the solution to the deterministic ELSP; the first of these papers considers a discrete time problem with non-stationary demand. Sharifnia et al. (1991) employ a hierarchical approach to develop heuristic policies for a stochastic fluid version of the problem, where demand is deterministic but the production process is uncertain; Sethi and Zhang (1995) derive some asymptotic results for this problem. Federgruen and Katalan (1995, 1996) develop accurate distributional approximations for polling systems, and use these to analyze the performance of a class of (periodic and cyclic, respectively) base stock policies for the SELSP. Anupindi and Tayur (1998) also consider a class of periodic base stock policies, and use a simulation-based approach (infinitesimal perturbation analysis and gradient search) to obtain good base stock levels for a variety of performance measures. Sox and Muckstadt (1997) formulate the SELSP as a stochastic program and propose a heuristic decomposition algorithm to solve it. Qiu and Loulou (1995) formulate the problem as a semi-Markov decision process, and numerically compute the optimal solution in the two-product case; this is the only paper to date to gain any insight into the nature of the optimal solution to the SELSP.

As in Reiman and Wein, we employ heavy traffic approximations in an attempt to make further progress with this problem. This approach assumes that the server must be busy the great majority of time in order to meet average demand over the long term. We draw heavily upon the results of Coffman et al. (1995, 1998), who derive a heavy traffic averaging principle for a two-class queue (in the absence and presence of setup times, respectively) that employs an exhaustive polling mechanism. Guided by these limit theorems, we assume that the heavy traffic averaging principle holds for our multiclass queueing system under all dynamic cyclic policies. This key assumption allows us to approximate the setup cost and setup time problems by diffusion control problems that are amenable to analysis. Our analysis of the two diffusion control problems leads to proposed policies that are characterized by a dynamic lot-sizing policy and a server idling threshold.

In an attempt to both assess the effectiveness of some simpler policies and synthesize some of the existing literature on the SELSP, we perform a heavy traffic analysis of two alternative policies that reflect the two prototypical philosophies (order-up-to levels and constant lot sizes) that permeate SELSP theory and practice. Then a computational study is undertaken that compares our proposed policies and these two alternative policies to the numerically derived optimal policy for a variety of two-product problems; several five-product problems are also examined.

The explicitness of our results reveals some insights into the nature of the optimal solution to the SELSP. Readers who are not curious about the mathematical details but who wish to obtain a deeper understanding of the SELSP may find it useful to bypass the heavy traffic analysis and focus on §1.8, §2.5, and §3.4, where our key observations are discussed.

The remainder of the paper is divided into four sections. Sections 1 and 2 are devoted to the analysis of the setup cost and setup time problems, respectively. The computational study is described in §3, and concluding remarks are made in §4.

1. THE SETUP COST PROBLEM

1.1. Problem Description

A single server, or machine, produces N types of products. Each product $i = 1, 2, \ldots, N$ has its own general service time distribution with service rate μ_i and coefficient of variation (standard deviation divided by the mean) c_{si}. The demand for each product follows a renewal process, where the mean and coefficient of variation of the interdemand times are given by λ_i^{-1} and c_{di}, respectively. (Our results easily generalize to correlated compound renewal procesess; see Reiman 1984 for details.) Hence, the traffic intensity, or long run average server utilization, is $\rho = \sum_{i=1}^{N}(\lambda_i/\mu_i)$ and $\rho_i = \lambda_i/\mu_i$ is the utilization for product i.

Define the *inventory process* $\tilde{I}_i(t)$ to be the number of units of product i in finished goods inventory at time t. A service completion of product i at time t increases $\tilde{I}_i(t)$ by one, and a unit of product i demanded at time t depletes $\tilde{I}_i(t)$ by one. If a demand for a unit of product i occurs when $\tilde{I}_i(t) \leqslant 0$, we say that this unit of demand is backordered.

Because the scheduler follows a dynamic cyclic policy, only three options are available at each point in time: Produce the product that is currently set up, switch over and initiate production of the next product in the cycle, or sit idle. Because a setup is costly and instantaneous, the option of switching to another product and then idling is clearly suboptimal and will not be considered. The server is assumed to work in a preemptive-resume manner, although the subsequent heavy traffic analysis is too crude to distinguish between this and the non-preemptive discipline.

A cost $\bar{h}_i$ is incurred per unit time for holding a unit of product i in inventory, and a cost $\bar{b}_i$ is incurred per unit time for backordering a unit of product i. Let us also define the *cost indices* $h_i = \bar{h}_i\mu_i$ and $b_i = \bar{b}_i\mu_i$, which represent holding and backorder cost rates per unit of expected work in inventory. (These indices are the analog to the classic "$c\mu$" index in stochastic scheduling theory, see Cox and Smith 1961.) Without loss of generality, the products are numbered so that $h_N = \min_{1 \leqslant i \leqslant N} h_i$. For notational convenience, we assume that $b_N = \min_{1 \leqslant i \leqslant N} b_i$, so that the product with the smallest holding cost index also has the smallest backorder cost index; at the end of §1.6 we show that this restriction does not change the analysis from that of the more general case. Consequently, product N will often be referred to as the *least cost product*.

Because products are produced once per cycle in a fixed sequence, the performance of the system depends on the setup costs only through the total setup cost per cycle, a quantity we denote by K. (The optimal sequence of products can be found by solving a traveling salesman problem, where the "distance" between "cities" i and j is given by the setup cost incurred when switching from product i to j.) For notational convenience, we simply assume that a setup cost K/N is imposed whenever the server switches from one product to another.

The scheduling policy determines the inventory process $\tilde{I}_i$ and the counting process J, where $J(t)$ denotes the cumulative number of setups incurred by the server up to time t. (Although the scheduling policy, and hence the stochastic processes $\tilde{I}_i$ and J, can be rigorously defined in terms of a sequence of starting and stopping times for each product, we omit this detailed specification because it is not required in our subsequent analysis.) Our problem is to find a scheduling policy, which is nonanticipating with respect to $\tilde{I}_i$, to minimize

$$\limsup_{T\to\infty} \frac{1}{T} E\left[\int_0^T \sum_{i=1}^N (\bar{h}_i \tilde{I}_i^+(t) + \bar{b}_i \tilde{I}_i^-(t))dt + \frac{K}{N} J(T)\right], \tag{1}$$

where $x^+ = \max(x, 0)$ and $x^- = -\min(x, 0)$.

1.2. The Heavy Traffic Normalizations

To prove a heavy traffic limit theorem, one typically defines a sequence of control problems indexed by n, where the traffic intensity of the nth system approaches unity as $n \to \infty$. However, we will not prove a limit theorem here; inspired by previous limit theorems, we instead consider one control problem with a fixed large value of n, where the server must be busy most of the time to satisfy average demand. We define the positive constant $\theta = \sqrt{n}(1 - \rho)$, but our recommended policy will turn out to be independent of the heavy traffic scaling parameter n and the related constant θ.

Let the process $\tilde{W}_i$ denote the workload content embedded in the finished goods inventory $\tilde{I}_i$. If $\tilde{I}_i(t)$ is positive then $\tilde{W}_i(t)$ represents the sum of the service times for units in inventory; if $\tilde{I}_i(t)$ is negative then $\tilde{W}_i(t)$ represents the amount of time that a continuously busy server requires to clear product i's backlog. The process $\tilde{W}_i$ will be referred to as the *workload* process for product i. The standard diffusion scaling will be employed to define the normalized inventory process $I_i(t) = \tilde{I}_i(nt)/\sqrt{n}$ and the normalized workload process $W_i(t) = \tilde{W}_i(nt)/\sqrt{n}$ (throughout the remainder of the paper, a tilde ($\sim$) denotes the original unscaled version of the process). Although the workload process $\tilde{W}_i$ is not observable by the scheduler when inventory is backordered, the normalized workload process W_i is more convenient for analysis than the normalized inventory process I_i. The linear identity $I_i \approx \mu_i W_i$ holds for a wide variety of queueing systems in the heavy traffic limit (i.e., as $n \to \infty$), and we assume that it holds for our specific (i.e., fixed n) system. This assumption allows us to translate the solution of the heavy traffic control problem into a scheduling policy that is expressed in terms of the unscaled inventory process $(\tilde{I}_1, \ldots, \tilde{I}_N)$.

As is standard for heavy traffic optimization problems, we normalize the costs to account for distortions in the relative magnitudes of the inventory and setup costs that result from the diffusion scaling. The appropriate normalization (see Reiman and Wein for details) is to reduce the setup cost K by a factor of n relative to the inventory costs, and consequently we leave the inventory costs $\bar{h}_i$ and $\bar{b}_i$ unscaled, and define the normalized setup cost $k = K/n$. Thus, the heavy traffic conditions for the setup cost problem require the server to be busy most of the time, and the setup cost K to be large.

1.3. A Preliminary Heavy Traffic Result

A recent heavy traffic result obtained by Coffman et al. (1995) provides the basis for our analysis. This result considers a traditional multiclass, single-server queueing system with renewal arrival processes, general service time distributions and no setup times. A cyclic exhaustive polling mechanism is employed: The server serves each class to exhaustion and then switches to the next class. The authors prove a heavy traffic limit theorem for the $N = 2$ case, and conjecture that it holds for $N > 2$; we describe the result for $N \geqslant 2$. Let us reuse the service time notation μ_i and c_{si}, and allow the demand parameters λ_i and c_{di} in the SELSP to also characterize the arrival process to this queueing system. If we let $\tilde{V}_i$ denote the workload for class i in the queue, then limit theorems in Iglehart and Whitt (1970) and Reiman (1988) imply that for large n the normalized total workload $\sum_{i=1}^N \tilde{V}_i(nt)/\sqrt{n}$ is well approximated under heavy traffic conditions by a reflected Brownian motion V on $[0, \infty)$ with drift $-\theta = \sqrt{n}(\rho - 1)$ and variance

$$\sigma^2 = \sum_{i=1}^N \frac{\lambda_i}{\mu_i^2}(c_{di}^2 + c_{si}^2). \tag{2}$$

See Harrison (1985) for a definition of this process, which we denote by RBM $(-\theta, \sigma^2)$.

Let us define the normalized workload processes $V_i(t) = \tilde{V}_i(nt)/\sqrt{n}$ and the constants $\hat{\rho}_i = \rho_i(1 - \rho_i)$ and $\hat{\rho} = \sum_{i=1}^N \hat{\rho}_i$. The result by Coffman et al. (1995) is a *heavy traffic averaging principle* (abbreviated hereafter by HTAP) that provides information about the multidimensional workload process. It states that if the traffic intensity ρ is close to one and the variance of the arrival processes and service processes are bounded then for any continuous function f and any time $T > 0$,

$$\int_0^T f(V_i(t))\,dt \text{ is well approximated by}$$

$$\int_0^T \left[\int_0^1 f\left(\frac{2\hat{\rho}_i}{\hat{\rho}} V(t)u\right) du\right] dt, \tag{3}$$

where V is the RBM defined previously. The heavy traffic limit in Coffman et al. (1995) states that the left side of Equation (3) converges to the right side as $n \to \infty$ for the two-queue system. The two integrals on the right side of Equation (3) can be interpreted as a *time scale decomposition* of the system: On the time scale giving rise to the total workload RBM V, the individual workloads V_i move infinitely quickly (in the asymptotic limit) and can be represented as a uniform distribution over the path followed by the individual workloads. This scale captures gross changes in the total inventory over long periods of time. If we now expand time by a factor of $\sqrt{n}$ and consider $\bar{V}_i = V_i(\sqrt{n}t)/\sqrt{n}$, then a fluid scaling is obtained; this scale captures the movements of individual inventories over shorter time intervals. At this faster time scale, the total workload RBM V remains fixed (in the heavy traffic limit) and the individual workloads $\bar{V}_i$ evolve at a finite rate in a deterministic fashion. For a given total workload V, the individual fluid workloads form an N-dimensional process $(\bar{V}_1, \ldots, \bar{V}_N)$ that moves along a constant-workload, piecewise-linear path connecting points where the server exhausts a product. In the two-product case, the path is the line segment from $(0, V)$ to $(V, 0)$; in the case of three identical products (with the same service and demand rates), the path consists of the line segments connecting the points $(0, V/3, 2V/3)$, $(V/3, 2V/3, 0)$, and $(2V/3, 0, V/3)$.

Guided by these results, which are known to hold for a two-class queue employing an exhaustive polling scheme, we make the crucial assumption that the *HTAP holds for our N-product SELSP for all dynamic cyclic policies*. That is, we assume that an approximation like that in Equation (3) holds not just for the cyclic exhaustive policy (which corresponds to a cyclic base stock policy in the SELSP problem) in a make-to-order environment, but for more general cyclic policies in a make-to-stock setting.

1.4. An Overview of the Analysis

It is useful to view the control policy as consisting of two interrelated decisions: a busy/idle policy and a dynamic lot-sizing policy that specifies what the server should do while working. We begin by characterizing the busy/idle policy. Our HTAP assumption and the well-known relationship between queueing systems and production/inventory systems (e.g., Morse 1958) imply that the system state of the heavy traffic control problem is the one-dimensional total workload process $W = \sum_{i=1}^{N} W_i$, which measures the total machine time embodied in the current finished goods inventory. Furthermore, since setup times are zero, the total workload process is only affected by the server's busy/idle policy, not by how often the server switches among products. Because inventory costs become unbounded as $W(t) \to \infty$, the only reasonable form of the optimal busy/idle policy is for the server to stay busy if $W(t) < w_0$ and to idle if $W(t) \geqslant w_0$, for the unspecified control parameter w_0 (see Wein 1992 for a proof of optimality for a related, but simpler, cost structure). The quantity w_0 will often be referred to as the *idling threshold*, and can be viewed as an *aggregate base stock level*. Our HTAP assumption and the one-to-one relationship between queueing systems and production/inventory systems imply that the total workload process W is a RBM (θ, σ^2) on $(-\infty, w_0]$ under this busy/idle policy.

Although the lot-sizing policy does not influence the total workload W, it does affect the rate at which inventory costs and setup costs are incurred when $W(t) = w$. The HTAP assumption allows us to use a two-step procedure, where each step is performed at a different time scale, to find the optimal dynamic cyclic policy. In the first step, we find the lot-sizing policy that minimizes the average (inventory plus setup) cost incurred as the individual fluid inventory levels oscillate deterministically while the total diffusion workload W remains constant at w; let us call the resulting minimum cost $c(w)$. By solving a family of deterministic optimization problems indexed by the total workload w, we are able to construct a dynamic (i.e., workload-dependent) lot-sizing policy. In the second step, we find the aggregate base stock level w_0 that minimizes the long run average cost

$$\int_{-\infty}^{w_0} c(w) \frac{2\theta}{\sigma^2} e^{-2\theta(w_0 - w)/\sigma^2}\,dw, \tag{4}$$

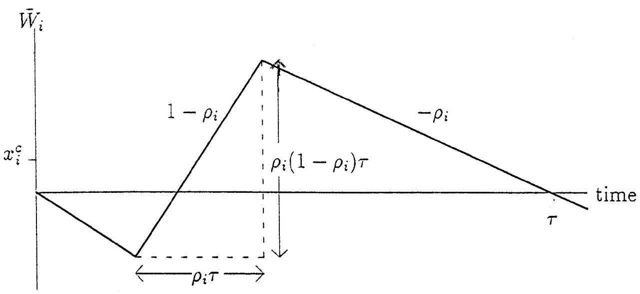
where we have used the fact (e.g., Harrison 1985) that RBM $(-\theta, \sigma^2)$ on $[0, \infty)$ has an exponential stationary distribution with parameter $2|\theta|/\sigma^2$.

In §1.5, we calculate the average cost incurred by a generic dynamic cyclic policy when the total workload equals w. The optimal heavy traffic cyclic policy is found in §1.6. Finally, the heavy traffic normalizations are reversed in §1.7 to obtain the proposed scheduling policy as a function of the unscaled inventory levels and the server location.

1.5. Construction of Dynamic Cyclic Policies

The goal of this subsection is to find the cost associated with any dynamic cyclic policy when the total workload $W(t) = w$. We shall detail this calculation with the aid of our HTAP assumption. Starting with the heavy traffic normalization $W_i(t) = \tilde{W}_i(nt)/\sqrt{n}$, we slow down time by a factor of $\sqrt{n}$ to obtain a fluid scaling, $\bar{W}_i(t) = \tilde{W}_i(\sqrt{n}t)/\sqrt{n}$. At this time scaling, the process $W(t) = \tilde{W}(nt)/\sqrt{n}$ is fixed at the value w, and the N-dimensional workload $\bar{W}_i$ moves at a finite rate in a deterministic manner. Because idleness is only incurred when the total workload reaches a certain base

Figure 1. Workload fluctuation over a cycle.



stock level, no idleness is incurred during a cycle and the process traverses the same path repeatedly, once per cycle, thereby generating a closed loop.

A cyclic policy (or, equivalently, the closed loop generated by the policy) will be defined by $N+1$ quantities: the *cycle length* τ and the *cycle center* $x^c=(x_1^c,\ldots,x_N^c)$. These control parameters are actually functions of the total workload w, but this dependence will be suppressed for improved readability. The cycle center x_i^c is the average amount of product i's inventory over the course of a cycle. Because the transient effects associated with initiating or temporarily moving a cycle vanish in the heavy traffic time scaling, the cycle center x^c can be placed anywhere in the constant workload hyperplane for each total workload level w.

We begin by examining the deterministic behavior of the individual product workload levels $\bar{W}_i$ under a cyclic policy when $W(t)=w$. For the system to remain balanced, the amount of each product produced per cycle must equal the amount demanded, and hence each product must be produced a fraction ρ_i of the time; we assume that ρ equals one throughout this fluid analysis, so that the server is busy throughout the cycle. Thus, for an arbitrary instantaneous total workload w and cycle length τ, each product i must be serviced for $\rho_i\tau$ units of time per cycle. Therefore, when the machine is servicing product i, the work content in this product's inventory is depleted at rate ρ_i and is replenished at rate one, and $\bar{W}_i$ increases at the fixed rate $1-\rho_i$ for $\rho_i\tau$ units of time per cycle. For the remaining $(1-\rho_i)\tau$ time units in the cycle when product i is not being produced, the workload inventory is decreasing at rate ρ_i. To uniquely determine the behavior of a cyclic policy, a reference starting point also needs to be specified. We use x_i^c, product i's average inventory level, as the reference point. Figure 1 illustrates these notions.

The cost of a cyclic policy can be expressed in terms of the cycle length τ, the cycle center x^c (an N-dimensional vector) and the total workload w. To find the average inventory cost for product i per unit time, $c_i(\tau,x_i^c,w)$, we integrate the inventory cost over the cycle and divide by the cycle length. The average cost breaks down into three regions depending on whether the product is entirely held throughout the cycle, is held and backordered or is entirely backordered. If we define the constant $r_i=\hat{\rho}_i/(b_i+h_i)$ for $i=1,\ldots,N$ then Figure 1 implies that

$$c_i(\tau,x_i^c,w) = \begin{cases} h_i x_i^c & \text{if } x_i^c>\tau\hat{\rho}_i/2,\\ \frac{(b_i+h_i)\tau\hat{\rho}_i}{8}+\frac{(x_i^c)^2}{2\tau r_i} + \frac{h_i-b_i}{2}x_i^c & \text{if } 0\in[x_i^c\pm\tau\hat{\rho}_i/2],\\ -b_i x_i^c & \text{if } x_i^c<-\tau\hat{\rho}_i/2. \end{cases} \tag{5}$$

The total average cost, $c(\tau,x^c,w)$, which includes inventory and setup costs, is

$$c(\tau,x^c,w)=\sum_{i=1}^N c(\tau,x_i^c,w)+\frac{k}{\tau}. \tag{6}$$

1.6. The Optimal Dynamic Cyclic Policy

Our cyclic policy consists of three controls: the aggregate base stock level w_0 and, for each total workload level w, the cycle length τ and the N-dimensional cycle center x^c. The optimal dynamic cyclic policy is derived in three stages: (i) Find the optimal cycle center x^c in terms of arbitrary w and τ; (ii) Optimize over the cycle length τ in terms of an arbitrary w; and (iii) Substitute the derived cost function $c(w)$ into Equation (4) and find the optimal idling threshold w_0. The first two stages can be performed by equating the derivative of the cost Equation (6) to zero and solving for the cycle center and cycle length. The details of the calculations are largely algebraic and can be found in the appendix; we only summarize the results here.

The solution to the optimal cycle center and cycle length has different forms depending on the total workload level. The total workload is divided into three regions: Region I is characterized by workload levels $w>w_1$, region II by $w_2\leqslant w\leqslant w_1$, and region III by $w<w_2$. If we let

$$\xi_1=\sum_{i=1}^{N-1}\frac{(b_i+h_i)\hat{\rho}_i}{8}-\frac{r_i}{2}\left(\frac{b_i-h_i}{2}+h_N\right)^2, \tag{7}$$

$$\xi_2=\frac{1}{8}\sum_{i=1}^N r_i\left[(b_i+h_i)^2-(b_i-h_i)^2 + \left(\frac{\sum_{j=1}^N(b_j-h_j)r_j}{\sum_{j=1}^N r_j}\right)^2\right], \tag{8}$$

$$\xi_3=\sum_{i=1}^{N-1}\frac{(b_i+h_i)\hat{\rho}_i}{8}-\frac{r_i}{2}\left(\frac{b_i-h_i}{2}-b_N\right)^2 \text{ and}$$

$$\xi_4=\frac{1}{2\sum_{i=1}^N r_i}, \tag{9}$$

then

$$w_1=\sqrt{\frac{\xi_2-\xi_1}{\xi_4\xi_1}k} \quad\text{and}\quad w_2=-\sqrt{\frac{\xi_2-\xi_3}{\xi_4\xi_3}k}. \tag{10}$$

The optimal cycle center in terms of cycle length τ is given below for the three regions. For $i<N$ we have

$$x_i^{c*}=\begin{cases}\tau r_i[\frac{b_i-h_i}{2}+h_N] & \text{in region I,}\\ \frac{\tau r_i}{2}(b_i-h_i-2\xi_4 \\ \quad \sum_{j=1}^{N}(b_j-h_j)r_j) & \\ \quad +2w\xi_4 r_i & \text{in region II,}\\ \tau r_i[\frac{b_i-h_i}{2}-b_N] & \text{in region III.}\end{cases} \quad (11)$$

The last component, x_N^{c*}, is equal to $w-\sum_{i=1}^{N-1}x_i^{c*}$. In region II, x_N^{c*} has the same form as the other products' cycle center, as given in Equation (11).

The optimal cycle length, τ^*, is given by

$$\tau^*=\begin{cases}\sqrt{\frac{k}{\xi_1}} & \text{in region I,}\\ \sqrt{\frac{\xi_4 w^2+k}{\xi_2}} & \text{in region II,}\\ \sqrt{\frac{k}{\xi_3}} & \text{in region III.}\end{cases} \quad (12)$$

The defining characteristic of this three-region breakdown is the relation between x_N^{c*} and τ^*. Product N's cycle center satisfies $x_N^{c*}>\tau^*\hat{\rho}_N/2$ in region I, $|x_N^{c*}|<\tau^*\hat{\rho}_N/2$ in region II, and $x_N^{c*}<-\tau^*\hat{\rho}_N/2$ in region III. The other $N-1$ products satisfy $|x_i^{c*}|<\tau^*\hat{\rho}_i/2$ for all three regions.

By (4) and (6), the total average cost under the optimal lot-sizing portion of the dynamic cyclic policy is given by

$$\int_{-\infty}^{w_0}\left(\sum_{i=1}^{N}c_i(\tau^*,x_i^{c*},w)+\frac{k}{\tau^*}\right)\frac{2\theta}{\sigma^2}e^{-\frac{2\theta}{\sigma^2}(w_0-w)}dw. \quad (13)$$

Having solved for τ^* in terms of the system parameters and w, we can substitute (12) into (11) to get x^{c*} in terms of w. Substituting this expression into (5) allows us to express the total average cost in (6) solely in terms of system parameters and total workload w; we will not explicitly write out this expression because it adds little to our understanding of the problem.

Equation (13) can be calculated. Depending on the idling threshold w_0, the integral can be broken into three or fewer terms based on the division of the total workload into the three regions defined in (10). Because of the irregular form of τ^* for $w\in[w_2,w_1]$, the integral over region II has no closed-form representation. Therefore, we resort to numerical methods to determine the optimal value of w_0, which is denoted by w_0^*.

Our derivation of the optimal dynamic cyclic policy in heavy traffic is now complete. Recall that when the problem was initially defined, we assume that product N had both the smallest holding and backorder cost indices. The analysis would remain unchanged, however, without this restriction. One needs only to introduce new notation (for example, $N-1$) to designate the product with the lowest backorder cost index, and use product $N-1$ in place of product N when the total workload level is less than w_2.

1.7. The Proposed Policy

The final step in our analysis is to employ the optimal heavy traffic policy derived in §1.6 to develop a proposed policy for the original SELSP. This is done in two stages: We reverse the heavy traffic scalings to express the solution in terms of the original problem parameters, and then interpret the resulting solution.

If we replace the normalized quantities w,k and τ by $\tilde{w}/\sqrt{n}$, K/n, and $\tilde{\tau}/\sqrt{n}$, respectively (τ undergoes this normalization because time is compressed by $\sqrt{n}$ in the fluid model), then regions $\tilde{\text{I}}$, $\tilde{\text{II}}$, and $\tilde{\text{III}}$ are defined by (10), with $\tilde{w}_i$ and K in place of w_i and k. The unscaled cycle center $\tilde{x}_i^{c*}$ for $i<N$ is defined by the right side of (11), with $\tilde{\tau},\tilde{w}$, $\tilde{\text{I}}$, $\tilde{\text{II}}$, and $\tilde{\text{III}}$ replacing τ,w, I, II, and III, respectively, and $\tilde{x}_N^{c*}$ is equal to $\tilde{w}-\sum_{i=1}^{N-1}\tilde{x}_i^{c*}$. The optimal cycle length $\tilde{\tau}^*$ is given by the right side of (12), with $\tilde{w},K$, $\tilde{\text{I}}$, $\tilde{\text{II}}$, and $\tilde{\text{III}}$ in place of w,k, I, II, and III. Hence, the unnormalized average inventory cost, $\tilde{c}_i(\tilde{\tau}^*,\tilde{x}^{c*},\tilde{w})$, is equal to $c_i(\tilde{\tau}^*,\tilde{x}^{c*},\tilde{w})$.

Therefore, if w_0^* minimizes Equation (13), then $\tilde{w}_0^*=\sqrt{n}w_0^*$ will minimize the unnormalized long run average cost expression

$$\int_{-\infty}^{\tilde{w}_0}\left(\sum_{i=1}^{N}\tilde{c}_i(\tilde{\tau}^*,\tilde{x}_i^{c*},\tilde{w})+\frac{K}{\tilde{\tau}^*}\right) \cdot\frac{2(1-\rho)}{\sigma^2}e^{-\frac{2(1-\rho)}{\sigma^2}(\tilde{w}_0-\tilde{w})}d\tilde{w}. \quad (14)$$

Notice that the heavy traffic parameter n does not appear in (14). To compute $\tilde{w}_0^*$ in §3, we use Maple V to numerically solve the first-order optimality conditions associated with (14).

Our proposed dynamic cyclic policy for the SELSP must be expressed in terms of the original $(N+1)$-dimensional system state, which is given by the current inventory levels $\tilde{I}_1(t),\ldots,\tilde{I}_N(t)$ and the server location. There are many ways in which the unnormalized policy $(\tilde{x}_i^{c*},\tilde{\tau}^*,\tilde{w}_0^*)$ can be interpreted for purposes of implementation. Perhaps the most natural way to express a dynamic lot-sizing policy is to specify a state-dependent maximum inventory (or "produce-up-to") level for the product currently being produced. By Figure 1, when the unnormalized total workload level $\tilde{W}(t)$ equals $\tilde{w}$, the maximum workload level for product i is $\tilde{x}_i^{c*}+\hat{\rho}_i\tilde{\tau}^*/2$, which occurs when the production of product i is finished. Making use of the heavy traffic identity $I_i\approx\mu_i W_i$, let us define the three unnormalized workload regions in terms of the inventory process and the thresholds $\tilde{w}_1$ and $\tilde{w}_2$: Region $\tilde{\text{I}}$ is $\sum_{i=1}^{N}\mu_i^{-1}\tilde{I}_i(t)>\tilde{w}_1$, region $\tilde{\text{II}}$ is $\sum_{i=1}^{N}\mu_i^{-1}\tilde{I}_i(t)\in[\tilde{w}_2,\tilde{w}_1]$, and region $\tilde{\text{III}}$ is $\sum_{i=1}^{N}\mu_i^{-1}\tilde{I}_i(t)<\tilde{w}_2$. Then our proposed policy can be described as follows: *If* $\sum_{i=1}^{N}\mu_i^{-1}\tilde{I}_i(t)>0$ *then let* N *refer to the product with the smallest holding cost index* h_i; *otherwise, let* N *denote the product with the smallest value of* b_i. *The server should idle if* $\sum_{i=1}^{N}\mu_i^{-1}\tilde{I}_i(t)>\tilde{w}_0^*$;

otherwise, if set up for product $i<N$ *then produce this product as long as*

$$\mu_i^{-1}\tilde{I}_i(t) < \begin{cases} \sqrt{\frac{K}{\xi_1}}(b_i + h_N)r_i, & \tilde{\mathrm{I}}, \\ \sqrt{\frac{\xi_4(\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t))^2+K}{\xi_2}}\,r_i \\ \quad\cdot(b_i - \xi_4\sum_{j=1}^N (b_j - h_j)r_j) \\ \quad +2\xi_4 r_i\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t) & \tilde{\mathrm{II}}, \\ \sqrt{\frac{K}{\xi_3}}(b_i - b_N)r_i, & \tilde{\mathrm{III}}. \end{cases} \tag{15}$$

Once $\mu_i^{-1}\tilde{I}_i(t)$ *reaches or exceeds this level, switch to the next product. If setup for product* N, *then produce this product while*

$$\mu_N^{-1}\tilde{I}_N(t) < \begin{cases} \sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) \\ \quad +\sqrt{\frac{K}{\xi_1}}\Big(\frac{\hat{\rho}N}{2} - \sum_{i=1}^{N-1} r_i \\ \quad \cdot\big[\frac{b_i - h_i}{2} + h_N\big]\Big), & \tilde{\mathrm{I}}, \\ \sqrt{\frac{\xi_4(\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t))^2+K}{\xi_2}}\,r_N \\ \quad\cdot(b_N - \xi_4\sum_{j=1}^N (b_j - h_j)r_j) \\ \quad +2\xi_4 r_N\sum_{j=1}^N \mu_j^{-1}\tilde{I}_j(t) & \tilde{\mathrm{II}}, \\ \sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t) \\ \quad +\sqrt{\frac{K}{\xi_3}}\Big(\frac{\hat{\rho}N}{2} - \sum_{i=1}^{N-1} r_i \\ \quad \cdot\big[\frac{b_i - h_i}{2} - b_N\big]\Big), & \tilde{\mathrm{III}}, \end{cases} \tag{16}$$

and then switch to the next product when $\mu_N^{-1}\tilde{I}_N(t)$ *reaches or exceeds this level.*

Our results simplify considerably under the *cost-symmetric* case, where $h_i = h$ and $b_i = b$ for all $i = 1, \ldots, N$. This case will arise, for example, if all products are relatively indistinguishable, except for their color. Then $\xi_1 = \xi_3 = 0$ and, by (10), the workload always resides in region II. When the server is busy and is set up for product i, this product is produced as long as

$$\mu_i^{-1}\tilde{I}_i(t) < \hat{\rho}_i \left[\sqrt{\frac{(\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t))^2}{\hat{\rho}^2} + \frac{2K}{\hat{\rho}(b+h)}} + \frac{\sum_{i=1}^N \mu_i^{-1}\tilde{I}_i(t)}{\hat{\rho}} \right]. \tag{17}$$

It is evident from this expression that the cost structure effects the lot sizes only through the ratio $K/(b+h)$; not surprisingly, the lot size is an increasing function of this quantity.

1.8. Discussion

Our analysis reveals several insights into the behavior of the optimal policy in heavy traffic. (Before reading this subsection, readers may want to digest the graphs marked "proposed" in Figures 3 and 4 in §3.2, which depict the proposed policies in both the symmetric (each product has the same parameters) and asymmetric two-product settings.)

Three Workload Regions. An essential feature of the heavy traffic policy is its characterization via three workload regions, as described in (10). There is substantial inventory in region I, significant backorders in region III, and region II represents the intermediate case where the total workload is in an interval containing zero.

State-Dependent Lot Sizes. Because the time spent producing product i within a cycle is $\rho_i\tau$, the optimal cycle length τ^* determines the optimal lot sizes in heavy traffic (and determines the optimal *expected* lot sizes for the SELSP). We can observe from (12) that the optimal lot sizes are state-dependent when the total workload is in region II. In contrast, the lot sizes are constant in regions I and III; in these regions, surplus or deficit inventory is unavoidable, and the trade-off between lot sizes and setup costs stabilize, thereby generating constant lot sizes. This observation and (11) imply that the optimal cycle center x^{c*} remains constant in regions I and III, and gradually shifts between these two points in the intermediate area of region II. It is worth pointing out that in nearly all of the deterministic ELSP literature (Dobson 1987 is a notable exception), the analysis is restricted to policies with constant lot sizes.

Relationship to the EOQ Model. As in the economic order quantity (EOQ) model, the lot size in (12) is proportional to the square root of the setup cost in regions I and III. In region II, the setup cost again appears in the numerator of the square root term.

Inventory is Focused in the Least Cost Products. In region I, excess inventory is built up in the product with the smallest h_i, which is a product that is inexpensive to hold (small $\bar{h}_i$) and lengthy to process (small μ_i). Similarly, in region III, excess negative inventory (i.e. backorders) is held in the product with the smallest backorder cost index b_i; this product is inexpensive to backorder and has a long expected processing time. In both regions, inventory is held in the least cost product to reduce the absolute value of the inventory of the higher (holding in region I and backorder in region III) cost products. In this regard, the dynamic lot-sizing policy derived here is similar to the heavy traffic policy derived for the corresponding problem without setups in Wein (1992), in which instantaneous switching causes the inventories of all the higher cost products to vanish in the heavy traffic normalization. When setup costs are introduced, breadth is added to the normalized cycle length and, for a fixed total workload, a "corridor" of possible inventory states replaces the least cost axes. In fact, if we consider the special case $K=0$, then region I (region III) corresponds to $w>0$ ($w<0$); in both regions, $\tau^*=0$ and $x_N^{Nc}=w$, and the solution reverts to that of Wein (1992).

Lot Sizes Grow with Absolute Value of Total Workload. By (12), we see that the optimal lot size is smallest when the total workload equals zero, and grows with the absolute value of the workload. When the total workload is near zero, costly backorders can be avoided by switching frequently between products. In contrast, when the absolute value of the workload is large, it is possible to employ large lot sizes without adversely affecting the inventory costs (because inventory tends to be held in the minimum cost product in regions I and III); in this case, it is advantageous to avoid setup costs and produce products in large batches.

Inventory Levels at Switching Epochs. In heavy traffic, the maximum normalized workload for product i under our proposed policy is $x_i^{c*}+\hat{\rho}_i\tau^*/2$. It follows that for $i<N$, product i inventory is negative (i.e., backordered) when the server switches into product i and is positive when the server switches out of product i. For product N the sign of the inventory level during the switching epochs depends on the region: In region II product N inventory is negative when the server switches into product N and is positive when the server switches out of product N. In contrast, product N inventory is always positive in region I and is always negative in region III. Moreover, it can be shown that the maximum normalized workload for product i is a non-decreasing function in total workload w for systems with only two products or for the cost-symmetric case. In more general cases, this quantity can both increase and decrease in region II.

2. THE SETUP TIME PROBLEM

2.1. Problem Description

In the setup time problem, a random setup time rather than a setup cost) is incurred when the server switches from one product to another. In all other respects, the setup cost problem and the setup time problem are identical, and all relevant notation from §1 will be retained. As in the setup cost problem, we shall consider only dynamic cyclic policies. The server has three scheduling options at each point in time: Produce a unit of the product that is currently set up, initiate a setup for the next product in the cycle, or sit idle. The control problem is to find a nonanticipating scheduling policy to minimize

$$\limsup_{T\to\infty}\frac{1}{T}E\left[\int_0^T\sum_{i=1}^N(\bar{h}_i\tilde{I}_i^+(t)+\bar{b}_i\tilde{I}_i^-(t))\,dt\right]. \tag{18}$$

2.2. The Diffusion Control Problem

We use the same heavy traffic normalizations as in the setup cost problem. As in Coffman et al. (1998), we do not normalize the setup times in heavy traffic. Coffman et al. (1998) derive a HTAP for a two-class queue with setup times under an exhaustive polling mechanism. They show that the HTAP in Equation (3) still holds, but with V on the right side of this equation being a diffusion process with variance σ^2 and a state-dependent drift, rather than a $(-\theta,\sigma^2)$ Brownian motion. The drift is the limit of $\sqrt{n}(\rho-f(v))$ as $n\to\infty$, where $f(v)$ is the fraction of time during a cycle that the server serves customers, as opposed to incurring setups, when the total workload equals v. Because the setup times are unscaled, they occur instantaneously in the heavy traffic limit, and so, for a given total workload, the fluid trajectories of the individual queues are identical to those in the setup cost problem.

As in §2, we make the key assumption that a HTAP corresponding to (3) holds for all dynamic cyclic policies in our dynamic N-product SELSP. Under this assumption, when the normalized workload in our SELSP equals w and the normalized cycle length of the dynamic cyclic policy is $\tau(w)$, the fraction $f(w)$ equals $\sqrt{n}\tau(w)/(\sqrt{n}\tau(w)+s)$, where s is the mean setup time per cycle. Although we consider a fixed value of the heavy traffic parameter n, we assume that W has the state-dependent drift $\theta-s/\tau(w)$ because $\sqrt{n}(f(w)-\rho)\to\theta-s/\tau(w)$ as $n\to\infty$. (Under the HTAP assumption, the performance of the SELSP depends upon the setup time distributions only via the mean setup time per cycle, so the desired product sequence within a cycle is characterized by the traveling salesman tour, where the intercity distances are given by the mean setup times between products.)

Because the HTAP assumption implies that the N-dimensional fluid process $(\bar{W}_1,\ldots,\bar{W}_N)$ is identical to the corresponding process in the setup cost problem for a given total normalized workload, the optimal cycle center x^{c*} is given by Equation (11). The resulting inventory cost rate $c_i(\tau(w),w)$ is derived by substituting x^{c*} into (5), and can be found in Equations (52)–(53) in Markowitz et al. (1997, Appendix) (hereafter referred to as MRW).

If we define the cost function $c(\tau(w),w)=\sum_{i=1}^N c_i(\tau(w),w)$, then the approximating diffusion control problem is to choose the state-dependent cycle length $\tau(w)\geqslant 0$ and the threshold w_0 to minimize

$$\limsup_{T\to\infty}\frac{1}{T}E\left[\int_0^T c(\tau(W(t)),W(t))\,dt\right], \tag{19}$$

where W is a $(\theta-s/\tau(w),\sigma^2)$ reflected diffusion process on $(-\infty,w_0]$. Hence, the controllable cycle length $\tau(w)$ affects both the drift and the cost $c(\tau,w)$ in a nonlinear fashion.

2.3. The Optimality Conditions

Problem (19) involves a drift control $\tau(w)$ and a singular control via the reflecting barrier w_0. Although the drift is unbounded as $\tau(w)\to 0$, we proceed as if standard arguments apply (Mandl 1968, p. 159) and state the Hamilton–Jacobi–

Bellman optimality conditons:

$$\min_{\tau(w)\geqslant 0}\left\{c(\tau(w),w)-g+\left(\theta-\frac{s}{\tau(w)}\right)V'(w)+\frac{\sigma^2}{2}V''(w)\right\}=0 \quad \text{for } w\leqslant w_0, \tag{20}$$

and

$$V'(w)=0 \quad \text{for } w\geqslant w_0, \tag{21}$$

where g is the gain and $V(x)$ is the potential (relative value) function. We make two assumptions in our analysis of problem (19): First, we assume that a solution to (20)–(21) yields a solution to (19); we also assume that $V\in C^2$ and define $P(w)=V'(w)$; this assumption is known as the heuristic principle of smooth fit (Beneš et al. 1980) and is often imposed when solving diffusion control problems.

Using the workload cutoff levels w_1 and w_2 to distinguish among regions I, II, and III (w_1 and w_2 are unknown at this point and are not given by (10)), we substitute the three forms of cost function $c(\tau,w)$ into (20) to obtain

$$\min_{\tau(w)\geqslant 0}\left\{\xi_1\tau(w)+h_N w-g+\left(\theta-\frac{s}{\tau(w)}\right)P(w)+\frac{\sigma^2}{2}P'(w)\right\}=0 \quad \text{for } w_1\leqslant w\leqslant w_0, \tag{22}$$

$$\min_{\tau(w)\geqslant 0}\left\{\xi_2\tau(w)+\xi_5 w+\xi_4\frac{w^2}{\tau(w)}-g+\left(\theta-\frac{s}{\tau(w)}\right)P(w)+\frac{\sigma^2}{2}P'(w)\right\}=0$$
$$\text{for } w_2\leqslant w\leqslant w_1, \tag{23}$$

$$\min_{\tau(w)\geqslant 0}\left\{\xi_3\tau(w)-b_N w-g+\left(\theta-\frac{s}{\tau(w)}\right)P(w)+\frac{\sigma^2}{2}P'(w)\right\}=0 \quad \text{for } w\leqslant w_2, \tag{24}$$

where the new constant ξ_5 is $(\sum_{i=1}^N (h_i-b_i)r_i)/(2\sum_{i=1}^N r_i)$.

Solving the first-order optimality conditions for $\tau(w)$ yields

$$\tau^*(w)=\begin{cases}\sqrt{-\frac{sP(w)}{\xi_1}} & \text{for } w_1\leqslant w,\\ \sqrt{\frac{\xi_4}{\xi_2}w^2-\frac{sP(w)}{\xi_2}} & \text{for } w_2\leqslant w\leqslant w_1,\\ \sqrt{-\frac{sP(w)}{\xi_3}} & \text{for } w\leqslant w_2.\end{cases} \tag{25}$$

Substituting $\tau^*(w)$ into (22)–(24), we obtain the nonlinear ordinary differential equations (ODEs)

$$2\sqrt{-\xi_1 sP(w)}+h_N w-g+\theta P(w)+\frac{\sigma^2}{2}P'(w)=0$$
$$\text{for } w_1\leqslant w\leqslant w_0, \tag{26}$$

$$2\sqrt{\xi_2\xi_4 w^2-\xi_2 sP(w)}+\xi_5 w+\theta P(w)+\frac{\sigma^2}{2}P'(w)-g=0 \quad \text{for } w_2\leqslant w\leqslant w_1, \tag{27}$$

$$2\sqrt{-\xi_3 sP(w)}-b_N w-g+\theta P(w)+\frac{\sigma^2}{2}P'(w)=0$$
$$\text{for } w\leqslant w_2. \tag{28}$$

Because the nonlinear ODEs in (26)–(28) do not appear to admit a closed-form solution, we resort to an algorithmic procedure for solving the diffusion control problem, which is briefly described in §2.6. In the next two subsections, we state and discuss several structural properties of the optimal solution.

2.4. Structural Properties

We refer readers to Markowitz (1996) for the derivations of the following properties. The derivation of Property 1, which is nearly identical to the derivation in Reiman and Wein (1998, Appendix), assumes that $\tau^*(w)$ is nondecreasing in $|w|$ for all sufficiently small w (i.e. as $w\to-\infty$); this assumption holds in all our numerical results.

PROPERTY 1. *If* $w_2=-\infty$ *then*

$$P(w)=\frac{2\sqrt{\xi_2\xi_4}-\xi_5}{c}w+o(w) \quad \textit{as } w\to-\infty;$$

if $w_2\neq-\infty$ *then*

$$P(w)=-\frac{b_N}{c}w+o(w) \quad \textit{as } w\to-\infty.$$

PROPERTY 2. *The idling threshold* w_0 *is greater than or equal to* w_1, *and equality holds if and only if* $h_i=h_j$ *for all i and j. At the idling threshold, the cycle length* $\tau^*(w_0)=0$ *if* $w_0>w_1$ *and* $\tau^*(w_0)=\sqrt{\xi_4/\xi_2}w_0$ *if* $w_0=w_1$.

PROPERTY 3. $w_2=-\infty$ *if and only if* $b_i=b_j$ *for all i and j*.

2.5. Discussion

Because a closed-form solution is not obtained for the setup time problem, it is more difficult to develop insights into the behavior of the optimal solution. Nevertheless, several noteworthy comparisons can be made.

Cost Symmetry vs. Cost Asymmetry. The solution for these two cases are surprisingly different (see the graphs entitled "proposed" in §3.3, Figures 5 and 6). In the cost-symmetric case the workload stays in region II, and the cycle length at the idling threshold, $\tau^*(w_0^*)$, is proportional to w_0^*. In the cost-asymmetric case, the policy is characterized by three regions, and the lot size approaches zero as the workload approaches the idling threshold; these results follow from Properties 2 and 3. Evidently, near the idling threshold, small lot sizes are used in the asymmetric case to reduce inventory costs, whereas this option is not available in the cost-symmetric case. Finally, by Properties 1 and 3 and Equation (25), as the current total workload w tends to $-\infty$, the cycle length grows with $\sqrt{-w}$ in the asymmetric case and with $-w$ in the symmetric case. Hence, when backorders are large, there is less opportunity to reduce inventory costs in the cost-symmetric case, and larger lot sizes prevail in order to reduce the amount of time devoted to setups.

Setup Costs vs. Setup Times. It is interesting to note both the similarities and differences between the setup cost and setup time problems: The behavior of the proposed policies for both problems can be distinctly broken down into three workload regions (one region, respectively) when costs are asymmetric (symmetric, respectively). For both problems, lot sizes are state-dependent and inventory is focused in the least cost products; moreover, the description of the inventory levels at the switching epochs in §1.8 carries over to the setup time problem. The proposed policies for the two problems are qualitatively similar in the region around $w=0$ (i.e., region II), but have different characteristics in the two extreme regions (regions I and III). In the asymmetric setup cost problem, the cycle length τ^* remains constant throughout these two regions. In the asymmetric setup time problem, the cycle length contracts to zero as w approaches the idling threshold and grows as $\sqrt{-w}$ when w tends to $-\infty$.

As noted in Reiman and Wein (1998), the two problems lead to qualitatively different solutions because queueing effects cause setup times to consume available capacity in a highly nonlinear manner. Therefore, *the effective cost of a setup time is workload-dependent in the setup time problem*: There is no direct penalty for a set up, only an increased probability that the total workload will fall. As the total workload approaches $-\infty$, many items are backordered and the effective cost of a setup is very high; thus, the scheduler attempts to use the capacity efficiently by running large lot sizes, so as to recover from the low workload level. In contrast, as the total workload approaches the idling threshold w_0^*, the effective cost of a setup time decreases, and the scheduler can afford to employ small lot sizes to reduce inventory costs. As a consequence of our analysis, it is clear that setup costs should not be used as a surrogate for setup times in the SELSP; unfortunately, this practice is quite common in the deterministic ELSP literature. See Markowitz and Wein (1996) for an analysis of the SELSP with setup costs and setup times.

Setup Times vs. No Setup Times. Like the setup cost policy, the proposed setup time policy is a generalization of Wein (1992). As setup times vanish, all the inventory becomes stored only in product N. Using the symmetric cost results, the optimal idling threshold w_0^* goes to $-\ln(\frac{h}{b+h})\sigma^2/(2\theta)$, which is the same as that derived by Wein.

2.6. An Algorithmic Solution

Because problem (19) cannot be solved analytically, we use the *Markov chain approximation* technique described in Kushner and Dupuis (1992). This method systematically discretizes both time and the state space and approximates a diffusion control problem by a control problem for a finite state Markov chain. Weak convergence methods have been developed by Kushner and his colleagues to verify that the controlled Markov chain (and its corresponding optimal cost) approximates arbitrarily closely the controlled diffusion process (and its corresponding optimal cost).

With their approximation, we derive an optimal policy with a dynamic programming policy improvement algorithm. The algorithm yields values of $\tau^*(w)$ for a finite set of total workload levels w, the idling threshold w_0^* and the region-defining constants w_1 and w_2. It is worth emphasizing that the *computational complexity of the algorithmic approach is independent of the number of products*; this important fact is a result of the state space collapse inherent in the HTAP, which leaves us with a one-dimensional diffusion process, and our optimization of the cycle center in (11).

In our implementation of the algorithm (see Markowitz 1996 for a detailed description of the algorithm), we introduce a slight modification to the heavy traffic analysis to account for the fact that setup times do not vanish in the original problem. The cycle length $\tau(w)$ consists of the time devoted to processing and the time allocated to setups. In the fluid scaling, $s/\sqrt{n}$ units of time are spent setting up over the course of a cycle; although this quantity vanishes in the limit, we replace $\tau(w)$ by $\tau(w)+s/\sqrt{n}$ as an intended refinement. Beyond this modification, we refer the reader to Markowitz (1996) for a detailed description of this algorithm.

2.7. The Proposed Policy

The mapping from a diffusion control solution to a proposed policy is less straightforward when a numerical solution is obtained than when an analytical solution is derived. More specifically, the drift of the underlying diffusion process is $\sqrt{n}(1-\rho)-s/(\tau(w)+s/\sqrt{n})$, and a value of n must be chosen in order to compute a numerical solution to the Markov chain control problem. This quandary is dealt with in the most natural way: We set θ equal to one and let $n=(1-\rho)^{-2}$. Moreover, we set the finite difference interval used in the Markov chain approximation equal to $1/\sqrt{n}$, so that the discretization in the Markov chain corresponds to individual units of inventory in the original problem. Exploratory computations revealed that the parameters $(\tilde{x}_c^*, \tilde{\tau}^*, \tilde{w}_0^*)$ of the proposed policy were very insensitive to our choice of n.

Recall that we replaced the cycle length $\tau(w)$ by $\tau(w)+s/\sqrt{n}$ in the Markov chain algorithm. Therefore, in creating our proposed policy we employ the cycle center $x^c(\tau(w)+s/\sqrt{n}, w)$. The proposed policy for the setup cost case produced product i until its inventory reached $\tilde{x}_i^c(\tilde{\tau}(\tilde{w}), \tilde{w})+\tilde{\tau}(\tilde{w})\hat{\rho}_i/2$. In order for the expected total busy time in a cycle to be equal to $\tilde{\tau}(\tilde{w})$, the $s/\sqrt{n}$ term is not added in the second of the two terms in this expression, and we produce product i until its inventory reaches $\tilde{x}_i^c(\tilde{\tau}(\tilde{w})+s(1-\rho), \tilde{w})+\tilde{\tau}(\tilde{w})\hat{\rho}_i/2$.

The proposed policy is constructed just as in the setup cost problem. Define the unnormalized workload regions $\tilde{\mathrm{I}}$, $\tilde{\mathrm{II}}$, and $\tilde{\mathrm{III}}$ according to whether the quantity $(1-\rho)\sum_{i=1}^{N}\mu_i^{-1}\tilde{I}_i(t)$ is greater than w_1, in the interval $[w_2, w_1]$ or less than w_2, respectively. Then our proposed policy is: *If* $(1-\rho)\sum_{i=1}^{N}\mu_i^{-1}\tilde{I}_i(t)>0$, *then let* N *refer to the product with the smallest holding cost index* h_i; *otherwise, let* N *denote the product with the smallest value of* b_i. *The server should idle if* $(1-\rho)\sum_{i=1}^{N}\mu_i^{-1}\tilde{I}_i(t)>w_0^*$; *otherwise, if set up for product* $i<N$, *then produce this product as long as*

$$\mu_i^{-1}\tilde{I}_i(t)<\begin{cases}[\tau^*\left((1-\rho)\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\right)+s(1-\rho)] \\ \quad\cdot\frac{r_i}{1-\rho}\left[\frac{b_i-h_i}{2}+h_N\right] \\ \quad+\tau^*\left((1-\rho)\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\right)\frac{\hat{\rho}_i}{2(1-\rho)}, & \tilde{\mathrm{I}}, \\ \tau^*\left((1-\rho)\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\right) \\ \quad\cdot\frac{r_i}{1-\rho}\left(b_i-\xi_4\sum_{j=1}^{N}(b_j-h_j)r_j\right) \\ \quad+\frac{sr_i}{2}\left(b_i-h_i-2\xi_4\sum_{j=1}^{N}(b_j-h_j)r_j\right) \\ \quad+2\xi_4 r_i\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t), & \tilde{\mathrm{II}}, \\ [\tau^*\left((1-\rho)\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\right)+s(1-\rho)] \\ \quad\cdot\frac{r_i}{1-\rho}\left[\frac{b_i-h_i}{2}-b_N\right] \\ \quad+\tau^*\left((1-\rho)\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\right)\frac{\hat{\rho}_i}{2(1-\rho)}, & \tilde{\mathrm{III}},\end{cases} \tag{29}$$

Once $\mu_i^{-1}\tilde{I}_i(t)$ *reaches or exceeds this level, switch to the next product. If set up for product* N, *then produce this product while*

$$\mu_N^{-1}\tilde{I}_N(t)<\begin{cases}\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)+\tau^*\Big((1-\rho) \\ \quad\cdot\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\Big)\frac{\hat{\rho}_N}{2(1-\rho)} \\ \quad-\sum_{i=1}^{N-1}\left[\tau^*\left((1-\rho)\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\right)\right. \\ \quad\left.+s(1-\rho)\right]\frac{r_i}{1-\rho}\left[\frac{b_i-h_i}{2}+h_N\right], & \tilde{\mathrm{I}}, \\ \tau^*\left((1-\rho)\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\right) \\ \quad\cdot\frac{r_N}{1-\rho}\left(b_N-\xi_4\sum_{j=1}^{N}(b_j-h_j)r_j\right) \\ \quad+\frac{sr_N}{2}\left(b_N-h_N-2\xi_4\sum_{j=1}^{N}(b_j-h_j)r_j\right) \\ \quad+2\xi_4 r_N\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t), & \tilde{\mathrm{II}}, \\ \sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)+\tau^*\Big((1-\rho) \\ \quad\cdot\sum_{j=1}^{N},\mu_j^{-1}\tilde{I}_j(t)\Big)\frac{\hat{\rho}_N}{2(1-\rho)} \\ \quad-\sum_{i=1}^{N-1}\left(\tau^*\left((1-\rho)\sum_{j=1}^{N}\mu_j^{-1}\tilde{I}_j(t)\right)\right. \\ \quad\left.+s(1-\rho)\right)\frac{r_i}{1-\rho}\left[\frac{b_i-h_i}{2}-b_N\right], & \tilde{\mathrm{III}},\end{cases} \tag{30}$$

and then switch to the next product when $\mu_N^{-1}\tilde{I}_N(t)$ *reaches or exceeds this level.*

The solution proposed above is specified in terms of the original problem parameters, and the constants w_0^*, w_1, w_2 and the function $\tau^*(w)$ generated by the algorithmic procedure. Because $\tau^*(w)$ is defined only on a discrete state space, the argument of this function is rounded to the closest discrete value in the algorithmic discretization.

As with the setup cost problem, the *cost-symmetric case* for the setup time problem also simplifies; see MRW (1997, §2.7) for details.

3. COMPUTATIONAL STUDY

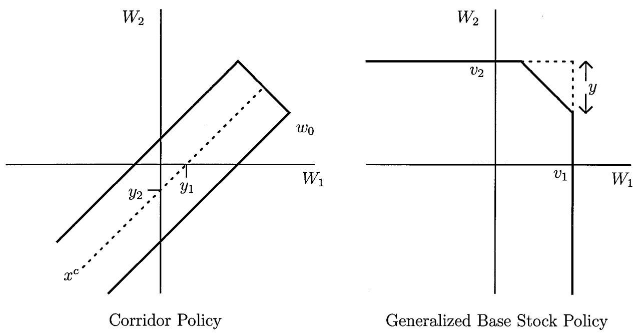
In this section we evaluate the effectiveness of our proposed policies by conducting a series of two-product and five-product experiments for both the setup cost and setup time problems. For the two-product cases, we compare the performance of our proposed policy and two alternative policies against a numerically derived optimal policy. A dynamic programming value iteration algorithm is used to find the optimal policy and evaluate the performance of all four policies. From these data we compute the *suboptimality* for the proposed and two alternative policies by

$$\text{policy's suboptimality} = \frac{\text{policy's cost} - \text{optimal cost}}{\text{optimal cost}} \times 100\%.$$

(See Markowitz 1996 for a detailed specification of the value iteration algorithm and MRW 1997, §3 for details on the implementation of the algorithm.) Because of the large number of inventory states, a dynamic programming algorithm is not feasible for the five-product cases, and thus no optimal policy is derived. Instead, discrete event simulation is used to evaluate the proposed policy and the two alternative policies. Details on the design of the simulation runs can be found in MRW (1997, §3). For all scenarios in this section, we assume that the demand interarrival times, service times, and setup times are exponentially distributed; service is preemptive in the two-product cases and nonpreemptive in the five-product cases.

For systems with two products, we consider 20 setup cost cases and 14 setup time cases; all but two cases for each type of problem assume that the products have identical parameters. Although nearly all of our cases are symmetric, the numerical results in Reiman and Wein (1998) suggest that the heavy traffic analysis is equally accurate for symmetric and asymmetric problems. For systems with five products, we consider six setup cost cases and four setup time cases. We focus on the two-product setting for several reasons. The optimal solution can be numerically computed in this setting, which allows us to assess the suboptimality of our proposed policies; because the optimal policy is a dynamic cyclic policy in the two-product case (i.e., the optimal policy chooses one of the three scheduling options that we allow at each point in time), we conjecture that our proposed policies are optimal in the heavy traffic limit. Also, the

Figure 2. The two alternative policies.



graphical depictions of the various policies in two dimensions (see Figures 2 through 6) help us to understand the subtleties of the behavior of this system. The two alternative policies are described in §3.1, and the numerical results for the setup cost and setup time problems are given in §§3.2 and 3.3, respectively. Our key observations are summarized in §3.4.

3.1. Alternative Policies

To help assess the effectiveness of the proposed policy, we consider two simpler classes of cyclic policies and use heavy traffic analysis to optimize within these classes. One is a variant of a base stock policy, and the other has constant expected lot sizes. Neither alternative policy employs the $s/\sqrt{n}$ refinement that was mentioned in §2.7; we discuss this issue in §3.4.

Generalized Base Stock Policy. The generalized base stock policy has two parameters per product, $\tilde{v}_i$ and $\tilde{y}_i$. If the server is set up for product i, then serve this product if $\tilde{W}_i(t) < \tilde{v}_i$. If $\tilde{W}_i(t) \geqslant \tilde{v}_i$, then idle if product j, the next product to be produced in the cycle, has a workload level $\tilde{W}_j(t) \geqslant \tilde{v}_j - \tilde{y}_j$; otherwise, switch to product j at this point. Hofri and Ross (1987) prove that the make-to-order version of this policy is optimal in a two-product symmetric polling system. The generalized base stock policy can be thought of as a refined version of the cyclic base stock policy considered by Federgruen and Katalan (1996), in the sense that their policy can insert idleness only in a state-independent manner. Although the generalized base stock policy contains $2N$ parameters, the heavy traffic behavior of this policy (see Reiman and Wein 1995 for details) depends on the $\tilde{y}_i$s only via $\max_{1 \leqslant i \leqslant N} \tilde{y}_i$; let us denote this quantity by $\tilde{y}$. Hence, we set each $\tilde{y}_i$ equal to $\tilde{y}$, and optimize over the $N+1$ normalized parameters $(v_1, \ldots, v_N, y)$, where $y = \tilde{y}/\sqrt{n}$ and $v_i = \tilde{v}_i/\sqrt{n}$. If we define $v = \sum_{i=1}^{N} v_i$, then in heavy traffic this policy is equivalent to one that completes production of product i when its inventory level reaches v_i, and employs the workload idling threshold $v - y$; see Figure 2.

To calculate the cost associated with this policy, we move from these natural parameters to those used in §§2 and 3. Under the HTAP assumption, for a given total workload w a generalized base stock policy has cycle center $x_i^c(w) = v_i - \hat{\rho}_i(v - w)/\hat{\rho}$ and cycle length $\tau(w) = 2(v - w)/\hat{\rho}$. Product i's average inventory cost is obtained by substituting these parameters into Equation (5); i.e., $c_i(2(v-w)/\hat{\rho}, v_i - \hat{\rho}_i(v-w)/\hat{\rho}, w)$. Under the HTAP assumption, we can derive the total average cost for the generalized base stock policy for both the setup cost and time problems, and then use a steepest descent algorithm to find the optimal parameter values; see MRW (1997) for details.

Table 1. Test cases for the symmetric two-product problems

	Backorder cost b	Setup cost K	Setup time s	Traffic intensity ρ
Low	5	20	2	0.5
Medium		100		0.7
High	10	200	20	0.9

There are 18 cases for the setup cost problem ($K > 0$, $S = 0$) and 12 cases for the setup time problem ($K = 0$, $s > 0$).

The Corridor Policy. This policy can be stated in terms of switching hyperplanes in the product workload space. The hyperplanes are created to form a fixed width corridor with its long axis orthogonal to the constant workload plane (see Figure 2). This policy is very similar to the "prism corridor" policy depicted in Sharifnia et al. (1991, Figure 6). The corridor policy represents a natural embodiment of the "constant lot size" philosophy within a dynamic stochastic framework, and is defined by $N+2$ parameters: the cycle length $\tilde{\tau}$ (or corridor width), the idling threshold $\tilde{w}_0$, and the parameters $(\tilde{y}_1, \ldots, \tilde{y}_N)$, which determine the intercept of the corridor's axis. We can use these variables and the notation of the previous two sections to formulate the average inventory cost of the policy in heavy traffic. For a given workload w, the cycle center x_i^c is equal to $w/N + y_i$ and the cycle length is τ. Product i's average inventory cost for workload w is then $c_i(\tau, w/N + y_i, w)$. Steepest descent is used to derive the optimal parameter values; see MRW (1997) for details.

3.2. The Setup Cost Problem

Two-Product Cases. To standardize the two-product scenarios, we set the service rates $\mu_1 = \mu_2 = 1$ and control the utilization rates ρ_i by varying the demand rates λ_i. We also set $h_2 = 1$ and — by modifying h_1, b_1, and b_2 — select product 2 as the least cost product. Inventory costs and

Table 2. Average suboptimality of the proposed policy: setup cost problem.

	Backorder cost b	Setup cost K	Traffic intensity ρ
Low	5.1%	8.7%	9.3%
Medium		4.3%	5.8%
High	6.9%	5.0%	2.9%

Average suboptimality of 2 asymmetric cases = 3.0%.
Overall average suboptimality of 20 cases = 5.7%.

Table 3. Average suboptimality of the corridor and generalized base stock (GBS) policies: setup cost problem.

	Corridor Policy			Generalized Base Stock Policy		
	Back-order cost b	Setup cost K	Traffic intensity ρ	Back-order cost b	Setup cost K	Traffic intensity ρ
Low	5.6%	5.6%	5.9%	21.5%	22.4%	17.9%
Medium		6.1%	3.3%		23.4%	22.4%
High	6.6%	6.6%	9.1%	25.7%	25.1%	30.5%

Average suboptimality of 2 asymmetric cases = 14.0% (corrider) = 40.8% (GBS).
Overall average suboptimality of 20 cases = 6.9% (corrider) = 25.3% (GBS).

arrival rates are identical across products in the 18 *symmetric* cases, and each case is characterized by three parameters: backorder cost, traffic intensity, and setup cost per cycle. We examine all permutations of values shown in Table 1; notice that some of these scenarios grossly violate the heavy traffic conditions. The parameters for the first asymmetric case are $\lambda_1 = 0.6$, $\lambda_2 = 0.3$, $h_1 = 2, b_1 = 10$, $b_2 = 5$ and $K = 200$. The second asymmetric case is the same as the first, except that the backorder cost is doubled to $b_1 = 20$ and $b_2 = 10$.

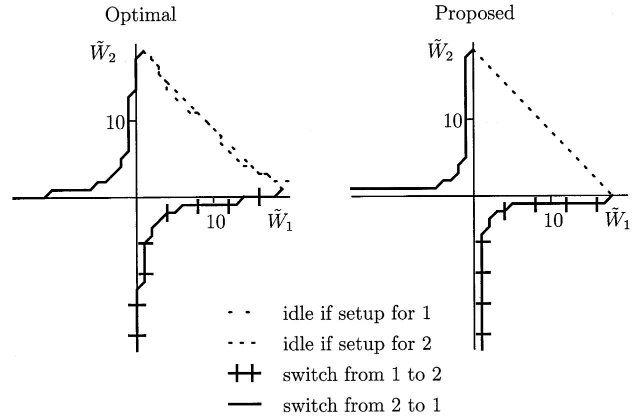
MRW (1997, Table 2) displays the optimal cost and the costs for the three policies for the 20 two-product cases. These results are summarized in Tables 2 and 3, which show the average suboptimalities over individual parameters for each policy (the main body of these tables refers to the 18 symmetric cases). The switching curves for the optimal and proposed policies for the ($b = 5$, $K = 200$, $\rho = 0.9$) two-product symmetric case are depicted in Figure 3, and corresponding curves for the $b_1 = 10$ asymmetric case are displayed in Figure 4.

Five-Product Cases. We set $\lambda_i = 0.18$ and $\mu_i = 1$ for $i = 1, \ldots, 5$ for each of the six cases, resulting in a traffic intensity of 0.9. We also set $b_i = 5h_i$ for $i = 1, \ldots, 5$ for half the cases and $b_i = 10h_i$ for the other half. Each case is characterized by h_i, b_i, and the setup cost. Four of the six cases are symmetric ($h_i = 1$ for $i = 1, \ldots, 5$), and two of the six cases are asymmetric ($h_i = i$ for $i = 1, \ldots, 5$). The average cost for each policy (along with 95% confidence intervals) is displayed in the first six rows of Table 4.

3.3. The Setup Time Problem

As in the two-product setup cost test cases, we assume that $\mu_1 = \mu_2 = 1$ and $h_2 = 1$. In the 12 symmetric scenarios, each product's inventory costs and service utilizations are identical and we vary only the backorder cost, the traffic intensity and the average setup time per cycle. Table 1 reports all the permutations of values analyzed. The first asymmetric scenario is defined by $\lambda_1 = 0.6$, $\lambda_2 = 0.3$, $\mu_1 = \mu_2 = 1$, $h_1 = 2$, $h_2 = 1$, $b_1 = 10$, $b_2 = 5$ and $s = 20$. The second asymmetric scenario is identical except that the backorder costs are $b_1 = 20$ and $b_2 = 10$.

Figure 3. Switching curves for a symmetric setup cost case.



The individual results for the 14 runs are displayed in MRW (1997, Table 8), and policy summaries for these runs are given in Tables 5 and 6. In addition, Figures 5 and 6 provide a graphical depiction of the proposed and optimal policies for a symmetric case ($b = 5$, $s = 2$, $\rho = 0.9$) and the $b_1 = 10$ asymmetric case, respectively.

Results for two five-product scenarios can be found in Table 4; they are identical to the setup cost scenarios described in §3.2 except that setup times (with $s = 50$) are incurred rather than setup costs.

3.4. Discussion

Our observations from the numerical results are summarized in this subsection. The five-product cases are discussed after two-product cases.

Performance of the Proposed Policy. In the setup cost cases, the proposed policy's average suboptimality is 6.0% over the 18 symmetric scenarios. The policy performs very well when the heavy traffic conditions are satisfied; for example, the suboptimality is 0.7% when $b_1 = 5$, $K = 200$,

Figure 4. Switching curves for the $b_1 = 10$ asymmetric setup cost case.

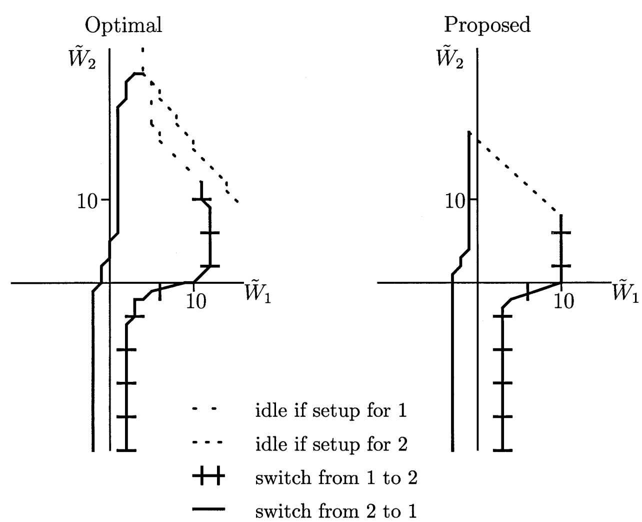


Table 4. Results for the five product cases.

Back-order Cost	Setup Cost or Time	Cost Structure	Cost of Proposed Policy	Cost of Corridor Policy	Cost of Gen. Base Stock Policy
$b=5$	$K=50$	Symmetric	25.32($\pm$0.46)	28.78($\pm$0.63)	33.23($\pm$0.45)
$b=5$	$K=500$	Symmetric	37.23($\pm$0.10)	37.25($\pm$0.32)	44.02($\pm$0.20)
$b=10$	$K=50$	Symmetric	36.79($\pm$0.73)	39.39($\pm$0.91)	40.54($\pm$0.68)
$b=10$	$K=500$	Symmetric	47.08($\pm$0.38)	46.02($\pm$0.27)	54.29($\pm$0.52)
$b=5$	$K=500$	Asymmetric	79.91($\pm$0.41)	86.65($\pm$0.67)	121.11($\pm$1.57)
$b=10$	$K=500$	Asymmetric	98.46($\pm$0.77)	105.98($\pm$1.49)	138.88($\pm$1.14)
$b=5$	$s=50$	Symmetric	215.4 ($\pm$4.9)	228.0 ($\pm$16.1)	214.1 ($\pm$2.6)
$b=10$	$s=50$	Symmetric	264.7 ($\pm$10.4)	532.5 ($\pm$136.8)	260.2 ($\pm$4.7)
$b=5$	$s=50$	Asymmetric	610.8 ($\pm$8.9)	683.9 ($\pm$35.2)	661.0 ($\pm$9.1)
$b=10$	$s=50$	Asymmetric	737.4 ($\pm$18.7)	827.9 ($\pm$66.5)	791.7 ($\pm$16.1)

and $\rho=0.9$. Considering that the proposed policy was constructed via a heavy traffic approximation, it operates reasonably well over a wide range of system parameters, including a low utilization rate of 0.5. Not surprisingly, the policy performs worst when the traffic intensity is low, the setup costs are small, and the backorder costs are high. The policy also performs well (2.6% and 3.4% suboptimalities) in the asymmetric cases.

In the setup time cases, the average suboptimality over the 12 symmetric cases is 7.2%. The policy performs very well (1.8% average suboptimality) when the traffic intensity is high, but degrades somewhat in the lighter traffic cases. It also performs well in the asymmetric cases (1.5% and 3.3% suboptimalities).

Switching Curves. The switching curves of the proposed and optimal policies are remarkably similar in Figures 3 to 6 and are unlike either the corridor or generalized base stock policies. In the two symmetric problems (Figures 3 and 5), these curves have the same general shape as predicted by our heavy traffic analysis: a distinctive constant-workload idling threshold, a wide cycle length for large positive and negative inventories, and a narrow cycle length about the zero total workload level. In the asymmetric setup cost problem in Figure 4, the three-region categorization predicted by the heavy traffic theory is easily recognizable in the optimal policy. Figure 6 confirms that lot sizes shrink as the idling threshold is approached. Finally, as the total workload $\tilde{w}$ tends to minus infinity, lot sizes appear to be growing roughly with $-\tilde{w}$ in Figure 5 and with $\sqrt{-\tilde{w}}$ in Figure 6.

Two key differences between the proposed and optimal policies emerge from studying Figures 3 to 6; numerical results (not reported here) verify that both discrepancies dissipate as the traffic intensity approaches unity, and then get more severe in the lower utilization cases. First, in all four figures, the proposed heavy traffic policies have a tendency to backorder more than the optimal policy; this observation is most obvious in the upper right portion of Figure 5. Because the HTAP does not hold precisely for the original stochastic system, the optimal policy hedges against backorders slightly more than the proposed heavy traffic policy, which assumes that the inventory levels respond in a deter-

Table 5. Average suboptimality of the proposed policy: setup time problem.

	Backorder Cost b	Setup Time s	Traffic Intensity ρ
Low	5.9%	6.2%	11.8%
Medium			8.1%
High	8.5%	8.2%	1.8%

Average suboptimality of 2 asymmetric cases = 2.4%.
Overall average suboptimality of 14 cases = 6.5%.

Figure 5. Switching curves for a symmetric setup time case.

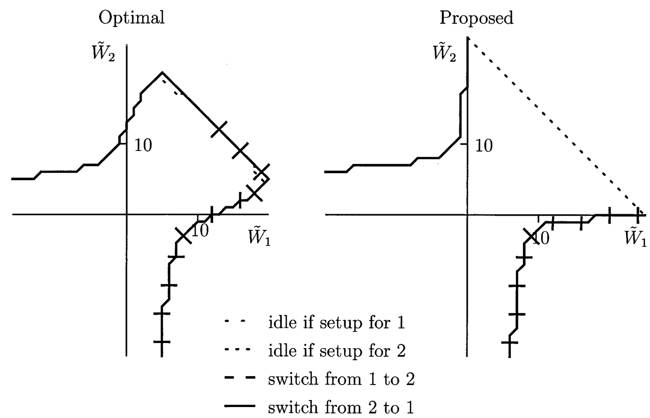


Figure 6. Switching curves for the $b_1=10$ asymmetric setup time case.

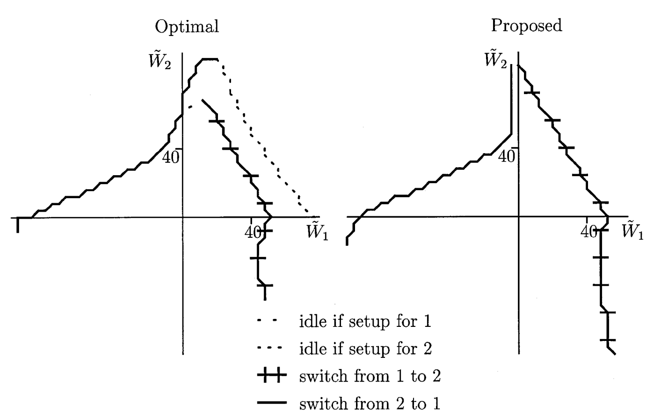


Table 6. Average suboptimality of the corridor and generalized base stock (GBS) policies: setup time problem.

	Corridor Policy			Generalized Base Stock Policy		
	Backorder Cost b	Setup Time s	Traffic Intensity ρ	Backorder Cost b	Setup Time s	Traffic Intensity ρ
Low	12.8%	8.4%	18.7%	9.5%	9.8%	14.0%
Medium			12.1%			9.8%
High	12.1%	16.5%	6.7%	11.8%	11.6%	8.2%

Average suboptimality of 2 asymmetric cases = 45.8% (corridor) = 13.7% (GBS).
Overall average suboptimality of 14 cases = 17.3% (corridor) = 11.1% (GBS).

ministic fashion in the fluid limit. In terms of these figures, the cycle lengths (i.e., the distance along the total workload line between the solid and dashed curves) tend to be slightly smaller in the optimal policy; consequently, the workload process spends less time in the backorder region and some of our remarks in §2.5 regarding the inventory levels at switching epochs hold only in very heavy traffic. This limitation of the heavy traffic theory was also noted in Wein (1992).

The other main discrepancy occurs near the idling threshold in the asymmetric cases: In Figure 4, the optimal lot sizes for product 1 decrease, rather than staying constant, as the workload idling threshold is approached; and in Figures 4 and 6 there is a different idling threshold for each product. This discrepancy can be explained as follows. When the total workload is positive, the switchover cost (in Figure 4) or time (in Figure 6) makes it beneficial to be set up for product 1, so as to efficiently protect against costly product 1 backorders. If ρ is not close to one, then it is likely that the total inventory workload will increase while producing product 2; that is, the increase in product 2's inventory workload will exceed the reduction in product 1's inventory workload. The optimal policy takes advantage of this imbalance by allowing product 2's inventory to grow beyond the product 1 idling threshold; this extra product 2 inventory allows the server to idle while setup for product 1.

Probability of Stockout. Gallego (1990) uses stochastic optimization and Anupindi and Tayur (1998) use perturbation analysis to derive base stock policies for which the time average probability of being out of stock of product i is $h_i/(b_i+h_i)$. Although product i's stockout probability for our dynamic cyclic policy is not $h_i/(b_i+h_i)$ for a fixed total workload level w, computational experiments show that the time average stockout probability (derived by integrating over the stationary distribution of the total workload w) for product i is indeed very close to $h_i/(h_i+b_i)$ when $\rho = 0.9$.

Performance of the Corridor Policy. The corridor policy exhibits erratic behavior. The policy performs very well in the symmetric setup cost cases (it outperforms the proposed policy in six of the 18 scenarios, all of which have low or medium utilizations), but degrades slightly at high utilization. A comparison of Figures 2 and 3 leads us to believe that the parameters of the policy are being set correctly at high utilizations, and the performance degradation is a result of the corridor policy's inability to employ mean lot sizes that are state-dependent.

The corridor policy does not perform as well in the symmetric setup time cases; it is not able to increase the cycle length τ for small total workloads and so has difficulty recovering from this high backorder region. In contrast to the symmetric setup cost cases, the corridor policy's performance diminishes in light traffic; we have not determined how much of this degradation is because of the inaccuracy of the heavy traffic approximation at low utilizations, and how much is intrinsic to the policy.

The corridor policy performs much worse when asymmetry is present: Its suboptimality is 13.1% and 14.9% in the two setup cost cases and increases to 30.6% and 60.9% in the two setup time cases. Comparising Figures 2, 4, and 6, it would appear that the corridor policy would never be very close to optimal for an asymmetric problem. In fact, Figure 6 suggests that a *hyperplane corridor* policy (see Sharifnia et al. 1991, Figure 7) might perform reasonably well in the asymmetric setup time problem; in the two-product case, the two lines forming the corridor in Figure 2 would not be parallel in the hyperplane corridor policy; but would intersect at an idling point in the upper right portion of the graph and generate a cone-shaped corridor emanating out in the southwesterly direction.

Performance of the Generalized Base Stock Policy. The generalized base stock policy performs better in the setup time cases than in the setup cost cases: Its average suboptimality is 23.6% for the 18 symmetric setup cost scenarios and 10.7% for the 12 symmetric setup time cases. In contrast to the corridor policy, the generalized base stock policy's use of large lot sizes when the total workload is negative is a key reason for its ability to avoid poor performance in the symmetric setup time cases; however, these large lot sizes lead to considerable backordering in the setup cost scenarios. Like the corridor policy, the generalized base stock policy's performance deteriorates at high utilizations in the setup cost cases and at low utilizations in the setup time cases.

The generalized base stock policy's suboptimality is 35.8% and 45.8% in the asymmetric setup cost cases and 11.5% and 15.8% in the asymmetric setup time case. It is interesting to note that the generalized base stock policy handles expensive inventory in a manner opposite to that of the proposed policy. The order-up-to level of the most costly good is set larger than those of less expensive products to reduce the risk of expensive backordering; in contrast, the proposed policy minimizes the excess or deficit amounts of expensive inventory. It is clear that the generalized base stock policy is incapable of closely approximating the optimal setup cost solution in Figure 4.

Five-Product Examples. In the six setup cost cases in Table 4, the proposed and corridor policies are roughly comparable in the two $K = 500$ symmetric cases, and the corridor policy is about 10% more costly in the two $K = 50$ symmetric cases and about 8% more expensive in the two asymmetric cases. The generalized base stock policy does not fare as well in the four symmetric setup cost cases, incurring an 18.7% cost increase relative to the proposed policy, on average. Once again, the generalized base stock policy performs very poorly in the asymmetric setup cost cases.

In contrast, the generalized base stock policy performs slightly better than the proposed policy in the two symmetric setup time cases in Table 4 and is about 8% more costly than the proposed policy in the two asymmetric cases. Both of these policies outperform the corridor policy in the four setup time cases. The corridor policy is about 12% more costly than the proposed policy in the two asymmetric cases, but performs extremely poorly in one of the two symmetric cases.

To compare the relative cost differences in the two-product cases and the five-product cases, we can identify the six symmetric cases in Table 4 with their two-product counterparts in MRW (1997, Tables 2 and 8); for example, the first scenario in Table 4 corresponds to the $b = 5,\ K = 20,\ \rho = 0.9$ case in MRW (1997, Table 2). For the four setup cost cases, the cost increases of the alternative policies relative to the proposed policy are somewhat larger for the two-product cases: The generalized base stock policy's average cost increase is 5.8% for the two-product cases versus 4.7% for the five-product cases, and the corresponding quantities for the generalized base stock policy are 26.3% and 18.7%, respectively. For the two setup time cases, the average cost increase of the generalized base stock policy is 2.7% for the two-product scenarios and -1.1% for the five-product cases. Disregarding the poor performance of the corridor policy in one of the five-product symmetric setup cost scenarios, it appears that the relative cost advantage of the proposed policy degrades slightly when the number of products increases from two to five; however, further experiments are required to fully investigate this issue. This degradation may occur because the time scale decomposition underlying the HTAP is a less accurate (for a given traffic intensity) approximation when the number of products is increased.

Lack of Robustness. Simulation results not reported here show that the performance of the three policies are rather sensitive to the policy parameters, particularly in the setup time problem; this is somewhat surprising, given the robustness of some simpler models (e.g., the EOQ model) that capture the tradeoff between inventory costs and setups. Because it is unable to increase its lot sizes as the total inventory decreases, the corridor policy is clearly the least robust of the three policies: If the corridor width is set too narrow (as apparently happened in the eighth row of Table 4), then stability problems can set in (notice the confidence intervals for this case).

The $s/\sqrt{n}$ Refinement. Recall that the $s/\sqrt{n}$ refinement described in §2.7 is incorporated into the proposed policy, but not the two alternative policies. We tested all three policies with and without the refinement, and summarize our findings here. The refinement had a minor effect on the performance of the proposed policy in the $\rho = 0.9$ cases; however, by decreasing the cycle length, it significantly improved performance in the lower utilization cases. The refinement had a mixed influence on the generalized base stock policy, sometimes improving and sometimes degrading performance. Overall, it slightly impaired performance. The refinement had a negative effect on the corridor policy and led to a severe stability problem in the eighth row of Table 4.

Summary. Although additional asymmetric cases need to be investigated before drawing definitive conclusions for the two-product problems, our observations can be summarized as follows.

The proposed policy performs very well in the 34 two-product cases: Figures 3 to 6 confirm that it captures nearly all of the complexities of the optimal policy, its suboptimality is 6.0% over the 34 cases (and 2.1% over the 12 cases that do not obviously violate the heavy traffic conditions), and it is quite robust with respect to the heavy traffic conditions. However, the relative superiority of the proposed policy appears to degrade slightly as the number of products increases, and this issue requires further investigation.

The two alternative policies are not flexible enough to consistently capture the subtleties of the optimal policy. The corridor policy outperforms the generalized base stock policy in 24 of the 34 two-product examples, and its average suboptimality is 11.2% as compared to 19.5% for the generalized base stock policy. Nonetheless, in the setup time cases the corridor policy fails to use large lot sizes when the total workload is negative and can perform erratically (see Table 6). The generalized base stock policy does not perform well with setup costs or with asymmetric costs in the setup time problem. It does, however, perform well in the symmetric cost, setup time problem.

4. CONCLUDING REMARKS

The stochastic economic lot scheduling problem is a long standing problem in operations management. Unfortunately, there has been no success in obtaining an optimal solution to this problem using standard techniques (e.g., semi-Markov decision process theory). In this paper we restrict ourselves

Table 7. Summary of cycle length and inventory behavior by region.

	Cycle Length in Setup Time Problem	Cycle Length in Setup Cost Problem	Cheapest Product	Other Products
Region I	Set small to avoid inventory costs, time is cheap	Set large to avoid setups at expense of holding costs	Excess buffer is created	
Region II	Dynamic trade-off between wasted time and inventory costs, time is important	Dynamic trade-off between setup costs and inventory costs	Balanced between excess and backorder	Balanced between excess and backorder
Region III	Set large to reduce backorders, time is critical	Set small to minimize backordering of all products	Severe backordering is allowed	

to the class of dynamic cyclic policies, where the server has three options at each point in time: idle, produce the product that is currently set up, or switch over to the next product in the fixed sequence. By assuming (and conjecturing) that the heavy traffic averaging principles derived by Coffman et al. (1998) hold in our more general setting, we make considerable progress on this problem: For the setup cost problem, the optimal heavy traffic lot-sizing policy is derived in closed form, and the idleness policy is reduced to the numerical calculation of a single threshold value. For the setup time problem, some key qualitative characteristics of the optimal heavy traffic policy are derived; moreover, regardless of the number of products, we reduce the problem to a one-dimensional diffusion control problem that is solved numerically.

The explicitness of our results, coupled with the surprisingly intricate behavior of the optimal policy, leads to some new insights into the optimal solution to the SELSP. These insights are summarized in §§1.8 and 2.5 and describe how the dynamic lot-sizing policy depends upon whether (i) setup costs or setup times are incurred; (ii) the cost structure is symmetric or asymmetric across products; and (iii) the total workload embodied in the current finished goods inventory is much less than zero (region III), in a neighborhood containing zero (region II), or larger than zero and near the optimal idling threshold (region I). We summarize our results for the three regions in Table 7.

We also perform a heavy traffic analysis of two classes of policies that are closely related to ones analyzed by Federgruen and Katalan (1996) and Anupindi and Tayur (1998) and by Sharifnia et al. (1991), respectively; the unified treatment of the optimal policy and the two alternative policies (see, in particular, Figures 2 through 6) makes transparent the relative strengths and weaknesses of the alternative policies. A computational study is undertaken to compare the proposed policy and these two alternative policies to the numerically computed optimal policy in 34 two-product examples. The computational study (see §3.4 for a description of the key observations) confirms that the insights summarized in §§1.8 and 2.5 do indeed occur in the optimal policy. Moreover, numerical results for the two-product examples show that the proposed policy is reasonably close to optimal, is robust with respect to the heavy traffic conditions, and outperforms the two alternative policies; in contrast, the two alternative policies lack the sophistication required to imitate the subtleties of the optimal policy, and their behavior is somewhat erratic.

The computational study also includes 10 five-product examples, and the results suggest that the relative superiority of the proposed policy degrades slightly as the number of products increases. Although we have investigated one heuristic heavy traffic refinement (the $s/\sqrt{n}$ refinement in §2.7), we believe that other refinements to the proposed policy (including nonheavy traffic refinements, such as Gallego's 1990 recovery from a disruption to a fluid cycle) may lead to improved robustness and should be investigated; such refinements may be necessary in order to develop effective policies for problems with many products and moderate traffic intensities.

This study has only considered dynamic cyclic policies, where each class is served once per cycle in a fixed sequence. More generally, the cost of dynamic *periodic* policies (or *polling tables*), where each product can be produced more than once in a cycle, can be evaluated using the methods developed here. However, we have not yet found an efficient method for optimizing within this class of policies, and a thorough investigation of dynamic periodic and nonperiodic policies is left for future research.

APPENDIX

In this Appendix we derive the formulas for the optimal cycle center Equation (11), the optimal cycle length Equation (12), and the workload level region separators in Equation (10). We do this in two stages: (i) find the optimal cycle center x^c in terms of arbitrary w and τ, and (ii) optimize over the cycle length τ in terms of an arbitrary w.

The Optimal Cycle Center

We begin by showing the existence of a cost-minimizing cycle center x^c for a given total workload w and cycle length τ. Note that the cost function $c(\tau, x^c, w)$ is differentiable with respect to x^c and its derivative is continuous. If one ignores the constant workload constraint $\sum_{i=1}^{N} x_i^c = w$, for fixed τ the cost function in terms of x^c is piecewise-quadratic with linear edges; its second derivative is a nonnegative step function. Thus $c(\tau, x^c, w)$ is convex and the restriction of the cost function of the constant workload hyperplane determined by w is also convex. This fact implies the existence of a solution to the constrained minimization problem: Choose x^c to minimize $c(\tau, x^c, w)$ subject to $\sum_{i=1}^{N} x_i^c = w$.

Now we use the cost function in (5) to find the optimal cycle center. The total workload constraint can be used to eliminate one variable and express the cost function as a piecewise polynomial function of $N-1$ variables. Any $N-1$ components of x^c can be used. Over the constant workload hyperplane, the polynomial order of the $N-1$ variables fluctuates between one and two, depending on whether $|x_i^c| > \tau\hat{\rho}_i/2$ or $|x_i^c| \leqslant \tau\hat{\rho}_i/2$, respectively. For the gradient to be equal to zero, each of the $N-1$ variables must be of second order. Consequently, at the optimal x^c, at least $N-1$ of the $c_i(\tau, x_i^c, w)$s are of order two, with the remaining component possibly being linear. To see this, suppose that some of the $c_i(\tau, x_i^c, w)$s are not of order two, and let j denote the index of such a term. If we eliminate x_j, the gradient equation can then be written as

$$\nabla_{x^c}\left[\sum_{\substack{i=1\\ i\neq j}}^{N} c_i(\tau, x_i^c, w) + c_j\left(\tau, w - \sum_{\substack{i=1\\ i\neq j}}^{N} x_i^c, w\right)\right] = 0.$$

This equation will have a solution only if the remaining $N-1$ $c_i(\tau, x_i^c, w)$s are quadratic.

The following proposition, which is proved in MRW (1997, Appendix), greatly simplifies our analysis.

PROPOSITION 1. *If there are only $N-1$ quadratic $c_i(\tau, x_i^c, w)$ terms in the total cost function at the optimal x^c, then the linear term must be $c_N(\tau, x_N^c, w)$.*

As a consequence, the optimal cycle center, or average amount of inventory per cycle, for product $i < N$ is restricted to the region $[-\tau\hat{\rho}_i/2, \tau\hat{\rho}_i/2]$, whereas product N's cycle center can be arbitrarily far from zero. Intuitively, this fact suggests that product N, which is the least cost product by our indexing convention, is the product that will hold the excess or deficit amounts of work when the total workload w fluctuates far from zero.

We now use Proposition 1 to find the optimal cycle center x^{c*}. Without loss of generality, the workload constraint is used to eliminate x_N^c from the cost function, so that $x_N^c = w - \sum_{i=1}^{N-1} x_i^c$. To find the optimal center, we take the gradient of (6) and set it equal to zero:

$$\nabla_{x^c}\left[\sum_{i=1}^{N-1} c_i(\tau, x_i^c, w) + c_N\left(\tau, w - \sum_{i=1}^{N-1} x_i^c, w\right) + \frac{k}{\tau}\right] = 0. \tag{31}$$

At this point, we do not know whether $c_N(\tau, x_N^c, w)$ is linear or quadratic. Let $\bar{x}^c$ be the $(N-1)$-dimensional vector that solves (31) under the assumption that all N the $c_i(\tau, x^c, w)$s are quadratic in x_i^c. Taking the $(N-1)$-dimensional gradient, we find that $\bar{x}^c$ satisfies (w and τ multiply their vectors component-wise in the analysis below)

$$\frac{1}{\tau}A\bar{x}^c - \gamma_1 - \frac{w}{\tau}\gamma_2 = 0, \tag{32}$$

where

$$A = \begin{bmatrix} \ddots & & 0 \\ & r_i^{-1} & \\ 0 & & \ddots \end{bmatrix} + r_N^{-1}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{bmatrix},$$
$$\gamma_1 = \begin{bmatrix} \vdots \\ \frac{b_i - h_i}{2} - \frac{b_N - h_N}{2} \\ \vdots \end{bmatrix}, \quad \gamma_2 = \begin{bmatrix} \vdots \\ r_N^{-1} \\ \vdots \end{bmatrix}. \tag{33}$$

Thus,

$$\bar{x}^c = \tau A^{-1}\gamma_1 + wA^{-1}\gamma_2, \tag{34}$$

where the matrix elements of A^{-1} are

$$\alpha_{ij} = -\frac{r_i r_j}{\sum_{l=1}^{N} r_l} \quad \text{for } i \neq j,$$

and

$$\alpha_{ii} = r_i \frac{\sum_{l=1}^{N} r_l - r_i}{\sum_{l=1}^{N} r_l}. \tag{35}$$

If $|w - \sum_{i=1}^{N-1} \bar{x}_i^c| \leqslant \tau\hat{\rho}_N/2$, then $c_N(\tau, x_N^c, w)$ is indeed quadratic and $\bar{x}_i^c$ determines the optimal center: $x_i^{c*} = \bar{x}_i^c$ for $i < N$ and $x_N^{c*} = w - \sum_{i=1}^{N-1} \bar{x}_i^{c*}$. If $|w - \sum_{i=1}^{N-1} \bar{x}_i^c| > \tau\hat{\rho}_N/2$, then we must solve the multivariate gradient equation with the linear form of $c_N(\tau, x_N^c, w)$. With this substitution, Equation (31) decomposes into univariate expressions of the form

$$\frac{h_i - b_i}{2} + \frac{x_i^c}{\tau r_i} - h_N = 0 \quad \text{if } w - \sum_{i=1}^{N-1} \bar{x}_i^c > \tau\hat{\rho}_N/2, \tag{36}$$

$$\frac{h_i - b_i}{2} + \frac{x_i^c}{\tau r_i} + b_N = 0 \quad \text{if } w - \sum_{i=1}^{N-1} \bar{x}_i^c < -\tau\hat{\rho}_N/2. \tag{37}$$

Putting the results from (34) and (36)–(37) together, we obtain a complete expression for the optimal cycle center, which is given in (11). See MRW (1997, Appendix) for more details on the region II calculation.

The Optimal Cycle Length

The optimal cycle length τ^* can be derived by simple calculus. Substituting the optimal cycle center x_i^{c*} into (5) yields the average inventory cost for each product as a function of τ and w, $c_i(\tau, w)$. The optimal value of τ is determined by equating the derivative of the total cost function

$\sum_{i=1}^{N} c_i(\tau, w) + k/\tau$ to zero, and is given by (12); see MRW (1997, Appendix) for details.

The Region Boundaries

With an expression for τ^* in hand, we can find the region boundaries w_1 and w_2. Using (12) to equate the cycle lengths at the borders of region II and its two adjacent regions yields the boundaries in (10).

ACKNOWLEDGMENTS

This research was supported by a grant from the Leaders for Manufacturing Program at MIT, NSF grant DDM-9057297, and EPSRC grant GR/J71786, The last author thanks the Statistical Laboratory at the University of Cambridge for its hospitality while some of this work was carried out. The authors also thank the referees for comments that improved the paper's exposition.

REFERENCES

Anupindi, R., S. Tayur. 1998. Managing stochastic multi-product systems: model, measures, and analysis. *Oper. Res.* **46** S98–S111.

Beneš, V. E., L. A. Shepp, H. S. Witsenhausen. 1980. Some solvable stochastic control problems. *Stochastics* **4** 39–83.

Bourland, K. E., C. A. Yano. 1995. The strategic use of capacity slack in the economic lot scheduling problem with random demand. *Management Sci.* **40** 1690–1704.

Coffman, E. G., Jr., A. A. Puhalskii, M. I. Reiman. 1995. Polling systems with zero switchover times: a heavy-traffic averaging principle. *Ann. Appl. Probab.* **5** 681–719.

——,——,——. 1998. Polling systems in heavy-traffic: a Bessel process limit. *Math. Oper. Res.* **23** 257–304.

Cox, D. R., W. L. Smith. 1961. *Queues.* Methunen, London, UK.

Dobson, G. 1987. The economic lot scheduling problem — achieving feasibility using time-varying lost sizes. *Oper. Res.* **35** 764–771.

Elmaghraby, S. E. 1978. The economic lot scheduling problem (ELSP): review and extensions. *Management Sci.* **24** 587–598.

Federgruen, A., Z. Katalan. 1995. Determining production schedules under base-stock policies in single facility multi-item production systems. To appear in *Oper. Res.*

——,——. 1996. The stochastic economic lot scheduling problem: cyclical base-stock policies with idle times. *Management Sci.* **42** 783–796.

Gallego, G. 1990. Scheduling the production of several items with random demands in a single facility. *Management Sci.* **36** 1579–1592.

——, D. Joneja. 1994. Economic lot scheduling problem with raw material consideration. *Oper. Res.* **42** 92–101.

Graves, S. C. 1980. The multi-product production cycling problem. *AIIE Trans.* **12** 233–240.

Harrison, J. M. 1985. *Brownian Motion and Stochastic Flow Systems.* John Wiley, New York.

Hofri, M., K. W. Ross. 1987. On the optimal control of two queues with server set-up times and its analysis. *SIAM J. Comput.* **16** 399–420.

Hsu, W. 1985. On the general feasibility test of scheduling lot sizes for several products on one machine. *Management Sci.* **29** 93–105.

Iglehart, D. L., W. Whitt. 1970. Multiple channel queues in heavy traffic, I and II. *Adv. in Appl. Probab.* **2** 150–177, 355–364.

Kushner, H. J., P. G. Dupuis. 1992. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, New York.

Leachman, R. C., A. Gascon. 1988. A heuristic scheduling policy for multi-item, single machine production systems with time-varying stochastic demands. *Management Sci.* **34** 377–390.

Mandl, P. 1968. *Analytical Treatment of One-Dimensional Markov Processes.* Springer-Verlag, New York.

Markowitz, D. M. 1996. A unified approach to single machine scheduling: heavy traffic analysis of dynamic cyclic policies. Ph.D. Thesis. Operations Research Center, MIT, Cambridge, MA.

——, M. I. Reiman, L. M. Wein. 1997. The stochastic economic lot scheduling problem: heavy traffic analysis of dynamic cyclic policies. Sloan School of Management, MIT, Cambridge, MA.

——, L. M. Wein. 1996. Heavy traffic analysis of dynamic cyclic policies: a unified approach to single machine scheduling. To appear in *Oper. Res.*

Morse, P. 1958. *Queues, Inventories and Maintenance.* John Wiley, New York.

Qiu, J., R. Loulou. 1995. Multiproduct production/inventory control under random demands. *IEEE Trans. Automatic Control* **40** 350–356.

Reiman, M. I. 1984. Open queueing networks in heavy traffic. *Math. Oper. Res.* **9** 441–458.

——. 1988. A multiclass feedback queue in heavy traffic. *Adv. Appl. Probab.* **20** 179–207.

——, L. M. Wein. 1998. Dynamic scheduling of a two-class queue with setups. *Oper. Res.* **46** 532–547.

Sethi, S. P., Q. Zhang. 1995. Hierarchical production and setup scheduling in stochastic manufacturing systems. *IEEE Trans. Automatic Control* **40** 924–930.

Sharifnia, A., M. Caramanis, S. B. Gershwin. 1991. Dynamic setup scheduling and flow control in manufacturing systems. *Discrete Event Dynamic Systems: Theory and Appl.* **1** 149–175.

Sox, C. R., J. A. Muckstadt. 1997. Optimization-based planning for the stochastic lot scheduling problem. *IIE Trans.* **29** 349–358.

Wein, L. M. 1992. Dynamic scheduling of a multiclass make-to-stock queue. *Oper. Res.* **40** 724–735.

Zipkin, P. 1991. Computing optimal lot sizes in the economic lot scheduling problem. *Oper. Res.* **39** 56–63.