DYNAMIC SCHEDULING OF A PRODUCTION/INVENTORY SYSTEM WITH BY-PRODUCTS AND RANDOM YIELD

Jihong Ou and Lawrence M. Wein

OR 250-91 May 1991

DYNAMIC SCHEDULING OF A PRODUCTION/INVENTORY SYSTEM WITH BY-PRODUCTS AND RANDOM YIELD

Jihong Ou

Operations Research Center, M.I.T.

and

Lawrence M. Wein

Sloan School of Management, M.I.T.

Abstract

Motivated by semiconductor wafer fabrication, we consider a scheduling problem for a single-server multiclass queue. A single workstation fabricates semiconductor wafers according to a variety of different processes, where each process consists of multiple stages of service with a different general service time distribution at each stage. A batch (or lot) of wafers produced according to a particular process randomly yields chips of many different product types, and completed chips of each type enter a finished goods inventory that services exogenous customer demand for that type. The scheduling problem is to dynamically decide whether the server should be idle or working, and in the latter case, to decide which stage of which process type to serve next. The objective is to minimize the long run expected average cost, which includes costs for holding work-in-process inventory (which may differ by process type and service stage) and backordering and holding finished goods inventory (which may differ by product type). We assume the workstation must be busy the great majority of the time in order to satisfy customer demand, and approximate the scheduling problem by a control problem involving Brownian motion. A scheduling policy is derived by interpreting the exact solution to the Brownian control problem in terms of the production/inventory system. The proposed dynamic scheduling policy takes a relatively simple form and appears to be effective in numerical studies.

May 1991

DYNAMIC SCHEDULING OF A PRODUCTION/INVENTORY SYSTEM WITH BY-PRODUCTS AND RANDOM YIELD

Jihong Ou

Operations Research Center, M.I.T.

and

Lawrence M. Wein

Sloan School of Management, M.I.T.

1. Introduction and Summary

Our study is motivated by a scheduling problem faced in semiconductor wafer fabrication, which consists of the production of wafers that typically contain between 10 and several thousand computer chips. Because the production technology is very complex and not well understood, each completed wafer contains a random number of chips of varying grades of quality, including some chips that are deemed useless and consequently scrapped. Chips of different grades of quality are classified into different types of products, where each type has its own market and demand. Since manufacturing cycle times are very long and prompt customer delivery is required to remain competitive (see Harrison et al. 1990), many wafer fabrication facilities are forced to operate (at least partially) in the make-to-stock mode of production. That is, the facility produces according to a forecast of customer demand, and completed chips enter a finished goods inventory, which in turn services actual customer demand. Also, wafer fabrication facilities often have only one bottleneck station (the photolithography workstation, see the simulation studies of Atherton and Dayhoff 1985, Glassey and Recende 1988 and Wein 1988), and wafers usually visit this station ten to twenty times during processing.

In this paper, we will focus on the bottleneck workstation and consider the scheduling problem faced by a single-station, make-to-stock production facility in a dynamic stochastic environment. The facility employs K different processes to produce K types of products, where each product type has a designated process that is primarily used to produce it. A process can have multiple stages of service at the workstation, and we model the workstation as a single-server multiclass feedback queue. The server represents the photolithography machine and the entities populating the queue will be referred to as jobs, where each job represents a lot of wafers. A different job class is defined for each service stage of each process type, and each job class is allowed to have its own general service time distribution. If there are J job classes, the dynamic scheduling decisions consist of choosing among $J+1$ options at each point in time: either serve a class j job, $j = 1, ..., J$, or allow the server to be idle. Preemptive resume scheduling is allowed, but as will be seen later, our method of analysis is crude enough that the resulting scheduling policy is independent of the particular assumptions made with regard to preemption. We assume an ample supply of raw wafers is available for each process type, and a lot of raw wafers is released to the queueing system whenever the scheduler decides to start working on a new job. Also, no set-up times or costs are incurred when switching production from one job class to another. Photolithography operations generally require a new set-up for each lot of wafers, regardless of product type or stage of service, and thus set-up times can be incorporated into the processing times.

A partial ordering is assumed to exist among the K product types with respect to quality. When a job's processing is complete, the wafers in the job are sliced into individual chips and the product type (that is, quality grade) of each chip is determined. The product type of each completed chip is random and depends on the process used to make the chip, but is independent of the product type of other chips in the job and chips in previously completed jobs. In particular, we assume that a given process can only produce chips of its corresponding product type and all lower quality types (including chips that need to be discarded). Completed chips of each product enter a finished goods inventory that services the exogenous demand for that product. The cumulative demand for each product (measured in chips) is modelled as an arbitrary point process that satisfies a functional central limit theorem (for example, a compound Poisson process), and all unsatisfied demand is backordered. Note that the entities in the queue (which will also be referred to as work-in-process, or WIP, inventory) and finished goods inventory are measured in different units; WIP is measured in lots of wafers and finished goods inventory is measured in chips.

The scheduler can observe the current queue length (or WIP inventory) of each job class and finished goods inventory of each product type before making a scheduling decision, and the objective of the scheduling problem is to minimize the long run expected average cost, which includes linear costs for holding WIP inventory (which may differ by job class) and linear costs for backordering and holding finished goods inventory (which may differ by product type).

This paper is a sequel to Wein (1990), who considers the same production/inventory scheduling problem, but assumes that all production processes have perfect yield, produce no by-products and possess only one stage of service. Because this scheduling problem appeared to be difficult to analyze in its exact form, Wein (1990) employed a Brownian model developed by Harrison (1988) that approximates, under so-called *heavy traffic conditions*, a dynamic scheduling problem for a multiclass queueing system by a dynamic control problem involving Brownian motion. The heavy traffic condition assumes that the server must be busy the great majority of the time (for example, 90% of the time) in order to satisfy customer demand over the long run. Wein (1990) derived a closed form solution to an equivalent reformulation, called the workload formulation, of the Brownian control problem and interpreted the solution in terms of the original production/inventory system in order to develop an effective scheduling policy, and we follow the same procedure here.

As explained in Wein (1990), the rescaling of time and space that occurs in the approximating procedure prevents us from deriving an explicit detailed scheduling policy to the actual problem. Thus, we propose a parametric policy and say that a product is *in danger of being backordered* if its current finished goods inventory level is less than a certain critical level. These levels, one for each product type, are not derived from our analysis, but are instead parameter values chosen by the scheduler. In the simulation studies here and in Wein (1990), the best performance for our particular examples was achieved with critical values less than or equal to the number of chips in two lots of wafers.

Our analysis leads to the concept of *expected effective resource consumption* for each product type. For the lowest quality product (or, more generally, for any product whose designated process produces no by-products), this quantity is simply the total expected processing time to produce a completed lot of wafers using the corresponding designated process *divided by* the expected number of chips of that product type in a completed lot. For a higher quality product, the expected effective resource consumption equals the corresponding ratio for this product *minus* the expected number of chips of each lower quality product in a completed lot of wafers (normalized by the expected number of chips of the higher quality product in a completed lot) times the expected effective resource consumption for the corresponding lower quality product. Thus, the expected effective resource consumption for a product type represents the expected amount of machine time consumed to produce one unit of that product for finished goods inventory, after accounting for the random production of all other product types. For scheduling purposes, this quantity plays the role of a product's *expected total processing time* in a production system that produces by-products and defective products; it may also be useful for cost accounting purposes.

The proposed parametric scheduling policy dynamically tracks the *weighted inventory process*, which is the sum of a linear combination of the current WIP and finished goods inventory, where the WIP of each job class is weighted by the expected processing time already received by jobs of that class and the finished goods inventory of each product is weighted by its expected effective resource consumption. Thus, the weighted inventory process represents the machine time that has already been invested in the current WIP and finished goods inventory. Whenever the weighted inventory process is above a certain critical level and there are no products in danger of being backordered, the machine sits idle; otherwise, the machine is kept busy. When the machine is working, our dynamic scheduling policy is reminiscent of the so-called $c\mu$ rule (see, for example, Klimov 1974) that minimizes the weighted average cycle time in a conventional multiclass queue, in that it employs an index policy for each job class and product type that measures a cost divided by an expected processing time; in the $c\mu$ rule, c_k is the holding cost for class k jobs in queue and μ_k is the service rate, and the rule gives priority to larger values of the index $c_k\mu_k$. In our setting, if there are any products in danger of being backordered, the scheduler attempts to satisfy the backordered demand for the product that has the largest ratio of its backorder cost divided by its expected effective resource consumption. If no products are in danger of being backordered and the machine needs to stay busy, then the scheduler tries to keep all its *excess inventory* in one location, either as WIP of a certain job class or as finished goods inventory of a certain product. In particular, an index is calculated for each of the J job classes and K products, and the work is kept in the location that corresponds to the smallest of these $J + K$ indices. The index for each job class is the WIP holding cost for that class divided by the expected processing time already received by the class, and the index for each product type is the product's finished goods holding cost divided by its expected effective resource consumption. Thus, excess inventory is held in the location that achieves the smallest holding cost per unit of expected machine time already invested.

A simulation experiment is performed with four numerical examples, and the proposed policy is compared to two policies that keep the workstation busy when the total (unweighted) finished goods inventory process is below a critical level or when products are backordered; one policy dynamically awards priority to the class with the smallest finished goods inventory level and the other policy is more complex and is described in Section 6. The cost reductions achieved by the proposed policy (relative to the two other policies) range from 7.3% to 20.3% in the four numerical examples.

Although there is a huge literature on scheduling traditional make-to-order queueing systems, Wein (1990) appears to be the first paper to explicitly analyze a dynamic scheduling problem for a multiclass make-to-stock queue, and readers are referred there for a brief review of the control and scheduling of stochastic production/inventory systems. Pierskella and Deuermeyer (1978) are among the first to consider the control of a production/inventory system with by-production, and earlier works on this problem are sited there. These authors examine a system with two processes, where process A produces both products in some known deterministic proportion, while process B produces only one product. They formulate the problem as a dynamic program, derive properties of the optimal policy, and show that the two-dimensional state space can be divided into 4 regions, where a different control is exerted in each region. Another related paper is Courcoubetis et al. (1989), who investigate a production system of m machines that can each produce n product types, but according to different probabilities. If a randomly arriving demand for a particular product type cannot be satisfied by the current finished goods inventory, then the scheduler must decide which machine to use in order to satisfy this demand. This machine continues working until the demand is satisfied, and all other products produced by this machine enter the finished goods inventory. They find necessary and sufficient conditions for the existence of a scheduling policy that makes the inventory finitely bounded, and provide such a strategy when these conditions are satisfied. Bitran and Dasu (1989) and Bitran and Leong (1989) study production planning problems for a process that randomly yields a variety of different products. Their models consider only a single production process (and hence no scheduling options exist), but finished units of one product can be used to satisfy customer demand for a lower quality product. They consider the joint decisions of choosing a production quantity and allocating finished goods inventory to customers, and propose several heuristics for solving the multi-period problems. It should be noted that all of the existing papers on scheduling production/inventory systems with by-production assume *instantaneous* production, whereas we explicitly model the production system as a multiclass feedback queue.

Although we are motivated by a specific industry, our analysis may also be applied to other examples of random production, such as fiber optics (the segment length may be random), ingot cutting in the steel industry (where the size may be random), crystal cutting in electronics applications (where the frequency may be random) and blending problems in the petroleum industry (where random quantities of by-products may be obtained). The remainder of this paper is organized as follows. The scheduling problem is formulated in Section 2 and the corresponding Brownian control problem is defined in Section 3. In Section 4, the latter problem is reformulated in terms of workloads, and the workload formulation is solved. The solution is interpreted in Section 5 to propose a scheduling policy, and the policy is tested on two systems in Section 6, a by-production system with random yield and no job reentry (that is, no feedback), and a reentry system with perfect yield and no by-production.

2. Problem Formulation

Before stating the problem, we will describe the probabilistic formalisms that will be adopted in this paper. When we say that X is a $K-$dimensional (μ, Σ) Brownian motion (readers are referred to Karatzas and Shreve 1988 for a definition), it is assumed that $(\Omega, \mathbf{F}, \mathbf{F_t}, X, P_x)$ is given, where $(\Omega, \mathbf{F})$ is a measurable space, $X = X(\omega)$ is a measurable mapping of Ω into $\mathbf{C}(\mathbf{R^K})$, which is the space of continuous functions on $\mathbf{R^K}$, $\mathbf{F_t} = \sigma(X(s), s \leq t)$ is the filtration generated by X, and P_x is a family of probability measures on Ω such that the process $\{X(t), t \geq 0\}$ is a Brownian motion with drift μ, covariance matrix Σ, and initial state x under P_x. Let E_x be the expectation operator associated with P_x. If $Y = \{Y(t), t \geq 0\}$ is a process that is $\mathbf{F_t}$-measurable for all $t \geq 0$, then we say that the process Y is *nonanticipating* with respect to the Brownian motion X. More generally, we will say that one process Y is nonanticipating with respect to another process X when Y is adapted to the coarsest filtration with respect to which X is adapted.

We consider a facility that produces K types of products. Each product type has its own independent demand process denoted by $D_k(t)$, which is the cumulative amount of demand for type k products up to time t. We assume that the process D_k satisfies a functional central limit theorem and has asymptotic rate λ_k and variance a_k^2. The model can actually accommodate dependencies among the various product's demands (as long as the vector demand process satisfies a functional central limit theorem), but this issue will not be pursued here. The facility can produce these K product types according to K different production processes. The K processes will usually be indexed by $j = 1, ..., K$ in order to distinguish them from the K product types, which will be indexed by $k = 1, ..., K$. Process j has $l(j)$ stages of service at the workstation, and these service stages are indexed by $i = 1, ..., l(j)$. The entities residing in the queueing system will be referred to as jobs rather than customers, so as not to confuse them with the actual customer demand. As in Kelly (1979), we define a different job class for each combination of process type and stage of completion. There are $\sum_{j=1}^{K} l(j)$ different job classes and each class is denoted by the pair (j, i), which is its process type and stage of completion. Thus, customers change class in a deterministic fashion as they proceed through the system. More generally, we can allow probabilistic routing for each process type, but the notation required to perform the subsequent analysis would be significantly more tedious. Each job class (j, i) has its own general service time distribution with mean m_{ji} and variance s_{ji}^2, and the service rate is denoted by $\mu_{ji} = m_{ji}^{-1}$. Let $\{S_{ji}(t), t \geq 0\}$ be the renewal process corresponding to the service times of class (j, i), so that $S_{ji}(t)$ represents the number of class (j, i) job completions up to time t if the server were continuously working on this job class during the interval $[0, t]$. We also define $M_{ji} = \sum_{l=1}^{i-1} m_{jl}$ to be the total expected service time that a class (j, i) job has received from the workstation, and $M_j = \sum_{i=1}^{l(j)} m_{ji}$ to be the total expected service time for jobs produced acccording to process type j.

To complete the specification of the state dynamics, we need to describe how the processes randomly produce products. We assume that a partial ordering of the product types exists with respect to quality, and the product types are indexed so that if two products are comparable, then the one of higher quality is given a larger index. Each product type has a designated process that is primarily used to produce it; in particular, process k is primarily used to produce product k, for $k = 1, ..., K$. However, each process may also produce lower quality products and defective products. Each completed job (or lot of wafers) produced according to process j contains L_j items (or chips) that can potentially be placed in the finished goods inventory, and we assume that each of these items is a unit of product k with probability p_{jk}, and is a defective item with probability $p_{j0} = 1 - \sum_{k=1}^{K} p_{jk}$, independent of all other items in the completed job. We call the matrix $P = (L_j p_{jk} : j, k = 1, \ldots, K)$ the *by-production matrix*. Since the product types are partially ordered, we have $p_{jk} = 0$ for $k > j$, and hence the by-production matrix P is lower triangular with positive diagonal elements.

At time t, the state of the system is described by $Q_{ji}(t), j = 1, ..., K$ and $i = 1, ..., l(j)$, which is the number of class (j, i) jobs in queue or in service, and $Z_k(t), k = 1, ..., K$, which is the number of units of product k in the finished goods inventory. The vector processes $Q = (Q_{ji})$ and $Z = (Z_k)$ will be referred to as the *WIP process* and *inventory process*, respectively. The system cost consists of linear WIP holding, inventory backordering and inventory holding costs, where c_{ji} represents the WIP holding cost for class (j, i) jobs, b_k represents the backorder cost for product k units, and h_k is the finished goods holding cost for product k units. It will be assumed that $b_k > h_k > 0$ for $k = 1, ..., K$, and for future use we define the convex finished goods inventory cost function $g_k, k = 1, ..., K$, by

$$g_k(x) = \begin{cases} -b_k x & \text{if } x \leq 0, \\ h_k x & \text{if } x > 0. \end{cases} \tag{2.1}$$

In order to employ the approximating Brownian procedure, we express the scheduling decisions in terms of cumulative *allocation processes*, $\{T_{ji}(t), t \geq 0\}$, for $j = 1, ..., K$ and $i = 1, ..., l(j)$, where $T_{ji}(t)$ is the total amount of time that the workstation has devoted to class (j, i) jobs up to time t. We assume an adequate supply of raw materials (that is, raw wafers) is available whenever the scheduler decides to begin processing a new job. Since raw material inventory is not included in our model, $Q_{j1}(t)$ is never larger than one for all processes $j = 1, ..., K$. Because of the rescaling that takes place in the Brownian approximation (see (3.10)), no holding costs will be incurred for classes $(j, 1)$, $j = 1, ..., K$, and thus $Q_{j1}(t), j = 1, \ldots, K$, can be omitted from the problem formulation.

Define

$$A_j(t) = S_{jl(j)}(T_{jl(j)}(t)) \quad \text{for } j = 1, \ldots, K, \tag{2.2}$$

so that $A_j(t)$ represents the total number of jobs (or lots of wafers) completed according to process j during the interval $[0, t]$. Furthermore, for each process j, define a doubly-indexed sequence of independent and identically distributed random variables q^j_{nm}, $n = 1, 2, \ldots$ and $m = 1, \ldots, L_j$. Each q^j_{nm} is equal to k with probability p_{jk}, $k = 0, 1, \ldots, K$. Let $\chi(\cdot)$ be an indicator function defined by

$$\chi(x = y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{if } x \neq y. \end{cases} \tag{2.3}$$

If we assume that $Z(0) = Q(0) = T(0) = 0$, then the state equations of the system are

$$Q_{ji}(t) = S_{j,i-1}(T_{j,i-1}(t)) - S_{ji}(T_{ji}(t)) \quad \text{for } j = 1, \ldots, K, i = 2, \ldots, l(j) \text{ and } t \geq 0, \tag{2.4}$$

and

$$Z_k(t) = \sum_{j=1}^{K} \sum_{n=1}^{A_j(t)} \sum_{m=1}^{L_j} \chi(q^j_{nm} = k) - D_k(t) \quad \text{for } k = 1, \ldots, K \text{ and } t \geq 0. \tag{2.5}$$

We assume that the allocation process T is nonanticipating with respect to the WIP process Q and the inventory process Z, which implies that the scheduler cannot observe actual demands, service times and process realizations before they occur. The allocation process must also be continuous and nondecreasing in time. Furthermore, if we define the *cumulative idleness process* $I(t)$ to be the cumulative amount of time the server is idle in $[0, t]$, then

$$I(t) = t - \sum_{j=1}^{K} \sum_{i=1}^{l(j)} T_{ji}(t) \text{ for } t \geq 0, \tag{2.6}$$

which must also be nondecreasing.

Thus, the scheduling problem is:

$$\min_{T_{ji}} \quad \limsup_{T \to \infty} \frac{1}{T} E\left[\int_0^T \sum_{j=1}^{K} \sum_{i=2}^{l(j)} c_{ji} Q_{ji}(t) + \sum_{k=1}^{K} g_k(Z_k(t))\right] \tag{2.7}$$

$$\text{subject to} \quad Q_{ji}(t) = S_{j,i-1}(T_{j,i-1}(t)) - S_{ji}(T_{ji}(t))$$

$$\text{for } j = 1, \ldots, K, \ i = 2, \ldots, l(j) \text{ and } t \geq 0, \tag{2.8}$$

$$Z_k(t) = \sum_{j=1}^{K} \sum_{n=1}^{A_j(t)} \sum_{m=1}^{L_j} \chi(q^j_{nm} = k) - D_k(t)$$

$$\text{for } k = 1, \ldots, K \text{ and } t \geq 0, \tag{2.9}$$

$$I(t) = t - \sum_{j=1}^{K} \sum_{i=1}^{l(j)} T_{ji}(t) \text{ for } t \geq 0, \tag{2.10}$$

$$Q(t) \geq 0 \text{ for } t \geq 0, \tag{2.11}$$

$$I \text{ is nondecreasing for } t \geq 0, \tag{2.12}$$

$$T \text{ is continuous and nondecreasing in } t, \text{ and} \tag{2.13}$$

$$T \text{ is nonanticipating with respect to } Q \text{ and } Z. \tag{2.14}$$

3. The Limiting Control Problem

In this section, we follow the approach taken in Sections 3 through 5 of Harrison (1988) to develop a Brownian approximation to the control problem (2.7)-(2.14). Only the basics of the approximation are provided, and readers are referred to Harrison (1988) for a more detailed presentation and justification. Let $\beta_k, k = 1, ..., K$, denote the rate at which jobs of class $(k, l(k))$, which corresponds to the last stage of process k, are completed per unit of time over the long run. Then the vector $\beta = (\beta_k)$ must satisfy

$$\lambda = P^T \beta \tag{3.1}$$

in order for the average customer demand for all products to be satisfied exactly over the long run, where $\lambda = (\lambda_k)$ is the vector of exogenous demand rates for each product type. Since P is a lower triangular matrix with positive diagonal elements, the flow balance equations (3.1) uniquely determine a vector β. However, for the system to be stable, we need to assume that

$$\beta_k \geq 0 \text{ for } k = 1, \ldots, K. \tag{3.2}$$

If any components of β are negative, then in order to keep up with the long run average demand of some high quality product type, the system will inevitably overproduce at least one lower quality product type, and the finished goods inventory of the lower quality product will grow infinitely large. Fortunately, production facilities often have more possible production processes than product types, in which case a subset of the production processes should be employed so that (3.2) is satisfied and system instability is avoided. Condition (3.2) is similar to the necessary and sufficient conditions for stability derived in Theorem 2.1 of Courcoubetis et al. (1989).

By equation (3.1), the proportion of time over the long run that the workstation must work on class (j,i) jobs is $\rho_{ji} = \beta_j m_{ji}$. Thus, the *traffic intensity* of the system, that is, the long run average workstation utilization, is $\rho = \sum_{j=1}^{K} \sum_{i=1}^{l(j)} \rho_{ji}$. For $j = 1, ..., K$ and $i = 1, ..., l(j)$, let $\alpha_{ji} = \rho_{ji}/\rho$, and define the *centered* allocation and service processes, respectively, by

$$Y_{ji}(t) = \alpha_{ji}t - T_{ji}(t) \quad \text{for } t \geq 0 \tag{3.3}$$

and

$$\eta_{ji}(t) = S_{ji}(t) - \mu_{ji}t \quad \text{for } t \geq 0. \tag{3.4}$$

To simplify the formulas for the WIP process Q and the inventory process Z, we define

$$X_{ji}(t) = \eta_{j,i-1}(T_{j,i-1}(t)) - \eta_{ji}(T_{ji}(t)) \quad \text{for } t \geq 0 \tag{3.5}$$

and

$$\begin{aligned} X_k(t) &= \sum_{j=1}^{K} \sum_{n=1}^{A_j(t)} \sum_{m=1}^{L_j} (\chi(q^j_{nm} = k) - p_{jk}) + \sum_{j=1}^{K} L_j p_{jk} \eta_{jl(j)}(T_{jl(j)}(t)) \\ &\quad - D_k(t) + \lambda_k t + \frac{\lambda_k}{\rho}(1-\rho)t \quad \text{for } t \geq 0. \end{aligned} \tag{3.6}$$

Then it follows from (2.4)-(2.6) and (3.3)-(3.6) that

$$\begin{aligned} Q_{ji}(t) &= X_{ji}(t) + \mu_{ji}Y_{ji}(t) - \mu_{j,i-1}Y_{j,i-1}(t) \\ &\qquad \text{for } j = 1, \ldots, K,\ i = 2, \ldots, l(j) \text{ and } t \geq 0, \end{aligned} \tag{3.7}$$

$$Z_k(t) = X_k(t) - \sum_{j=1}^{K} L_j p_{jk} \mu_{jl(j)} Y_{jl(j)}(t) \quad \text{for } k = 1, \ldots, K \text{ and } t \geq 0, \tag{3.8}$$

and

$$I(t) = \sum_{j=1}^{K} \sum_{i=1}^{l(j)} Y_{ji}(t) \quad \text{for } t \geq 0. \tag{3.9}$$

As in Harrison (1988), the key to the approximation is to replace the allocation process $T_{ji}(t)$ in (3.5)-(3.6) by its nominal allocation process $\rho_{ji}t$. Readers are referred to Sections 5 and 11 of Harrison (1988) for an informal defense of this substitution. After this replacement, we assume that the production/inventory system is under *heavy traffic conditions*, which require the existence of a larger integer n that approximately equals $(1-\rho)^{-2}$. A representive example is to choose $n = 100$ if the traffic intensity $\rho = 0.9$. The system parameter n is used to rescale the processes, but the rescaled processes will retain the same notation as the original processes in order to reduce the notational burden. For $t \geq 0$, let

$$Q_{ji}(t) = \frac{Q_{ji}(nt)}{\sqrt{n}}, \tag{3.10}$$

$$Z_k(t) = \frac{Z_k(nt)}{\sqrt{n}}, \tag{3.11}$$

$$Y_{ji}(t) = \frac{Y_{ji}(nt)}{\sqrt{n}}, \tag{3.12}$$

$$I(t) = \frac{I(nt)}{\sqrt{n}}, \tag{3.13}$$

$$X_{ji}(t) = \frac{X_{ji}(nt)}{\sqrt{n}}, \tag{3.14}$$

and

$$X_k(t) = \frac{X_k(nt)}{\sqrt{n}}. \tag{3.15}$$

The approximating Brownian control problem is obtained by letting the parameter n tend to infinity. The processes Q, Z, Y, and I now represent limiting scaled processes, although they will still be referred to simply as the WIP, inventory, allocation and cumulative idleness processes, respectively. A straightforward application of weak convergence results (see Billingsley 1968) reveals that the limiting processes X_{ji} and X_k are Brownian motion processes. Define $v_{ji} = \mu_{ji}^2 s_{ji}^2$ and $v_k = {\lambda_k}^2 a_k^2$, so that v_{ji} and v_k represent the squared coefficients of variation corresponding to the service process of class (j, i) and the demand process of product k, respectively. Then the drift and variance of the Brownian motion processes X_{ji} and X_k are given by

$$\begin{aligned} E[X_{ji}] &= 0 \text{ for } j = 1, \ldots, K \text{ and } i = 2, \ldots, l(j), && (3.16) \\ Var[X_{ji}] &= \beta_j(v_{j,i-1} + v_{ji}) \ \text{ for } j = 1, \ldots, K \text{ and } i = 2, \ldots, l(j), && (3.17) \end{aligned}$$

$$\begin{aligned} E[X_k] &= \frac{\lambda_k}{\rho}\sqrt{n}(1-\rho) \ \text{ for } k = 1, \ldots, K, && (3.18) \\ Var[X_k] &= \lambda_k v_k + \sum_{j=1}^{K} \beta_j L_j p_{jk}(1 - p_{jk} + L_j p_{jk} v_{jl(j)}) \ \text{ for } k = 1, \ldots, K, && (3.19) \end{aligned}$$

and their covariances are given by

$$Cov[X_{ji}, X_{jl}] = \begin{cases} -\beta_j v_{ji} & \text{if } l = i+1, \\ 0 & \text{otherwise,} \end{cases} \qquad (3.20)$$

$$Cov[X_{ji}, X_{kl}] = 0 \text{ if } j \neq k, \qquad (3.21)$$

$$Cov[X_{ji}, X_k] = \begin{cases} -\beta_j L_j p_{jk} v_{jl(j)} & \text{if } i = l(j), \\ 0 & \text{otherwise,} \end{cases} \qquad (3.22)$$

$$Cov[X_k, X_l] = \sum_{j=1}^{K} \beta_j L_j p_{jk} p_{jl}(L_j v_{jl(j)} - 1)$$
$$\text{for } k, l = 1, \ldots, K \text{ and } k \neq l. \qquad (3.23)$$

We are now in a position to state the limiting control problem, which is to choose a RCLL (right continuous with left limits) process $Y = (Y_{ji})$ to

$$\text{minimize} \quad \limsup_{T\to\infty} \frac{1}{T} E[\int_0^T \sum_{j=1}^{K} \sum_{i=2}^{l(j)} c_{ji} Q_{ji}(t) + \sum_{k=1}^{K} g_k(Z_k(t))] \qquad (3.24)$$

$$\text{subject to} \quad Q_{ji}(t) = X_{ji}(t) + \mu_{ji} Y_{ji}(t) - \mu_{j,i-1} Y_{j,i-1}(t)$$
$$\text{for } j = 1, \ldots, K, \ i = 2, \ldots, l(j) \text{ and } t \geq 0, \qquad (3.25)$$

$$Z_k(t) = X_k(t) - \sum_{j=1}^{K} L_j p_{jk} \mu_{jl(j)} Y_{jl(j)}(t)$$

$$\text{for } k = 1, \ldots, K \text{ and } t \geq 0, \tag{3.26}$$

$$I(t) = \sum_{j=1}^{J} \sum_{i=1}^{l(j)} Y_{ji}(t) \text{ for } t \geq 0, \tag{3.27}$$

$$Q(t) \geq 0 \text{ for } t \geq 0, \tag{3.28}$$

$$I \text{ is nondecreasing for } t \geq 0, \text{ and} \tag{3.29}$$

$$Y \text{ is nonanticipating with respect to } X_{ji} \text{ and } X_k$$

$$\text{for } j = 1, \ldots, K,\ i = 2, \ldots, l(j) \text{ and } k = 1, \ldots, K. \tag{3.30}$$

4. The Workload Formulation and Its Solution

The limiting control problem (3.24)-(3.30) has an equivalent alternative formulation in which the control process $Y = (Y_{ji})$ is not explicitly exposed and only a one-dimensional Brownian motion process is involved. Recall that M_{ji} is the total expected service time that a class (j, i) job has received, and M_j is the total expected service time for process j jobs. Denote the K-vector (M_j) by M, and define the K-vector $M^* = (M_k^*)$ by

$$M^* = P^{-1} M. \tag{4.1}$$

As will be discussed in the next section, M_k^* represents the expected effective resource consumption (machine time consumed) for producing one unit of type k product according to process k. In order to have a practically meaningful solution to the workload formulation, we assume that

$$M_k^* > 0 \text{ for } k = 1, \ldots, K. \tag{4.2}$$

We will discuss the implications of this assumption in the next section, and provide a set of conditions under which this assumption is satisfied.

Define the one-dimensional Brownian motion B by

$$B(t) = \sum_{k=1}^{K} M_k^* X_k(t) + \sum_{j=1}^{K}\sum_{i=2}^{l(j)} M_{ji}X_{ji}(t) \text{ for } t \geq 0. \tag{4.3}$$

By (3.16)-(3.23), the drift of B is $\sqrt{n}(1-\rho) > 0$ and the variance of B is

$$\begin{aligned}\sigma^2 &= \sum_{k=1}^{K} M_k^{*2}\{\lambda_k v_k + \sum_{j=1}^{K} \beta_j L_j p_{jk}(1 - p_{jk} + L_j p_{jk} v_{jl(j)})\} \\ &\quad + \sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{j=1}^{K} M_k^* M_l^* \beta_j L_j p_{jk} p_{jl}(L_j v_{jl(j)} - 1) \\ &\quad - \sum_{k=1}^{K}\sum_{j=1}^{K} 2M_k^* M_{jl(j)} L_j p_{jk} \beta_j v_{jl(j)} \\ &\quad + \sum_{j=1}^{K}\sum_{i=2}^{l(j)}\{M_{ji}^2 \beta_j (v_{j,i-1} + v_{ji}) - 2M_{ji}M_{j,i-1}\beta_j v_{j,i-1}\}.\end{aligned} \tag{4.4}$$

The *workload formulation* of the limiting control problem (3.24)-(3.30) is to choose RCLL processes $Q = (Q_{ji}), Z = (Z_k)$ and I to

$$\text{minimize} \quad \limsup_{T\to\infty} \frac{1}{T} E[\int_0^T \sum_{j=1}^{K}\sum_{i=2}^{l(j)} c_{ji}Q_{ji}(t) + \sum_{k=1}^{K} g_k(Z_k(t))] \tag{4.5}$$

$$\text{subject to} \quad \sum_{k=1}^{K}\sum_{i=2}^{l(j)} M_{ji}Q_{ji}(t) + \sum_{k=1}^{K} M_k^* Z_k(t) = B(t) - I(t) \text{ for } t \geq 0, \tag{4.6}$$

$$Q(t) \geq 0 \text{ for } t \geq 0, \tag{4.7}$$

$$I(t) \text{ is nondecreasing in t, and} \tag{4.8}$$

$$Q, Z \text{ and } I \text{ are nonanticipating with respect to } X_{ji} \text{ and } X_k. \tag{4.9}$$

Proposition 1. *Every feasible policy Y for the limiting control problem* (3.24)-(3.30) *yields a corresponding feasible policy (Q, Z, I) for the workload formulation* (4.5)-(4.9), *and every feasible policy (Q, Z, I) for the workload formulation* (4.5)-(4.9) *yields a corresponding feasible policy Y for the limiting control problem* (3.24)-(3.30). *Furthermore, the long run average expected costs of the corresponding policies are the same.*

Proof. Let Y be a feasible policy for the limiting control problem, and define Q, Z and I by

$$Q_{ji}(t) = X_{ji}(t) + \mu_{ji}Y_{ji}(t) - \mu_{j,i-1}Y_{j,i-1}(t) \quad \text{for } j = 1, \ldots, K, i = 2, \ldots, l(j) \text{ and } t \geq 0, \tag{4.10}$$

$$Z_k(t) = X_k(t) - \sum_{j=1}^{K} L_j p_{jk} \mu_{jl(j)} Y_{jl(j)}(t) \quad \text{for } k = 1, \ldots, K \text{ and } t \geq 0, \tag{4.11}$$

and

$$I(t) = \sum_{j=1}^{J} \sum_{i=1}^{l(j)} Y_{ji}(t) \text{ for } t \geq 0. \tag{4.12}$$

By the definition of the workload formulation, we see that (Q, Z, I) is a feasible policy for the workload formulation. Since the processes Q and Z are the same for the two policies, the costs are also the same.

Conversely, let (Q, Z, I) be a feasible policy for the workload formulation, and define Y to be the unique solution to the following system of linear equations for each $t \geq 0$:

$$Q_{ji}(t) = X_{ji}(t) + \mu_{ji}Y_{ji}(t) - \mu_{j,i-1}Y_{j,i-1}(t) \quad \text{for } j = 1, \ldots, K \text{ and } i = 2, \ldots, l(j), \tag{4.13}$$

and

$$Z_k(t) = X_k(t) - \sum_{j=1}^{K} L_j p_{jk} \mu_{jl(j)} Y_{jl(j)}(t) \quad \text{for } k = 1, \ldots, K. \tag{4.14}$$

Since the matrix P is invertible, a unique solution $Y_{jl(j)}, j = 1, ..., K$, exists to equation (4.14). For each $j = 1, ..., K$, the solution $Y_{j,l(j)}$ can be substituted into (4.13) and these $l(j) - 1$ equations can be solved sequentially from $i = l(j)$ to $i = 2$, where the i^{th} equation yields the unique solution $Y_{j,i-1}$. Thus, we have verified equations (3.25)-(3.26), (3.28) and (3.30). Substituting the right sides of (4.13) and (4.14) into (4.6) gives

$$I(t) = \sum_{j=1}^{J} \sum_{i=1}^{l(j)} Y_{ji}(t). \tag{4.15}$$

This and (4.8) establish (3.27) and (3.29). The cost is again the same because the derived solution to the limiting control problem has the same processes Q and Z. ∎

The workload formulation is solved in two steps. First, the optimal control processes Q and Z are derived in terms of the control process I using a convex program, and then the optimal control process I is found. Define the one-dimensional *weighted inventory process* W by

$$W(t) = B(t) - I(t) \text{ for } t \geq 0. \tag{4.16}$$

By (4.6), process W is a weighted sum of the components of the WIP process Q and the inventory process Z. For any given cumulative idleness process I, we can find a pathwise optimal WIP process Q and inventory process Z by solving the following convex program at every time moment $t \geq 0$:

$$\min_{Q(t),Z(t)} \quad \sum_{j=1}^{K}\sum_{i=2}^{l(j)} c_{ji}Q_{ji}(t) + \sum_{k=1}^{K} g_k(Z_k(t)) \tag{4.17}$$

$$\text{subject to} \quad \sum_{k=1}^{K}\sum_{i=2}^{l(j)} M_{ji}Q_{ji}(t) + \sum_{k=1}^{K} M_k^* Z_k(t) = W(t) \tag{4.18}$$

$$Q_{ji}(t) \geq 0 \text{ for } j = 1, \dots, K \text{ and } i = 2, \dots, l(j). \tag{4.19}$$

Let $Z_k^+(t)$ and $Z_k^-(t)$ represent the positive and negative parts of $Z_k(t)$ for $k = 1, \dots, K$. Then the convex program (4.17)-(4.19) is equivalent to the following linear program:

$$\min_{Q(t),Z^+(t),Z^-(t)} \quad \sum_{j=1}^{K}\sum_{i=2}^{l(j)} c_{ji}Q_{ji}(t) + \sum_{k=1}^{K} h_k Z_k^+(t) + \sum_{k=1}^{K} b_k Z_k^-(t) \tag{4.20}$$

$$\text{subject to} \quad \sum_{k=1}^{K}\sum_{i=2}^{l(j)} M_{ji}Q_{ji}(t) + \sum_{k=1}^{K} M_k^* Z_k^+(t) - \sum_{k=1}^{K} M_k^* Z_k^-(t) = W(t) \tag{4.21}$$

$$Q_{ji}(t) \geq 0 \text{ for } j = 1, \dots, K \text{ and } i = 2, \dots, l(j), \tag{4.22}$$

$$Z_k^+(t) \geq 0 \text{ and } Z_k^-(t) \geq 0 \text{ for } k = 1, \dots, K. \tag{4.23}$$

Notice that the form of the cost function g_k in (2.1) guarantees that

$$Z_k^+(t) Z_k^-(t) = 0 \text{ for } k = 1, \dots, K, \tag{4.24}$$

as required.

Analysis of (4.20)-(4.23) yields a closed formed solution to (4.17)-(4.19). To describe the solution, we define

$$h = \min\{\frac{c_{ji}}{M_{ji}}, \frac{h_k}{M_k^*} : j,k = 1,\ldots,K \text{ and } i = 2,\ldots,l(j)\} \tag{4.25}$$

$$b = \min\{\frac{b_k}{M_k^*} : k = 1,\ldots,K\}, \text{ and} \tag{4.26}$$

$$f(x) = \begin{cases} hx & \text{if } x \geq 0, \\ -bx & \text{if } x < 0. \end{cases} \tag{4.27}$$

Then the optimal solution $(Q_{ji}^*(t), Z_k^*(t))$ to (4.17)-(4.19) is

$$Q_{ji}^*(t) = \begin{cases} \frac{W(t)}{M_{ji}} & \text{if } h = \frac{c_{ji}}{M_{ji}} \text{ and } W(t) \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{4.28}$$

$$Z_k^*(t) = \begin{cases} \frac{W(t)}{M_k^*} & \text{if } h = \frac{h_k}{M_k^*} \text{ and } W(t) \geq 0, \\ \frac{W(t)}{M_k^*} & \text{if } b = \frac{b_k}{M_k^*} \text{ and } W(t) \leq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{4.29}$$

where it is understood that when more than one variable can be nonzero at a given time t (due to a tie in the index values in (4.25)-(4.26)), only one of them is assigned a nonzero value and the others are set to zero. Accordingly, the optimal objective function value in (4.17) is $f(W(t))$.

Thus, in order to solve the workload formulation of the limiting control problem, we only need to solve the following one-dimensional singular Brownian control problem: choose a RCLL process I to

$$\text{minimize} \quad \limsup_{T\to\infty} \frac{1}{T} E[\int_0^T f(W(t))dt] \tag{4.30}$$

$$\text{subject to} \quad W(t) = B(t) - I(t) \text{ for } t \geq 0 \tag{4.31}$$

$$I \text{ is nondecreasing in } t \text{ and nonanticipating with respect to } B. \tag{4.32}$$

This problem is solved in Wein (1990), and the optimal solution is

$$I^*(t) = \sup_{0\leq s\leq t} [B(s) - c^*]^+ \text{ for } t \geq 0, \tag{4.33}$$

where

$$c^* = \frac{\sigma^2}{2\sqrt{n}(1-\rho)} \ln(1 + \frac{b}{h}). \tag{4.34}$$

Thus, in the idealized Brownian limit, the weighted inventory process W^* is a reflected, or regulated, Brownian motion (RBM) on the halfline $(-\infty, c^*]$, and the cumulative idleness process I^* acts as a reflecting barrier at c^*.

Furthermore, by (8.6) in Wein (1990), the objective function value of the workload formulation under the policy (Q^*, Z^*, I^*) is simply hc^*, where h is defined in (4.25). Hence, by the rescaling in (3.10)-(3.11), $\sqrt{n}hc^*$ is the predicted expected cost under the optimal policy. This closed form expression allows one to carry out *performance analysis* of the production/inventory system. For example, the relative cost reduction achieved by various process improvements (such as reducing the variability in the demand process and the processing times, increasing the processing rate, and introducing processes with different by-production probabilities) can be quickly assessed.

5. Interpreting The Solution

Recall that the Brownian model scales both time and the magnitude of the various stochastic processes. For example, if the server utilization ρ equals 0.9 and the system parameter n is chosen to be 100, and time is measured in units of hours, then by (3.11), $Z_k(t)$ would be the number of tens of class k jobs in inventory at time $100t$ hours. Although this scaling is too crude to give rise to an explicit scheduling policy for the original production/inventory system, we can still use the insights from the solution (Q^*, Z^*, I^*) to the workload formulation to develop an effective scheduling policy. After providing a rather literal interpretation of the solution, we will take into account the heavy traffic scaling and propose a refined solution that attempts to overcome the shortcomings of the Brownian model.

We begin by providing a recursive formula for calculating the vector M^* defined in (4.1). This formula allows us to easily interpret the components of M^*.

Proposition 2. *The components of* $M^* = (M_k^*)$ *can be calculated recursively from* $k = 1$ *to* K *by*

$$M_1^* = \frac{M_1}{L_1 p_{11}} \tag{5.1}$$

and

$$M_k^* = \frac{M_k}{L_k p_{kk}} - \sum_{l=1}^{k-1} \frac{p_{kl}}{p_{kk}} M_l^* \text{ for } k = 2, ..., K. \tag{5.2}$$

Proof. Recall that the by-production matrix P is lower triangular and possesses positive diagonal elements. Thus, the first equation in (4.1) is

$$M_1^* L_1 p_{11} = M_1, \tag{5.3}$$

which is equivalent to (5.1). Suppose (5.2) is true up to $k-1$ (it holds for $k = 1$ where the empty sum is zero). Then the k-th equation in (4.1) is

$$\sum_{l=1}^{k} L_k p_{kl} M_l^* = M_k, \tag{5.4}$$

which can be rewritten as (5.2). ∎

Recall that each product type has a designated process that is primarily used to produce it, but each process can also produce lower quality products and defective products. Since lower quality products are given smaller indices, it follows that process 1 can only produce units of product 1 and defective units. Each job (or lot of wafers) completed according to process 1 requires M_1 time units of total processing on average and yields on average $L_1 p_{11}$ units (or chips) of product 1. Thus, $M_1/(L_1 p_{11}) = M_1^*$ is the expected amount of time required for the workstation to produce one unit of product 1, and we say that M_1^* is the *expected effective resource consumption* for producing one unit of product 1 according to process 1. If M_l^* is the expected effective resouce consumption for producing one unit of product l according to process l for $l = 1, \ldots, k-1$, then the expected effective resource consumption for producing one unit of product k according to process k can be interpreted as follows. Since process k is designated to produce product k, $M_k/(L_k p_{kk})$ is the expected amount of server time utilized to produce one unit of product k. However, for each unit of product k produced according to process k, the workstation on average also produces p_{kl}/p_{kk} units of product l, for $l = 1, \ldots, k-1$. Since M_l^* is the effective amount of server time used to produce one unit of product l according to process l, for $l = 1, ..., k-1$, the expected effective resource consumption for producing one unit of product k according to process k is given by M_k^* in (5.2). In essence, each product is rewarded (by receiving a smaller value of M_k^*) for the lower quality products by-produced by its corresponding process. Although M_k^* can actually be associated with product k or process k, we will often refer to this quantity as product k's expected effective resource consumption.

Recall that assumption (4.2) requires each product to have a positive expected effective resource consumption. Thus, the nominal expected amount of resource consumed by a particular process to produce one unit of its corresponding product (which is $M_k/(L_k p_{kk})$ for process $k = 1, ..., K$) must be larger than the expected effective amount of resource imbedded in the average by-production of this process. The following conditions are sufficient for assumption (4.2).

Proposition 3. *If*

$$\frac{L_k p_{kk}}{M_k} > \frac{L_l p_{lk}}{M_l} \text{ for } 1 \leq k < l \leq K \tag{5.5}$$

and

$$\frac{p_{kj}}{p_{kl}} < \frac{p_{lj}}{p_{ll}} \text{ for } 1 \leq j < l < k \leq K \text{ and } p_{kl} > 0, p_{ll} > 0, \tag{5.6}$$

then assumption (4.2) *holds.*

Proof. By (5.1), it follows that $M_1^* > 0$. Setting $k = 1$ and $l = 2$ in (5.5) yields $M_2^* > 0$. Suppose $M_j^* > 0$ for $j = 1, \ldots, k-1$, and let l be the largest index less than k for which $p_{kl} > 0$. Then

$$\begin{aligned} M_k^* &= \frac{M_k}{L_k p_{kk}} - \frac{p_{kl}}{p_{kk}} M_l^* - \sum_{j=1}^{l-1} \frac{p_{kj}}{p_{kk}} M_j^* \\ &= \frac{M_k}{L_k p_{kk}} - \frac{M_l}{L_l p_{ll}} \frac{p_{kl}}{p_{kk}} + \sum_{j=1}^{l-1} \frac{M_j^*}{p_{kk}} \left(\frac{p_{lj}}{p_{ll}} p_{kl} - p_{kj}\right), \end{aligned} \tag{5.7}$$

which is positive by (5.5)-(5.6). ∎

Condition (5.5) implies that for each $k = 1, ..., K$, process k produces product k at a faster average rate than the other $K - 1$ processes. One would expect this restriction to hold since process k is designated to produce product k. To interpret condition (5.6), notice that P_{kj}/P_{kl} (P_{lj}/P_{ll}, respectively) is the expected number of product j units produced divided by the expected number of product l units produced when employing process k (process l, respectively). Since product j is of lower quality than product l, which in turn is of lower quality than product k, condition (5.6) says that a higher quality process produces relatively less lower quality products.

The state of the system in the workload formulation is the weighted inventory process W, which, by equations (4.6) and (4.16), is the sum of a linear combination of the WIP and finished goods inventory. The weight for class (j, i) WIP is M_{ji}, the total expected service time already received by a class (j, i) job, and the weight for type k product's finished goods inventory is M_k^*, the expected effective resource consumption for producing one unit of product k. Thus, $W(t)$ represents the machine time that is invested in the current WIP and finished goods inventory. Notice that backordered units can lead to a negative value of the weighted inventory process.

As discussed in Section 1, the scheduling decisions in the production/inventory system are to dynamically decide (1) whether to have the workstation idle or working, and (2) if the workstation is to be working, which job class should be served. The idle/busy policy is represented by the scaled cumulative idleness process I in the workload formulation, and the solution in (4.33) implies that I^* increases only when the weighted inventory process W reaches c^*. Let $w(t)$ be the actual (unscaled) weighted inventory process for the original production/inventory system. By (4.34) and the rescaling in (3.10)-(3.11), the workstation will be idle only when $w(t) \geq \sqrt{n}c^*$, or when

$$w(t) \geq \frac{\sigma^2}{2(1-\rho)} \ln(1 + \frac{b}{h}), \tag{5.8}$$

and will be kept busy otherwise. However, another feature of the solution affects the idle/work policy. The solution Z^* in (4.29) dictates that no products should be backordered when $W(t)$ is nonnegative. Taking this and the fact that $c^* \geq 0$ into consideration, the proposed idle/busy policy permits the workstation to be idle when $w(t) \geq \sqrt{n}c^*$ and no products are backordered, and keeps the workstation busy otherwise.

The priority sequencing decisions can be interpreted in terms of the WIP process Q^* in (4.28) and the finished goods inventory process Z^* in (4.29). This solution implies that at most one component of the total inventory (WIP and finished goods) process is nonzero at any point in time in the limiting Brownian model. In particular, if the weighted inventory $W(t)$ is negative, then no WIP or finished goods inventory is held, and backorders are all of the product type possessing the minimum ratio of b_k/M_k^*. This product type is relatively inexpensive to backorder and consumes a relatively large amount of effective machine time per unit produced. Thus, if $W(t)$ is negative, the backorders of this product type must receive lowest priority among the set of backordered products; that is, this product's backorders should only be satisfied if this product type is the only product that is backordered at time t. Under heavy traffic conditions, it does not matter which of the other backordered products are satisfied; the *scaled* number of backorders of the other product types under such a priority scheme will be negligible in comparison to the scaled number of backorders of the product type possessing the minimum ratio of b_k/M_k^*. This phenomenon of the normalized inventory (or queue length) processes of high priority customers vanishing in the heavy traffic limit has been observed in previous work; see, for example, Whitt (1971). Although some ambiguity remains in specifying the product type whose backordered demand should be satisfied, the ratio b_k/M_k^* offers a natural ranking of the products when they are backordered. In particular, if any products are backordered, we should first attempt to satisfy the backordered demand for the product possessing the *largest* ratio of b_k/M_k^*. This product type is relatively expensive to backorder and consumes a relatively small amount of effective machine time per unit produced.

Thus far, we have only identified the product whose backordered demand we would like to satisfy, whereas the scheduling decision must specify the job class to be served.

Choosing a job class to serve in order to satisfy backorders for a particular product, say product j, is a straightforward task for a system that does not have both job reentry (that is, processes with multiple stages) and by-production. In a system with no by-production (that is, the by-production matrix is diagonal and only defective products can be by-produced), priority should be awarded to the latest stage of process j that has a positive WIP inventory level. If no WIP exists for process j, then a new job should be started according to process j. In a system where each process contains only one stage, then the number of job classes equals the number of products, and no WIP inventory is held. In this case, (5.5) implies that each product is most rapidly produced by its designated process, and so a new job of process j should be started.

Although a system with by-production and job reentry does not seem to possess an unambiguous sequencing policy for satisfying the backorders of a particular product, intuitively we would like to work on the job class that turns out this type of product in the largest quantity and in the shortest expected time. If the goal is to satisfy the backorders for product j, our proposed policy assigns the index $L_j p_{jk}/(M_j - M_{ji})$ to class (j,i), where the denominator equals the expected remaining processing time for class (j,i) jobs, and, among the classes that have $i = 1$ or have positive WIP inventory, serves the job class with the largest index.

If the weighted inventory $W(t)$ is positive in the Brownian model, then the solution (4.28)-(4.29) dictates that no items are backordered, and positive inventory is held in only one location, either as finished goods inventory of one product type or as WIP inventory of one job class. More specifically, we assign the index h_k/M_k^* to the finished goods inventory of product type k, and the index c_{ji}/M_{ji} to the WIP inventory of class (j,i). Although the WIP and finished goods inventories are measured in different units, these indices measure the holding cost per unit of expected machine time already invested. The inventory should be held in the location that has the minimum index; that is, in the place that achieves the smallest holding cost per unit of expected machine time already invested.

Thus, when no items are backordered and $W(t)$ is less than $\sqrt{n}c^*$, the scheduler should build up inventory in the minimum index location. If this location is class (j,i) of WIP inventory, then priority should be given to any existing WIP inventory of process j, beginning with stage $i-1$ and ending with stage 1. If no WIP inventory currently exists for these job classes, then a new job should be started according to process j. If the minimum index location is product type k of finished goods inventory, then the scheduler should clear the existing WIP inventory of process k, awarding higher priority to later stages. If no process k WIP exists, then the scheduler should begin a new job according to process k.

Notice that the scheduling policy described above insures that the WIP inventory of each job class (except possibly for the class that achieves the minimum value of c_{ji}/M_{ji}) in the actual production/inventory system is either zero or one at each point in time. When $W(t)$ is positive and no items are backordered, it is not obvious whether the scheduler should clear the existing WIP inventory of all classes *before* building inventory in the minimum index location, or, as suggested, to immediately build up inventory in the minimum index location, and hold the existing WIP inventory until it is required to satisfy backordered products. Both options were tested in the simulation experiments of the next section (although the numerical results are not reported here), and we found that the policy that holds the existing WIP inventory (as originally suggested) slightly outperformed (no more than 2% reduction in average cost) the policy that clears the existing WIP inventory. Since the amount of WIP inventory that can be held is quite limited, the small relative cost difference between the two options is not surprising.

The interpretation of the solution to the workload formulation has thus far considered a product to be backordered only when its inventory level is negative, and as a result, the policy does not *anticipate* product backorders. As noted earlier, this lack of anticipation is due to the rescaling that occurs in the Brownian approximation in both time and the inventory levels. If $\rho = 0.9$ and $n = 100$, the solution $Z_k^*(t) = 0$ in (4.29) only implies that the number of tens of product k units in finished goods inventory is zero. Although a small, positive number of unscaled product k units in inventory is consistent with this solution, the Brownian model is unable to distinguish at this finer level of detail. The policy may be more effective if it anticipates potential backorders in order to prevent actual backorders. To this end, we say that product k is *in danger of being backordered* at time t if the unscaled inventory $Z_k(t) < \epsilon_k$, and we alter the proposed policy by replacing the event *product k is backordered* with the event *product k is in danger of being backordered.* The value of the parameter ϵ_k is at the scheduler's discretion, and can be obtained via simulation (as we do in our simulation examples) or by experience. Roughly speaking, ϵ_k should be the minimum inventory level that prevents too many backordered products. In the simulation experiments performed in the next section and in Wein (1990), we found that the most effective choice of ϵ_k was somewhere between zero and $2L_k$.

To summarize, our proposed scheduling policy is a *parametric* policy, where the parameters ϵ_k, $k = 1, \ldots, K$, are chosen by the scheduler to define when a product is in danger of being backordered. The workstation is idle only when the weighted inventory $w(t) \geq \sqrt{n}c^*$ and no products are in danger of being backordered; otherwise, the workstation is busy. When the workstation is working, an indexing scheme is used to prioritize job classes. If any products are currently in danger of being backordered, the index of job class (j, i), $j = 1, \ldots, K$, $i = 1, \ldots, l(j)$, is equal to $L_j p_{jk}/(M_j - M_{ji})$, where $k = \max\{b_j/M_j^* : \text{type } j \text{ product is currently in danger of being backordered}\}$, and the larger the index, the higher the priority. The scheduler then serves the highest priority job class that has $i = 1$ (since raw materials are always available) or positive WIP inventory. When the workstation is busy and no products are in danger of being backordered, the index c_{ji}/M_{ji} is calculated for each job class (j, i), and the index h_k/M_k^* is calculated for each product type k. If job class (j, i) achieves the smallest value of all these indices, then the scheduler awards priority in the order $(j, i-1), (j, i-2), ..., (j, 2)$ to the first class that has WIP inventory, and serves a class $(j, 1)$ job if none of these classes have WIP inventory. Similarly, if the minimum index is achieved by product k, then the scheduler serves the latest stage of process k for which WIP is available, and serves class $(k, 1)$ if process k has no current WIP inventory.

Finally, we note that the policy described above reduces to the proposed policy in Wein (1990) when there is no by-production or job reentry, yield is perfect, and lot sizes are one. We also refer to that paper for a discussion of the similarities and differences of the proposed policy to the classic $c\mu$ rule used to minimize waiting costs in single server queues, to the heavy traffic limit of traditional single server queues, and to results from traditional multi-product inventory theory.

6. Examples

In this section, a simulation experiment is performed with two numerical examples, each composed of three product types and three production processes. In the first example, we consider a production/inventory system with by-production and random yield, but with no job reentry. The second example has perfect yield and no by-products, but processes can have multiple stages of service. Two different cases, corresponding to different holding and backordering costs, are tested for each example.

Since the problem considered in this paper has not been previously addressed in the literature, there are no obvious benchmark policies to test for purposes of comparison. However, in addition to the proposed policy, which is denoted by BROWNIAN in Tables I and II below, we tested two state-dependent heuristic policies that are representative of procedures that might reasonably be used in practice. Both heuristic policies keep the workstation busy when the total (unweighted) finished goods inventory is below a critical level or when products are backordered. The most effective value for this critical level is obtained by employing a one-dimensional search with simulation. The first heuristic policy, denoted by MINIMUM, identifies the product type with the lowest current finished goods inventory level, and gives priority to the latest stage of the corresponding process type that possesses a positive level of WIP. If there is no WIP of this process type, then the scheduler initiates production of a new job according to this process type. The second heuristic policy is denoted by $c\mu$/RUN-OUT because it employs a variation of the

$c\mu$ rule to satisfy backordered products, and uses the *shortest run-out time* philosophy when no products are backordered. In particular, among the subset of products that are currently backordered, the policy considers the type with the maximum index of $b_k/(M_k/p_{kk})$, and then gives priority to the latest stage of the corresponding process type that has a positive WIP inventory, or starts a new job according to this process if there is no existing WIP of this process. If there are no backordered products, then the policy determines the product type that has the minimum value of $Z_k(t)/\lambda_k$, where λ_k is the average demand rate for product k; this quantity represents the expected run-out time of the present finished goods inventory of product k if production for this product is halted. The scheduler serves the latest stage of the corresponding process that has WIP inventory, and starts a new job of this process if the process has no current WIP inventory.

For each policy tested in each cost structure of example 1 (respectively, example 2), 200 (respectively, 100) independent runs were carried out. Each run consisted of 11,000 time units, of which the first 1,000 time units were discarded to reduce the initialization bias. The average cost of the remaining 10,000 time units was observed for each run, and the mean and 95% confidence interval of the average costs for these runs are displayed in Tables I and II for the two respective examples.

6.1 A By-Production System With Random Yield and No Reentry

The first system is depicted in Figure 1, where product 1 is the lowest quality product and product 3 is the highest quality product. The by-production probabilities p_{kj} are shown in the figure, and each job contains a total of L_k =10 units that can potentially be placed in the finished goods inventory. The service times for the three processes are exponential with means $M = (1, 2, 3)$. By (5.1)-(5.2), the expected effective resource consumption vector is $M^* = (0.3, 0.2, 0.35)$. The customer demand for each product is an independent Poisson process, and the demand rates are $\lambda = (1.85, .85, .50)$. By (3.1), over the long run the workstation must complete jobs at the rates $\beta = (.30, .15, .10)$ according to the three processes, and thus $\rho_1 = \rho_2 = \rho_3 = 0.3$ and the traffic intensity

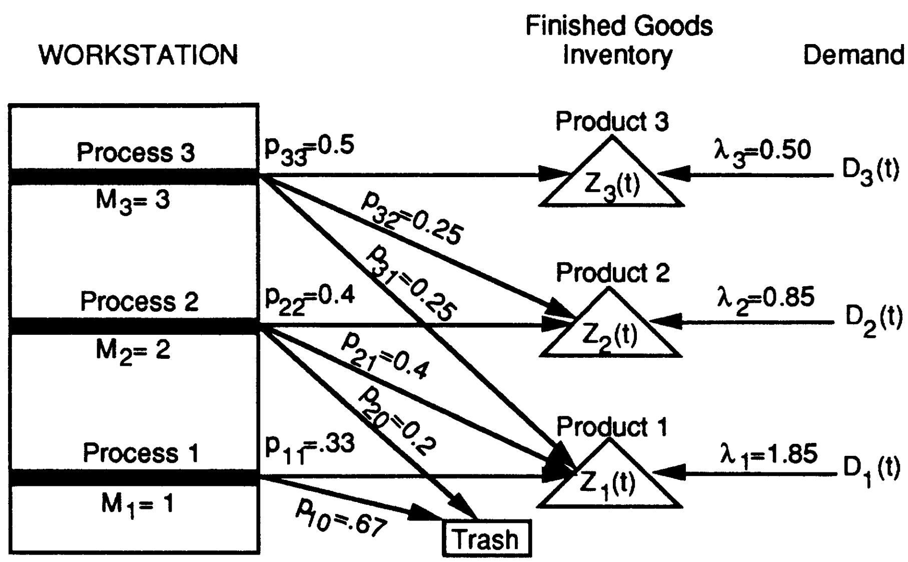

Figure 1. A By-Production System With Random Yield and No Reentry.

$\rho = 0.9$. Since this system has no job reentry, WIP does not need to be carried and the objective is to minimize the cost of backordering and holding finished goods inventory. We consider two cost structures: the backorder and finished goods holding cost are given by $b = (2, 2, 2)$ and $h = (1, 1, 1)$ for case 1, and $b = (6, 5, 11)$ and $h = (3, 2, 3.3)$ for case 2. Thus, product 2 (respectively, class 3) achieves the maximum value of the index b_k/M_k^* for case 1 (respectively, case 2), and product 3 achieves the minimum value of the index h_k/M_k^* in both cases.

Recall that the MINIMUM and $c\mu$/RUN-OUT scheduling policies keep the server busy whenever the total finished goods inventory $\sum_{k=1}^{K} Z_k(t)$ is less than some value, which we denote by c. The value of c, which can be found in Table I, was determined by making two independent runs of 11,000 time units (discarding the first 1000 time units) at various values and searching for the integer value of c that resulted in the lowest average cost. Under the BROWNIAN policy, the server is kept idle whenever the weighted inventory process $w(t) \geq \sqrt{n}c^*$ and $Z_k(t) \geq \epsilon_k$ for $k = 1, 2, 3$. The quantity $\sqrt{n}c^*$ in (5.8) equals 11.8 and 12.2 for the two cases. The search procedure described above was also used to determine the values of $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3)$, and the resulting values were $\epsilon = (12, 5, 0)$ for case 1 and $\epsilon = (20, 5, 0)$ for case 2. Thus, the maximum value of ϵ_k, which is in units of finished goods inventory, is equivalent to two units of WIP inventory.

In order to test the accuracy of the derived value $\sqrt{n}c^*$ in the BROWNIAN policy, the same search procedure (except the interval between tested values was 0.5 instead of 1.0) was used to find the best value of c such that $w(t) > c$. The resulting values of c were 13 for case 1 and 13.5 for case 2, which are slightly larger than the corresponding values of $\sqrt{n}c^*$. We performed 200 simulation runs of the BROWNIAN policy with these critical values of c substituted for $\sqrt{n}c^*$ (the values of ϵ used with $\sqrt{n}c^*$ were also the most cost effective here), and the resulting costs were 42.5(±1.06) and 132(±4.29) for the two cases, which are larger than the corresponding costs under the derived values. Hence the derived values are quite accurate at determining the most cost effective cut-off point for the busy/idle decision.

Table I. Simulation Results for Example 1.

CASE	POLICY	COST
1	BROWNIAN	41.8(±1.16)
1	MINIMUM ($c = 45$)	45.1(±1.37)
1	$c\mu$/RUN-OUT ($c = 55$)	47.6(±0.94)
2	BROWNIAN	129(±3.27)
2	MINIMUM ($c = 45$)	142(±5.14)
2	$c\mu$/RUN-OUT ($c = 60$)	142(±4.08)

Referring to Table I, we see that the BROWNIAN policy outperforms the MINIMUM and $c\mu$/RUN-OUT policies in both cases, with cost reductions ranging between 7.3% and 12.2%. Although the 95% confidence intervals of these costs are too large to confidently quote percentage cost reductions, the confidence intervals of the cost *differences* between policies are small enough to make these percentage cost reductions meaningful.

The expected cost under the BROWNIAN policy predicted by this analysis is $\sqrt{n}c^*h$, which equals 33.7 in case 1 and 115.0 in case 2. Thus, the Brownian model significantly underestimates (by 19.4% and 10.9%, respectively) the true cost. It appears that the heavy traffic analysis is quite accurate at predicting performance at the aggregate level characterized by the weighted inventory process, but is not as accurate at the more refined level of predicting the individual inventory levels Q_{ji} and Z_k that are consistent with this weighted inventory process via (4.18). In particular, the idealized solution in (4.27)-(4.28), where backordering and holding costs are only incurred by one product type, cannot be realized by the actual production/inventory system.

6.2 A Reentry System With Perfect Yield

The second system is illustrated in Figure 2, where process 2 jobs reenter the workstation once and process 3 jobs reenter the workstation twice. The service times for

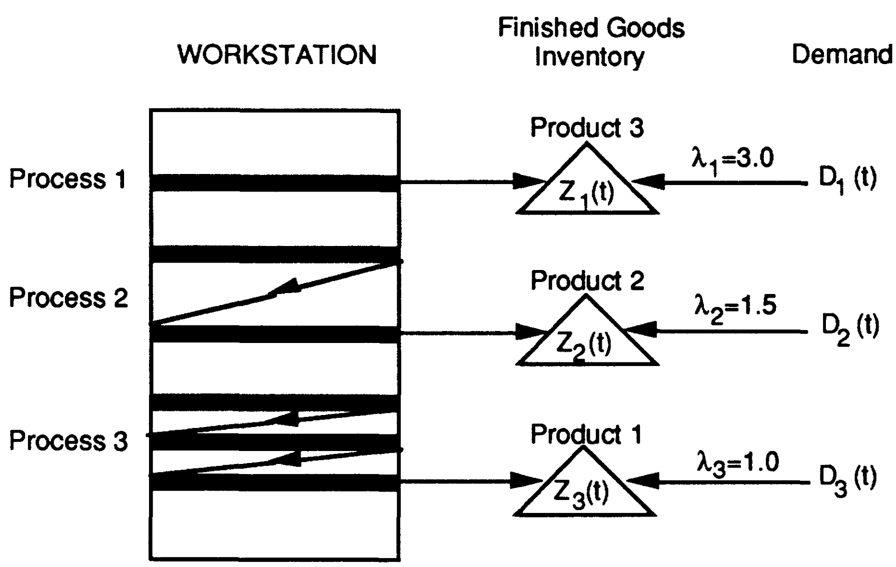

Figure 2. A Reentry System With Perfect Yield.

each of the six job classes are exponentially distributed with mean one, and each job again contains ten potential units for the finished goods inventory. Therefore, the expected effective resource consumption vector is $M^* = (0.1, 0.2, 0.3)$. The customer demand processes are Poisson with rates $\lambda = (3.0, 1.5, 1.0)$, and hence $\rho_{11} = 1/3$, $\rho_{21} = \rho_{22} = 1/6$, $\rho_{31} = \rho_{32} = \rho_{33} = 1/9$, and the traffic intensity $\rho = 0.9$. We again considered two cost structures; for both cases, the WIP holding cost is ten for all job classes, and the backorder and finished goods holding costs are $b = (2, 2, 2)$ and $h = (1, 1, 1)$ for case 1, and $b = (3, 8, 6)$ and $h = (2, 1, 4)$ for case 2. Since each job contains ten units of potential products, in both cases the WIP holding cost for every job class is no greater than the finished goods inventory holding cost for the corresponding product type. This cost structure forces the BROWNIAN policy to build up the finished goods inventory of a particular product type (rather than the WIP inventory of a particular job class) when the workstation is working and no products are in danger of being backordered.

Table II. Simulation Results for Example 2.

CASE	POLICY	COST
1	BROWNIAN	31.2(±0.89)
1	$c\mu$/RUN-OUT ($c = 33$)	34.2(±0.81)
1	MINIMUM ($c = 30$)	35.1(±1.21)
2	BROWNIAN	71.6(±1.84)
2	$c\mu$/RUN-OUT ($c = 35$)	88.5(±2.54)
2	MINIMUM ($c = 33$)	89.8(±3.43)

The parameters for the BROWNIAN policy are $\epsilon = (6, 6, 0)$ for case 1 and $\epsilon = (3, 0, 3)$ for case 2. The derived values for $\sqrt{n}c^*$ are 7.1 and 10.4 for the two respective cases. The best values of this critical quantity found by the search procedure were 5.5 and 8.0, which had corresponding costs of 31.8(±.90) and 76.5(±2.24). These costs are higher than the corresponding costs in Table II, and thus the derived values are again very reliable.

On the other hand, the predicted expected cost under the BROWNIAN policy is 23.7 and 52.0 for the two cases, which corresponds to underestimates of 25.2% and 27.6%, respectively. Table II shows that the BROWNIAN policy outperforms the other two policies, with cost reductions ranging between 8.8% and 20.3%. Also, neither benchmark policy appears to dominate the other in the simulation experiments.

Acknowledgements

This research is partially supported by a grant from the Leaders for Manufacturing Program at MIT, an IBM/University Manufacturing Systems Research Grant, a grant from Texas Instruments, and National Science Foundation Grant Award No. DDM-9057297.

References

Atherton, R. W., and J. E. Dayhoff, (1985). Introduction to Fab Graph Structures. *ECS Abstracts.*

Billingsley, P., (1968). *Convergence of Probability Measures*, John Wiley and Sons, New York.

Bitran, G .R. and S. Dasu, (1989). Ordering Policies in an Environment of Stochastic Yields and Substitutable Demands, *MIT Sloan School Working Paper* # 3019-89-MS.

Bitran, G. R. and T. Leong, (1989). Deterministic Approximations to co-production Problems with Service Constraints, *MIT Sloan School Working Paper* # 3071-89-MS.

Courcoubetis, C., P. Konstantopoulos, J. Walrand, and R. R. Weber, (1989). Stabilizing an Uncertain Production System, *Queueing System* **5**, 37-54.

Deuermeyer, B. L. and W. Pierskalla, (1978). A By-Production Production System with an Alternative, *Management Science* **24**, 1373-1383.

Glassey, R. C. and M. G. C. Resende, (1988). Closed-Loop Release Control for VLSI Circuit Manufacturing, *IEEE Trans. on Semiconductor Manufacturing* **1**, 36-46.

Harrison, J. M., (1988). Brownian Models of Queueing Networks with Heterogeneous Customer Populations, in Fleming and Lions (eds.), *Stochastic Differential Systems, Stochastic Control Theory and Applications*, IMA Volume **10**, Springer-Verlag, New York, 147-186.

Harrison, J. M., C. A. Holloway, and J. M. Patell, (1990). Measuring Delivery Performance: A Case Study from the Semiconductor Industry, Chapter 11 in *Measures for Manufacturing Excellence*, Robert S. Kaplan, Ed., Harvard Business School Press.

Kelly, F. P., (1979). *Reversibility and Stochastic Networks*, John Wiley and Sons, New York.

Klimov, G. P., (1974). Time Sharing Service Systems I, *Th. Prob. Appl.* **19**, 532-551.

Wein, L. M., (1988). Scheduling Semiconductor Wafer Fabrication, *IEEE Trans. on Semiconductor Manufacturing* **1**, 115-130.

Wein, L. M., (1990). Dynamic Scheduling of a Multiclass Make-To-Stock Queue, *MIT Sloan School Working Paper #* 3113-90-MS.

Whitt, W., (1971). Weak Convergence Theorems for Priority Queues: Preemptive-Resume Discipline. *J. Appl. Prob.* **8**, 74-94.