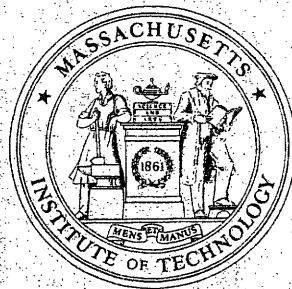


OPERATIONS RESEARCH CENTER

working paper



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

**The Impact of Processing Time
Knowledge on Dynamic Job-Shop
Scheduling**

by

Lawrence M. Wein and Jihong Ou

OR 205-89

November 1989

**THE IMPACT OF PROCESSING TIME KNOWLEDGE
ON DYNAMIC JOB-SHOP SCHEDULING**

Lawrence M. Wein

Sloan School of Management, M.I.T.

and

Jihong Ou

Operations Research Center, M.I.T.

Abstract

The goal of this paper is to determine if the results for dynamic job-shop scheduling problems are affected by the assumptions made with regard to the processing time distributions and the scheduler's knowledge of the processing times. Three dynamic job-shop scheduling problems (including a two station version of Conway et al.'s [2] nine station symmetric shop) are tested under seven different scenarios, one deterministic and six stochastic, using computer simulation. The deterministic scenario, where the processing times are exponential and observed by the scheduler, has been considered in many simulation studies, including Conway et al.'s. The six stochastic scenarios include the case where the processing times are exponential and only the mean is known to the scheduler, and five different cases where the machines are subject to unpredictable failures. Two policies were tested, the shortest expected processing time (SEPT) rule, and a rule derived from a Brownian analysis of the corresponding queueing network scheduling problem. Although the SEPT rule performed well in the deterministic scenario, it was easily outperformed by the Brownian policies in the six stochastic scenarios for all three problems. Thus, the results from simulation studies of dynamic, deterministic job-shop scheduling problems do not necessarily carry over to the more realistic setting where there is unpredictable variability present.

November 1989

THE IMPACT OF PROCESSING TIME KNOWLEDGE ON DYNAMIC JOB-SHOP SCHEDULING

Lawrence M. Wein

Sloan School of Management, M.I.T.

and

Jihong Ou

Operations Research Center, M.I.T.

1. Introduction and Summary

The job-shop scheduling literature can be categorized into *static* problems that address a fixed set of jobs and *dynamic* problems that allow jobs to arrive to the shop in an ongoing, and usually random, fashion. In this paper we are concerned with dynamic job-shop scheduling problems, which can be further categorized according to the assumptions made regarding the processing times. If the processing times are known with certainty by the scheduler before the processing actually occurs, then the dynamic scheduling problem will be referred to as *deterministic*. However, it is sometimes assumed that the scheduler cannot observe the processing times in advance, but only has knowledge of a probability distribution for the various processing times, in which case the dynamic scheduling problem will be referred to as *stochastic*.

The two main questions faced in dynamic scheduling problems are “how to schedule?” and “how much improvement will take place?”; that is, ideally one would like to find an optimal scheduling policy and then assess the increase in performance that this scheduling policy will achieve relative to the commonly used policy, which is usually taken to be the first-come first-serve (FCFS) policy. The goal of this paper is to determine whether the answers to these two questions for a dynamic job-shop scheduling problem depend upon the assumptions made with regard to the processing time distributions and the scheduler’s

knowledge of the processing times.

Dynamic scheduling problems, both deterministic and stochastic, can be defined as scheduling problems for queueing networks, where each machine in the shop is a server in the network, and each job in the shop corresponds to a customer in the queueing system. These scheduling problems tend to be very difficult to analyze, except for the case where there is only a single machine in the shop, and the objective is to minimize the expected average cycle time of jobs, where a job's cycle time is the amount of time it spends in the shop. In this case, the answer to the question "how to schedule?" does not depend very much upon the processing time distributions or the scheduler's knowledge of the processing times. Ignoring the specific assumptions made regarding processing time distributions, interarrival times, the discount rate, and preemption, the optimal policy is the shortest expected remaining processing time rule (or a minor perturbation of this rule), which gives priority to the job with the shortest expected remaining processing time, where, in the deterministic case, a job's expected remaining processing time is equal to the job's actual remaining processing time. If no feedback of jobs and no preemption is allowed, then this policy reduces to the shortest expected processing time (SEPT) rule. In this paper, the SEPT policy will always refer to the policy that awards priority at each machine to the job whose upcoming operation has the shortest expected processing time, regardless of the processing time distributions and the scheduler's knowledge of the processing times.

In the network setting, the primary tool of analysis has traditionally been computer simulation, and the classic study on this topic is contained in Chapter 11 of Conway et al. [2]. Here, the authors study a nine station symmetric job-shop with Poisson arrivals, and the processing times at each station are exponential with mean one. The scheduler has perfect knowledge of a job's route and each of its processing times at the moment of the job's arrival to the shop, and so this is a *deterministic* dynamic scheduling problem. The main conclusion of this often quoted study is that the SEPT policy is very effective at

minimizing average job cycle time (outperforming FCFS by 53.2%), even when compared to more sophisticated policies that use dynamic, global information. As a result of this and related studies, the SEPT rule is still widely regarded as the most effective basic scheduling policy for reducing cycle time in job-shops (see, for example, Baker [1]). Furthermore, many simulation studies carried out through the years have considered dynamic, deterministic (as opposed to stochastic) scheduling problems.

Let us now turn to the modeling issues of processing time distributions and the scheduler's knowledge of processing times. Define a job's *effective* processing time (see Section 7 of Harrison [4]) for a particular operation to be the elapsed time between the start of the operation's processing and the completion of its processing. This elapsed time consists of the actual processing time (which may include time for set-up, inspection, and rework), and any interruptions (such as tool or operator unavailability, machine failure, and preventive maintenance) that occur before the job completes its processing. (We will assume that preemption by higher priority jobs is not possible in this paper.)

Thus, an operation's effective processing time may incorporate both *predictable* and *unpredictable* variability, and this variability may stem from either the process (for example, the machine or operator) or the job. The primary source of predictable variability typically stems from the job; some operations for some jobs may inherently take longer than others, and the scheduler may have some (or even perfect) knowledge of a job's actual processing time. Other sources of predictable variability originate from the process, and include preventive maintenance, coffee breaks, and end-of-shift effects. The primary source of unpredictable variability is usually process interruptions, such as machine failures, tool breakages, or operator unavailability. A second source of unpredictable variability is the length of an operation's processing time, which may not be known precisely. A third source of unpredictable variability is rework, which may be caused by the precision of the process or the difficulty of the particular operation.

It seems clear that all job-shops experience some amount of unpredictable variability,

and in most job-shops, the majority of variability is unpredictable. Therefore, the dynamic, deterministic scheduling problem often addressed in the literature is less realistic than the dynamic, stochastic scheduling problem.

In this paper, a simulation study involving three dynamic job-shop scheduling problems (which are actually queueing network scheduling problems) is carried out. Effective scheduling policies for two of the three problems have been proposed in Harrison and Wein [5] -[6] by analyzing a Brownian approximation to the queueing network scheduling problems. Both problems (see Figures 1 and 2) consist of two machine, two product job-shops, and one is *open* (it has exogenous arrivals) and one is *closed* (the total number of jobs in the shop is held constant; see Solberg [10] for more on the modeling of job-shops as closed queueing networks). The third problem is a two station version of Conway et al.'s [2] nine station, symmetric job-shop, and a Brownian analysis of this problem is performed here in order to develop an effective scheduling policy. These three shops, which are described and analyzed in sections 3 through 5, will be referred to as the open shop, the closed shop, and the symmetric shop, respectively.

For each of the three problems, seven different scenarios are considered, where each scenario makes different assumptions with regard to the processing time distributions and the scheduler's knowledge of the processing times. These seven scenarios will be described in detail in Section 2. Two of the seven scenarios assume that the processing times for a particular operation for a particular type of product is an independent and identically distributed (iid) exponential random variable. In one of these two scenarios, the scheduler knows only the mean of the the exponential random variables, which corresponds to the simulation studies in Harrison and Wein [5]-[6]. In the other scenario, the scheduler observes the actual processing time, which corresponds to the Conway et al. [2] simulation study. These two scenarios will be referred to as the EXPONENTIAL case and the DETERMINISTIC case, respectively.

The other five scenarios all assume that the processing times for a particular operation

for a particular type of product is a constant known by the scheduler. However, it is assumed that the server fails after an exponentially distributed amount of busy time, after which the server requires an exponentially distributed amount of repair time. The interruption is of a preemptive resume nature, so that when the server is repaired, it resumes where it left off. The five scenarios vary with respect to the mean length of the time between failures, the mean length of the repair time, and the relative amount of unpredictability in the effective processing times. Notice that the effective processing times in these five cases are stochastic, and so six of the seven scenarios are dynamic, stochastic scheduling problems.

For all seven scenarios of the three problems, the FCFS policy, the SEPT policy, and the scheduling policy proposed by the Brownian analysis, which will be referred to as the BROWNIAN policy, were tested in a simulation experiment. The Brownian analysis can be applied to all six stochastic scenarios, and, in fact, proposes the same policy in all six cases. For the DETERMINISTIC scenario, straightforward extensions of these policies were tested. The objective in all three problems was to minimize the mean cycle time subject to a given throughput rate. It should be pointed out that in the open problem and the symmetric problem, the exponential processing times were the same for all operations performed by a particular server (which is not uncommon in some manufacturing settings, such as semiconductor wafer fabrication), so that FCFS and SEPT were identical for the six stochastic scenarios.

The results of the simulation study were very conclusive. For the DETERMINISTIC scenario, the SEPT rule was very effective and easily outperformed FCFS. The difference in performance between SEPT and the BROWNIAN policy was small for two of the three problems, although SEPT was superior in all three problems. However, in the other six scenarios, the BROWNIAN rule easily outperformed SPT and FCFS. In the closed problem, where SEPT and FCFS resulted in different policies, neither policy consistently dominated the other. Furthermore, the percentage improvement from scheduling relative to FCFS was

often twice as large in the DETERMINISTIC scenario as in the other six cases. Also, the percentage improvement in performance of the Brownian policy (relative to FCFS) was quite stable over the six stochastic scenarios, after adjusting for the customer population in the closed problem. Thus, the answers to our two questions “how to schedule?” and “how much improvement will take place?” depend greatly on the scheduler’s knowledge of the processing times. However, if unpredictable variability is incorporated, the answers are quite robust with respect to the specific processing time distributions.

The implications of this experiment are clear. Results from simulation studies of dynamic, deterministic scheduling problems do not necessarily carry over to the more realistic setting where there is unpredictable variability present. In particular, policies performing well in the deterministic setting may not perform particularly well in a stochastic setting, and the reported improvements caused by scheduling in deterministic studies probably greatly overestimate the improvements that would result in the more realistic stochastic setting. On the other hand, it appears that stochastic scheduling problems offer essentially the same results, regardless of the various distributional assumptions made on the processing times. This study also raises concern about the applicability of the deterministic scheduling literature and the expert systems scheduling literature, neither of which explicitly deal with unpredictable variability, for scheduling actual job-shops.

Thus, we urge researchers who use computer simulation models to incorporate unpredictable variability in their models. This variability will allow the model to more accurately reflect the operations of a job-shop, and will prevent researchers from developing conclusions that do not apply to the actual scheduling problem.

2. The Different Scenarios

As mentioned earlier, the underlying job-shop in each of the three problems can be represented as a multiclass queueing network. Each shop has two machines that produces a variety of products and each product has its own deterministic route, or sequence of

operations. In keeping with traditional queueing network terminology (see Kelly [7], for example), we will define a different *customer class* for each operation of each product. For example, in Figure 1, Product *A* has one stage on its route and product *B* has two stages, so that this problem has three customer classes in total. Each customer class is served by a particular machine and has its own effective processing time distribution. For $k = 1, \dots, K$, the mean effective processing time for class k customers will be denoted by \bar{m}_k and the mean processing time will be denoted by m_k .

For a given class k , the mean effective processing time \bar{m}_k , which includes interruptions during service, will be the same for all seven scenarios of a given problem. Therefore, the average load on the shop will not differ by scenario. For the DETERMINISTIC and EXPONENTIAL cases, the effective processing time distributions are exponential, and $\bar{m}_k = m_k$. The only difference between the two cases is that the scheduler observes the actual processing time in the DETERMINISTIC case, but has knowledge of only \bar{m}_k in the EXPONENTIAL case.

In the other five cases, the processing times m_k will be a known constant, but there will be machine failures and repairs. For simplicity, we assume that, for a given scenario of a given problem, the two machines have identical failure and repair rates. Let the busy time between failures for each machine be iid exponential random variables with mean r^{-1} , and let the repair times be iid exponential random variables with mean c . Then

$$\bar{m}_k = m_k(1 + rc). \quad (1)$$

Let α be defined by

$$\alpha = \frac{1}{1 + rc}, \quad (2)$$

so that

$$\alpha = \frac{m_k}{\bar{m}_k}. \quad (3)$$

Thus α measures the fraction of the mean effective processing time that is known with certainty. If the machines never fail, then $\alpha = 1$, and the scheduler has perfect knowledge of

the effective processing times. As the failure rate and expected repair time go to infinity, α tends to zero, and the scheduler's knowledge of the actual processing times is overwhelmed by the failures and repairs.

In the remaining five scenarios, α will take on the values of .5, .7, and .9, which denote low reliability, medium reliability, and high reliability machines, respectively. Since \bar{m}_k is held constant for all the scenarios, it follows from (3) that we must allow the mean processing time m_k to vary by scenario. The mean repair time c will also vary by scenario. For a given problem, let $\bar{m} = K^{-1} \sum_{k=1}^K \bar{m}_k$ denote the average of the mean effective processing times over the various classes (not weighted by their relative usages). The parameter c will take on the values of $.5m$, $3m$, and $18m$, which represent machines with short failures, medium failures, and long failures, respectively. Given α and c for a particular scenario, the failure rate r is chosen to satisfy definition (2).

The five scenarios will be $(\alpha = .5, c = 3m)$, $(\alpha = .7, c = 3m)$, $(\alpha = .9, c = 3m)$, $(\alpha = .7, c = .5m)$, and $(\alpha = .7, c = 18m)$. For the remainder of this paper, these cases will be abbreviated by UNRELIABLE, MEDIUM, RELIABLE, SHORT FAILURES, and LONG FAILURES, respectively.

3. The Open Shop

This simple problem is pictured in Figure 1, where product A jobs visit station 1 and then exit, and product B jobs visit station 1, proceed to station 2, and then exit. Customers of each product arrive to station 1 according to independent Poisson processes with rates $\lambda_A = 1$ and $\lambda_B = .9$. The mean effective processing times $\bar{m}_k, k = 1, 2, 3$, are given in Figure 1. In the six stochastic scenarios, the only real scheduling decision is to decide whether to serve product A jobs or product B jobs at station 1. For these scenarios, we will assume that FCFS is used at station 2 for both scheduling policies displayed in Table I.

For the six stochastic scenarios, the queueing network scheduling problem was ana-

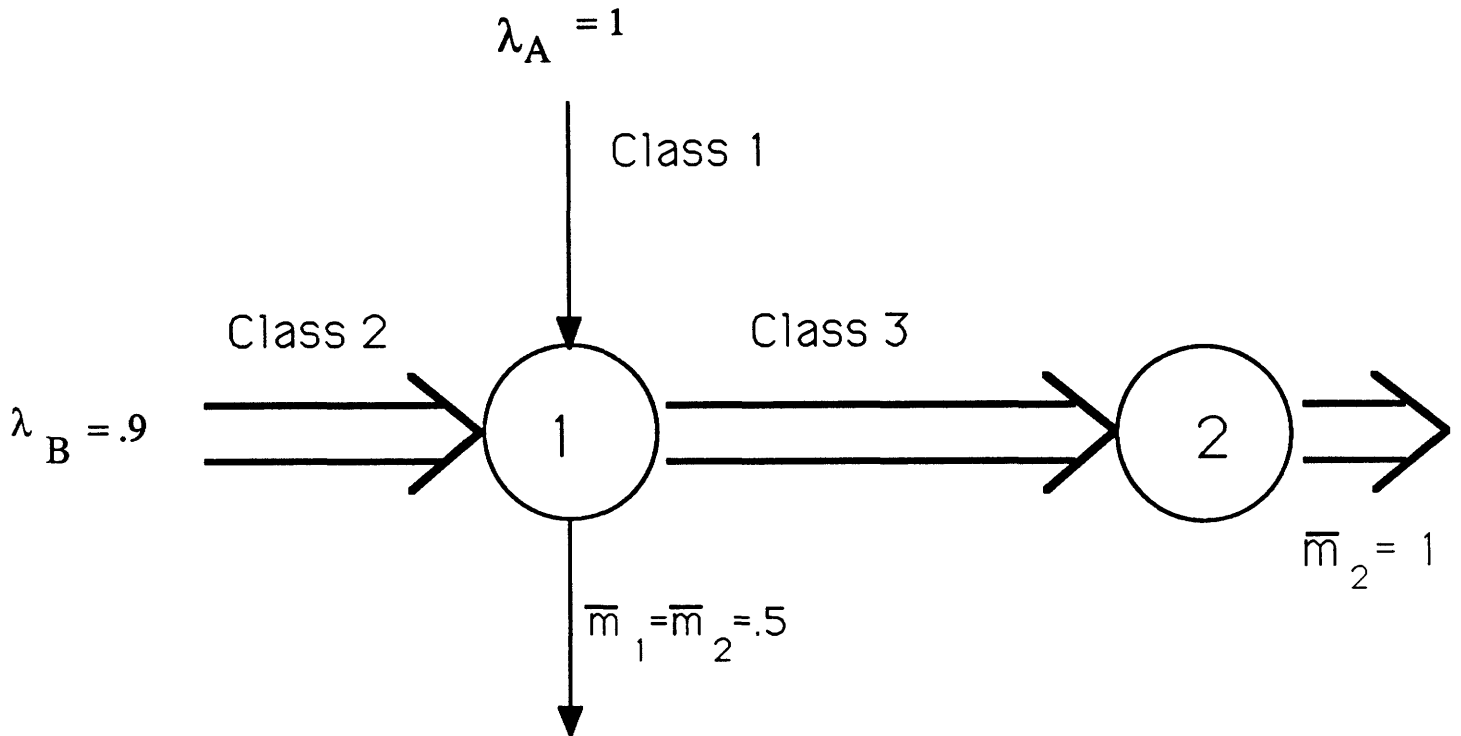


Figure 1. The open shop.

lyzed in Harrison and Wein [5]. The proposed scheduling policy awards priority to product *A* jobs unless there are c or fewer jobs in queue and in service at machine 2. In the latter case, priority is given to product *B* jobs in order to avoid idleness at machine 2. The most effective value of the parameter c was chosen via computer simulation. This policy, which is referred to as the BROWNIAN policy in Table I, was used for the six stochastic scenarios.

The scheduling problem, however, is fundamentally different under the DETERMINISTIC scenario. In particular, there would be an infinite number of customer classes (rather than just three) in this case, because a different customer class would be needed for each possible realization of the actual effective processing times. Rather than attempt to solve the resulting Brownian control problem in this case, a straightforward extension of the stochastic BROWNIAN policy was used. For the DETERMINISTIC case, the BROWN-

IAN policy used SEPT at station 2, ranked product *A* jobs and product *B* jobs according to SEPT at station 1, and then served product *A* jobs, unless the amount of remaining work in service and in queue at machine 2 was less than a parameter *c*, in which case type *B* jobs would be given priority.

The mean cycle times and 95% confidence intervals for the SEPT policy and the BROWNIAN policy are recorded in Table I for all seven scenarios. Notice that the mean cycle time for the FCFS policy can be inferred from this table. For the six stochastic scenarios, FCFS and SEPT are identical. Furthermore, FCFS is the same under the DETERMINISTIC and EXPONENTIAL cases, since the processing times are not used to make the scheduling decisions.

SCENARIO	SEPT	BROWNIAN
DETERMINISTIC	4.90 (± 1.19)	5.67 (± 0.28)
EXPONENTIAL	13.8 (± 0.75)	12.2 (± 0.71)
UNRELIABLE	30.1 (± 1.95)	27.3 (± 1.84)
MEDIUM	21.6 (± 1.42)	19.5 (± 1.48)
RELIABLE	12.0 (± 0.75)	10.8 (± 0.71)
SHORT FAILURES	9.43 (± 0.54)	8.47 (± 0.53)
LONG FAILURES	70.1 (± 5.45)	62.1 (± 5.00)

TABLE I. Mean cycle times for the open shop.

For the six stochastic scenarios, the BROWNIAN policy outperformed the SEPT policy. Since the confidence intervals of the two policies overlap in these six cases, a Bonferroni *t*-test was performed (see Miller [9], for example) to determine if the difference between the two policies was statistically significant. Results showed that the BROWNIAN policy was significantly better than the SEPT policy at a 95% level for all six stochastic scenarios. As in Harrison and Wein [5], the percentage improvements are quite modest,

on the order of 10%. This seems to be inherent in the scheduling problem, since a lower bound for any scheduling policy on the total number of jobs in the system was derived (assuming exponential and unobserved processing times) in Harrison and Wein [5], and the difference in performance between this bound and the BROWNIAN policy was quite small. The improvement of the BROWNIAN policy over SEPT appears to be relatively stable over the six stochastic scenarios.

For the DETERMINISTIC case, the SEPT policy and the BROWNIAN policy reduced the mean cycle time by more than 100% relative to the FCFS policy, and the SEPT policy outperformed the BROWNIAN policy. Thus, there is a significant difference between the DETERMINISTIC scenario and the six stochastic scenarios, in terms of the most effective scheduling policy and the improvements gained from scheduling.

4. The Closed Shop

In this section, we consider a scheduling problem for the closed shop pictured in Figure 2. Again there are two products, A and B , and A has two stages on its route and B has four stages. The six customer classes will be indexed by $k = 1, \dots, 6$ and referred to as classes $A1, A2, B1, B2, B3$, and $B4$. The mean effective processing times $\bar{m}_k, k = 1, \dots, 6$ are given in Figure 2. A closed network has a constant number of jobs, N , populating the shop, so that a new job enters whenever a job exits the shop. The entering jobs are deterministically chosen in the order $ABABAB\dots$ so that a 50-50 product mix is maintained.

The scheduling problem is to decide how to dynamically award priority among classes $A1, B1$, and $B3$ at station 1, and classes $A2, B2$, and $B4$ at station 2. Harrison and Wein [6] analyzed this scheduling problem and proposed the following static policy. Let \bar{M}_{ik} be the expected remaining effective processing time at station i for a class k customer before that customer exits the shop, and rank each class by the index $\bar{M}_{1k} - \bar{M}_{2k}$, which equals

$$\bar{M}_{1k} - \bar{M}_{2k} = (3 \quad -1 \quad -3 \quad -11 \quad -5 \quad -7) \quad \text{for } k = 1, \dots, 6, \quad (4)$$

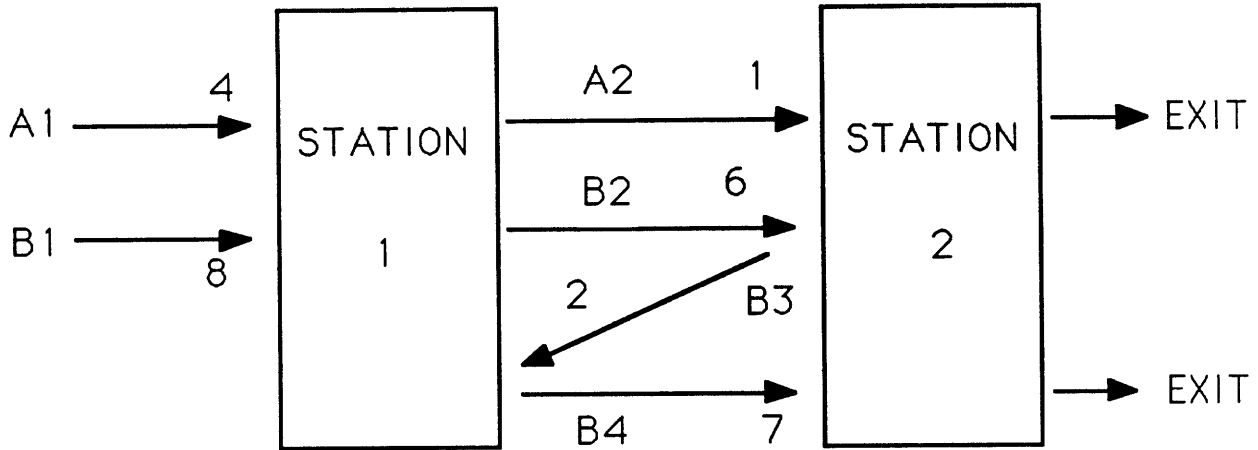


Figure 2. The closed shop.

in this example. At station 1, the policy gives priority to the smaller values of this index, and at station 2 gives priority to the larger values of this index. Thus, priorities (from highest to lowest) are in the order (B3, B1, A1) at station 1 and (A2, B4, B2) at station 2. This policy is referred to as the BROWNIAN policy in Table II. For the DETERMINISTIC scenario, the index in (4) contains the actual remaining effective processing times for each customer.

Since the mean effective processing times differ among the classes served at the same machine, the SEPT and FCFS policies differ for all seven scenarios. For the six stochastic scenarios, the SEPT policy awards priority in the order (B3, A1, B1) at station 1 and (A2, B2, B4) at station 2.

The population level N for each policy was chosen so that a specified average throughput rate of .127 customers per unit of time was achieved, which corresponds to a machine utilization of 88.9%. The mean cycle time of customers was then compared at this constant throughput rate. There were two reasons to use this comparison, rather than to hold the population level constant for all policies. The first reason is to maintain consistency with the two open shops, where the mean throughput rate is exogenously specified by the arrival rates, and the mean cycle times are then compared. Secondly, manufacturing sys-

tems using closed loop input (i.e., keeping the population level constant) will attempt to produce at the rate at which the products are demanded, and will choose the population level N accordingly. Thus, the percentage improvements reported in this way more accurately reflect what would be achieved in practice. Because of the discrete nature of the parameter N , there were a few policies that could not achieve the target throughput rate precisely. In these cases, we have reported linear interpolations of the various performance measures so that the average throughput rate would be .127.

SCENARIO	FCFS	SEPT	BROWNIAN
DETERMINISTIC	78.7 (± 0.46)	39.2 (± 0.23)	41.9 (± 0.25)
EXPONENTIAL	78.7 (± 0.46)	74.6 (± 0.48)	53.2 (± 0.33)
UNRELIABLE	157 (± 1.06)	172 (± 1.1)	110 (± 0.74)
MEDIUM	86.1 (± 0.56)	94.1 (± 0.63)	66.9 (± 0.40)
RELIABLE	29.3 (± 0.14)	25.0 (± 0.12)	25.4 (± 0.14)
SHORT FAILURES	31.5 (± 0.09)	29.4 (± 0.10)	28.4 (± 0.10)
LONG FAILURES	391 (± 9.23)	465 (± 10.0)	290 (± 5.4)

TABLE II. Mean cycle times for the closed shop.

The simulation results for the three policies are recorded in Table II. For the EXPONENTIAL, UNRELIABLE, MEDIUM, and LONG FAILURES scenarios, the BROWNIAN policy easily outperformed the SEPT policy, reducing the mean cycle time by an average of 32.8%. This is in striking contrast to the DETERMINISTIC, RELIABLE, and SHORT FAILURES scenarios, where the difference between the two policies is small, and SEPT is actually more effective in two of these cases. The FCFS policy was outperformed by the Brownian policy in all seven scenarios, but showed significant improvements over SEPT in three of the seven scenarios.

Although the throughput rates were held constant for all seven scenarios, the population level required to achieve this rate varied considerably. In particular, the population level was very low (in the three to five range) for the three cases where the BROWNIAN policy did not easily outperform the SEPT policy. The low population levels can be partially explained by the variability in effective processing times for our various scenarios. Let c_k^2 be the squared coefficient of variation (variance divided by the square of the mean) of the effective processing time for class k customers. Then for the five scenarios that have machine interruptions,

$$c_k^2 = \frac{2\alpha r c^2}{\bar{m}_k}, \quad (5)$$

and the three cases mentioned earlier had the lowest squared coefficients of variation.

SCENARIO	THROUGHPUT	SEPT	BROWNIAN
DETERMINISTIC	.140	70.6 ($\pm .45$)	70.9 ($\pm .46$)
RELIABLE	.138	101 ($\pm .41$)	72.2 ($\pm .33$)
SHORT FAILURES	.141	156 ($\pm .90$)	71.0 ($\pm .17$)

TABLE III. Mean cycle times for the closed shop at different throughput rates.

The BROWNIAN policy was derived by Harrison and Wein [6] under the condition that the population level was relatively large. It is not clear from Table II whether the relative ineffectiveness of the BROWNIAN policy in the RELIABLE and SHORT FAILURES scenarios is due to low squared coefficients of variation or to a low population level. Thus, these two cases and the DETERMINISTIC case were tested again, this time with the desired throughput rate set at a higher level. The results are shown in Table III, and it can be seen that for the two stochastic cases the BROWNIAN rule easily outperformed SEPT, while the two policies were virtually indistinguishable under the DETERMINISTIC case.

In summary, for the six stochastic scenarios, the BROWNIAN policy easily outperformed the SEPT and FCFS policies at moderate and high population levels, and the SEPT policy outperformed FCFS at lower population levels, but performed worse than FCFS at higher population levels. For the DETERMINISTIC case, SEPT slightly outperformed the BROWNIAN policy at lower population levels, and the two perform similarly at higher population levels.

5. The Symmetric Shop

In this section, we consider a two station version of the nine station symmetric job shop studied in Chapter 11 of Conway et al. [2]. Customers arrive according to independent Poisson processes at rate λ to each station. When a job completes service at a station, it visits the other station with probability one-half and exits the shop with probability one-half, independent of all previous history. For ease in developing the simulation results, we did not allow a customer to have more than six stages on its route. As in Conway et al. [2], a job's entire route is chosen at the time of its arrival to the shop, and is made known to the scheduler. The mean effective processing times for each stage of each customer's route is one. The corresponding arrival rate λ to each station was chosen to be $32/35$, so that the resulting traffic intensity at each station was $.9$, as in Conway et al. [2]. As mentioned earlier, the DETERMINISTIC scenario, where all processing times are exponential and known to the scheduler, corresponds to the assumptions made in Conway et al. [2].

Since the shop satisfies the balanced heavy loading conditions set forth in Harrison [4], we will analyze the Brownian approximation to the queueing network scheduling problem for the stochastic scenarios. As in the previous two problems, the proposed policy only depends on the effective processing time distributions through their mean, and thus is identical for all six stochastic scenarios.

As before, let $M_{i;k}$ be the expected remaining effective processing time for a class k customer at station i before that customer exits the shop. Since the mean effective

processing times are all one, M_{ik} also equals the remaining number of visits to station i by a customer of class k . Since no immediate feedback is allowed and customers are not allowed to have more than six stages on their route, then it follows that there are $K = 12$ customer classes and the *workload profile matrix* $M = (M_{ik})$ is given by

$$M = \begin{pmatrix} 1 & 0 & 1 & 1 & 2 & 1 & 2 & 2 & 3 & 2 & 3 & 3 \\ 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 \end{pmatrix}. \quad (6)$$

The two classes that correspond to columns three and four of this matrix each have one service remaining at each station, but one class is currently at station 1 and the other class is at station 2. For now, we will not need to distinguish between these two classes, or other similar pairs of classes.

Let $Q_k(t)$, $k = 1, \dots, 12$, be the number of class k jobs in the network at time t . Also, let $I_i(t)$, $i = 1, 2$, be the cumulative amount of idle time at station i in the interval $[0, t]$. Introducing a system parameter n , let us define the *scaled queue length process* $Z = (Z_k)$ by

$$Z_k(t) = \frac{Q_k(nt)}{\sqrt{n}}, \quad t \geq 0 \text{ and } k = 1, \dots, 12, \quad (7)$$

and the *scaled cumulative idleness process* $U = (U_i)$ by

$$U_i(t) = \frac{I_i(nt)}{\sqrt{n}}, \quad t \geq 0 \text{ and } i = 1, 2. \quad (8)$$

The approximating Brownian control problem is formulated by letting the parameter $n \rightarrow \infty$, and we will retain the notation Z and U for the two limiting scaled processes. Harrison [4] has argued that the queueing network scheduling problem described above is well approximated by a Brownian network control problem whose workload formulation is to choose RCLL (right continuous with left limits) processes Z and U to

$$\text{minimize } \limsup_{T \rightarrow \infty} \frac{1}{T} E \left[\int_0^T \sum_{k=1}^{12} Z_k(t) dt \right] \quad (9)$$

$$\text{subject to } \sum_{k=1}^{12} M_{ik} Z_k(t) = B_i(t) + U_i(t) \text{ for all } t \geq 0, \text{ and } i = 1, 2, \quad (10)$$

$$U \text{ is nondecreasing with } U(0) = 0, \quad (11)$$

$$Z(t) \geq 0 \text{ for all } t \geq 0, \text{ and} \quad (12)$$

$$Z \text{ and } U \text{ are nonanticipating with respect to } X, \quad (13)$$

where X is a twelve-dimensional Brownian motion with drift vector θ and covariance matrix Σ , and $B = (B_1, B_2)$ is a two-dimensional Brownian motion with drift $M\theta$ and covariance matrix $M\Sigma M^T$. It will turn out that the solution (Z, U) will be nonanticipating with respect to B , and the proposed scheduling policy is independent of the various drift and covariance values. Notice that the objective is to minimize the long run expected average number of customers in the system, which will also minimize the long run expected average cycle time of customers by Little's formula [8].

A solution to this problem will be found that minimizes the objective (9) for all times t with probability one. Readers are referred to Yang [11] for necessary and sufficient conditions under which stochastic control problems like (9)-(13) possess such *pathwise solutions*.

For now, suppose we are given a cumulative idleness process (U_1, U_2) that satisfies constraints (11) and (13). The right side of constraint (10), which we denote by

$$b_i(t) = B_i(t) + U_i(t), \text{ for } i = 1, 2 \text{ and } t \geq 0, \quad (14)$$

would be known, and assume for now that b_1 and b_2 are nonnegative processes.

Now consider the following linear program (LP) that is imbedded in (9)-(13) at each time t :

$$\min_{Z(t)} \sum_{k=1}^{12} Z_k(t) \quad (15)$$

$$\text{subject to } \sum_{k=1}^{12} M_{ik} Z_k(t) = b_i(t), \text{ for } i = 1, 2, \quad (16)$$

$$Z_k(t) \geq 0, \text{ for } k = 1, \dots, 12. \quad (17)$$

Define the dual variables $\pi_1(t)$ and $\pi_2(t)$ and consider the dual to this LP, which is

$$\max_{\pi_1(t), \pi_2(t)} b_1(t)\pi_1(t) + b_2(t)\pi_2(t) \quad (18)$$

$$\text{subject to } M_{1k}\pi_1(t) + M_{2k}\pi_2(t) \leq 1, \text{ for } k = 1, \dots, 12. \quad (19)$$

The solution to this dual LP is $\pi^*(t) = (1/3, 0)$ if $b_1(t) > b_2(t)$, and $\pi^*(t) = (0, 1/3)$ if $b_1(t) < b_2(t)$. If $b_1(t) = b_2(t)$, then any nonnegative π^* such that $\pi_1^*(t) + \pi_2^*(t) = 1/3$ is optimal.

Recall that the dual variable $\pi_i^*(t)$ represents the increase in the value of the objective function (15) per unit of increase in the right side value of constraint (16). Since $\pi^*(t)$ is nonnegative for all nonnegative values of $(b_1(t), b_2(t))$, it follows (see Yang [11] for details) that the workload formulation (9)-(13) can be solved by finding the processes U_1^* and U_2^* that minimize the processes b_1 and b_2 , respectively, and then solving the linear program (15)-(17) for $Z^*(t)$ at each point in time t .

It is well known (see Harrison [3]) that the processes defined by

$$U_i^*(t) = - \inf_{0 \leq s \leq t} B_i(s) \quad (20)$$

minimize the value of $U_i(t)$ for all times t with probability one, for $i = 1, 2$, subject to the constraints (10)-(13). Furthermore, the resulting processes $\{B_i(t) + U_i^*(t), t \geq 0\}, i = 1, 2$, are nonnegative.

Define the *dynamic reduced costs* $\bar{c}_k(t)$ by

$$\bar{c}_k(t) = 1 - \pi_1^*(t)M_{1k} - \pi_2^*(t)M_{2k} \text{ for } k = 1, \dots, 12, t \geq 0. \quad (21)$$

The value $\bar{c}_k(t)$ can be interpreted as the increase in the objective function of the LP (15)-(17) per unit of increase in the right side value of the nonnegativity constraint (17). Thus, the higher the value of $\bar{c}_k(t)$, the more expensive it is to hold class k customers in queue at time t . A natural sequencing policy to consider is to always serve the customer

class that has the largest dynamic reduced cost. Before we describe this policy, define the *workload process*

$$W_i(t) = \sum_{k=1}^{12} M_{ik} Q_k(t), \quad i = 1, 2, \quad \text{and } t \geq 0, \quad (22)$$

to equal the expected remaining amount of effective work anywhere in the shop for station i at time t . By (10), (14), and (22), it follows that the dual objective coefficients $b_i(t)$ correspond to the scaled workload process $W_i(nt)/\sqrt{n}$. Thus, the policy based on dynamic reduced costs is to give priority to classes with smaller values of M_{1k} when $W_1(t) > W_2(t)$, and to classes with smaller values of M_{2k} when $W_1(t) < W_2(t)$.

However, as can be seen from (6), this policy results in many ties. In order to break the ties, we suggest the following tie-breaking procedure. If $W_1(t) > W_2(t)$, then break ties by giving priority to larger values of M_{2k} at station 1 and smaller values of M_{2k} at station 2. If $W_1(t) < W_2(t)$, then break ties by giving priority to smaller values of M_{1k} at station 1 and larger values of M_{1k} at station 2. The reasoning behind this procedure is that station 2 may be threatened with idleness when $W_1(t) > W_2(t)$, and therefore station 1 should attempt to feed station 2 some work by breaking ties with larger values of M_{2k} . Station 2, however, cannot immediately feed itself any work, so it should break ties with smaller values of M_{2k} , so as to serve a job who is closest to exiting the shop. This policy, which uses the dynamic reduced costs and the tie-breaking rule, is referred to as the BROWNIAN policy in Table IV. For the deterministic case, the elements M_{ik} contain the actual remaining effective processing times.

As can be seen in Table IV, there is a striking contrast between the DETERMINISTIC scenario and the six stochastic scenarios. The BROWNIAN policy easily and consistently outperformed the SEPT policy in all six stochastic scenarios. The percentage reduction in mean cycle time ranged from 24.5% to 32.2%, and averaged 28.6%. In the DETERMINISTIC scenario, on the other hand, the SEPT policy outperformed the BROWNIAN policy by 27.2% and outperformed FCFS by 56.9%. This latter improvement is not far from the 53.2% reduction (from FCFS to SEPT) in mean cycle time observed by Conway et al. [2]

in the nine station symmetric shop.

SCENARIO	SEPT	BROWNIAN
DETERMINISTIC	8.88 (± 0.44)	12.2 (± 0.37)
EXPONENTIAL	20.6 (± 1.84)	14.0 (± 1.04)
UNRELIABLE	38.8 (± 2.97)	28.4 (± 1.84)
MEDIUM	25.7 (± 1.73)	19.4 (± 1.19)
RELIABLE	15.9 (± 0.84)	11.3 (± 0.40)
SHORT FAILURES	13.2 (± 0.62)	9.59 (± 0.39)
LONG FAILURES	88.3 (± 8.38)	59.9 (± 4.90)

TABLE IV. Mean cycle times for the symmetric shop.

Thus, although SEPT performs extremely well in the DETERMINISTIC scenario, it is no where near optimal in a stochastic setting. Furthermore, the percentage improvements claimed by scheduling in a deterministic setting are twice the size of the observed improvements in a stochastic setting.

In closing, we should comment on why SEPT performs so well in the DETERMINISTIC scenario. The exponential distribution, which will have some operations with extremely small effective processing times and a few operations with very large effective processing times, allows the SEPT policy to clearly differentiate between jobs. There is a basic tradeoff in scheduling open job-shops between the short term goal of reducing the number of customers in the system, and the longer term goal of avoiding machine idleness; the problem studied in Section 3 is perhaps the simplest problem that illustrates this tradeoff. Under the SEPT policy, jobs that are given priority will very quickly finish their current operation, and will either exit the shop, in which case the short term goal is achieved, or will move to another machine, in which case the longer term goal is achieved. In a closed shop, the only goal is to avoid machine idleness, and thus SEPT also addresses

this objective under the DETERMINISTIC scenario.

However, as soon as unpredictable variability is introduced, these arguments break down. In a stochastic scenario, SEPT is not as effective at reducing the number of customers in the system as the shortest expected remaining processing time rule, and is not as effective at avoiding machine idleness as the BROWNIAN policy in Section 4. Furthermore, in contrast to the dynamic BROWNIAN policies presented in Sections 3 and 5, the SEPT policy is incapable of addressing the tradeoff between these two goals in the open shop setting.

REFERENCES

- [1] Baker, K. R., "Sequencing Rules and Due-Date Assignments in a Job Shop," *Management Science* 30, 1093-1104 (1984).
- [2] Conway, R. W., W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*, Addison-Wesley, Reading, Mass. (1967).
- [3] Harrison, J. M., *Brownian Motion and Stochastic Flow Systems*, John Wiley and Sons, New York, (1985).
- [4] Harrison, J. M., "Brownian Models of Queueing Networks with Heterogeneous Customer Populations," in W. Fleming and P. L. Lions (eds.), *Stochastic Differential Systems, Stochastic Control Theory and Applications*, IMA Volume 10, Springer-Verlag, New York, 147-186 (1988).
- [5] Harrison, J. M. and L. M. Wein, "Scheduling Networks of Queues: Heavy Traffic Analysis of a Simple Open Network," to appear in *Queueing Systems* (1989).
- [6] Harrison, J. M. and L. M. Wein, "Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network," to appear in *Operations Research* (1989).
- [7] Kelly, F. P., *Reversibility and Stochastic Networks*, John Wiley and Sons, New York, (1979).
- [8] Little, J. D. C., "A Proof of the Queueing Formula $L = \lambda W$," *Operations Research* 9, 383-387 (1961).
- [9] Miller, R. G., *Simultaneous Statistical Inference*, McGraw-Hill, New York, (1966).
- [10] Solberg, J. J., "A Mathematical Model of Computerized Manufacturing Systems," paper presented at the 4th International Conference on Production Research, Tokyo (1977).
- [11] Yang, P., "Pathwise Solutions for a Class of Linear Stochastic Systems," unpublished Ph. D. Thesis, Dept. of Operations Research, Stanford University, Stanford, CA, (1988).