

AD-A118 086

GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBUTION--ETC F/G 5/1  
ON FLOW TIME AND DUE DATES IN STOCHASTIC OPEN SHOPS. REVISION: (U)  
APR 82 M PINEDO N00014-80-K-0709  
PDRC-82-05 NL

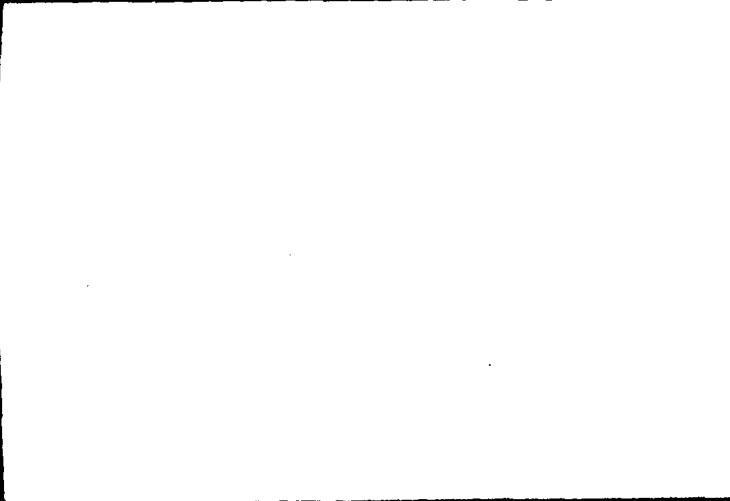
UNCLASSIFIED

10-1  
2000



END  
DATE  
FILMED  
9 82  
DTIC

AD A118086



201 11 08 82

P  
②  
PDRC 82-05  
March 1982  
Revised April 1982

ON FLOW TIME AND DUE DATES  
IN STOCHASTIC OPEN SHOPS

by

Michael Pinedo<sup>†</sup>

PDRC 82-05

DTIC  
SELECTED  
AUG 11 1982  
S H

<sup>†</sup>School of Industrial and Systems Engineering  
Georgia Institute of Technology  
Atlanta, Georgia 30332

DISTRIBUTION STATEMENT A

Approved for public release;  
Distribution Unlimited

This research was supported in part by the Office of Naval Research under Contract No. N00014-80-k-0709 and in part by the National Science Foundation under Grant No. ECS-8115344.

Abstract

In this paper we consider Open Shops where the jobs have exponentially distributed processing times. We determine the policies that in the class of preemptive policies minimize the expected Flow Time. We also consider Open Shops where the jobs have random due dates. Under certain conditions we determine the policies that maximize the expected number of jobs that complete their processing before their respective due dates.

Accession For	
NDIS	<input checked="" type="checkbox"/>
DTIC	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
J. Classification	<input type="checkbox"/>
<i>Little on file</i>	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist Special	
A	



## 1. Introduction

Consider two nonidentical machines and  $n$  identical jobs. Each job has to be processed on both machines in an arbitrary order. The processing time of job  $j$  on machine 1 (2) is a random variable  $X_j(Y_j)$ , exponentially distributed with rate  $\lambda(\mu)$ . This type of scheduling model is usually called an Open Shop.

We will determine policies that in the class of *preemptive* policies minimize or maximize given objective functions. In this class of policies the decision-maker is allowed to interrupt the processing of any job on either one of the two machines at any time. Two policies in this class are intuitively appealing; namely:

(i) The policy that at any point in time instructs the decision-maker, as long as it is possible, to process on each one of the two machines a job that already has been processed on the other machine. This policy may require the decision-maker at times to interrupt the processing of a job. In the sequel this preemptive policy is called the Shortest Expected Remaining Processing Time first (SERPT) policy.

(ii) The policy that, whenever one machine is freed, begins processing, when possible, a job that has not yet been processed on the other machine. This policy does not require any preemptions. Accordingly this nonpreemptive policy is called the Longest Expected Remaining Processing Time first (LERPT) policy in the sequel.

Two machine Open Shop models with *deterministic* processing times have received considerable attention in the literature. In the papers that have appeared in the literature on deterministic Open Shops it is usually assumed that the processing times of a given job on the different machines are not identical. Pinedo and Schrage [8] developed an  $O(n)$  algorithm for finding the sequence that minimizes the completion time of the last job, the so-called makespan, denoted by  $C_{\max}$ . Chin and Achagbue [1] recently showed that the

P

problem of minimizing the sum of the completion times of  $n$  jobs on two machines is NP-hard when preemptions are not allowed. The sum of the completion times of the  $n$  jobs is usually called the Flow Time and is denoted by  $\sum C_j$ , where  $C_j$  represents the completion time of job  $j$ . Whether or not minimization of Flow Time on two machines in the preemptive case is NP-hard, is an open question. Lawler et al. [4] looked at the two machine Open Shop, where job  $j$  has a due date  $D_j$ . They showed that minimization of the maximum lateness, where the lateness  $L_j$  of job  $j$  is defined as  $\max(C_j - D_j, 0)$  is NP-hard in the non-preemptive case, while in the preemptive case it can be solved in polynomial time. *Stochastic* Open Shop models have received less attention in the literature. Pinedo and Ross [7] considered the model with  $n$  identical jobs and two non-identical machines, where the processing time distributions of the two machines are New Better than Used (NBU). They showed that the LERPT policy minimizes the makespan in expectation in the class of non-preemptive policies. Pinedo and Ross also showed that when the processing time distributions of the two machines are exponential, i.e., the model discussed in this paper, the LERPT policy minimizes the makespan *stochastically* in the wider class of policies that do allow preemptions. For this particular model, in which the processing time distributions are exponential, Emmons [2] as well as Pinedo and Ross [7] obtained closed form expressions for the expected makespan under the LERPT policy.

In this paper several stochastic Open Shop models are considered. In Section 2 we use an increasing concave function  $g$  to represent the waiting cost of a job, i.e.,  $g(C_j)$  denotes the waiting cost of job  $j$  with completion time  $C_j$ . We show that SERPT stochastically minimizes  $\sum g(C_j)$ . We also show that SERPT stochastically minimizes the time of the  $i$ -th job completion,  $i = 1, \dots, n-1$ , while LERPT stochastically minimizes the time of the  $n$ -th job completion ( $C_{\max}$ ). In Section 3 we consider Open Shops where the jobs have due dates. The following variant

of the SERPT policy is relevant to this model: Whenever possible, process a job on one machine that is not yet due and that has already been processed on the other machine, while jobs which are already tardy should be dismissed from further consideration. This policy we call the *SERPT with deletions* policy. We assume that the random variables  $D_1, \dots, D_n$  are exchangeable, i.e.,  $D_{i_1}, \dots, D_{i_n}$  has the same joint distribution for all permutations  $i_1, \dots, i_n$  of  $1, \dots, n$ . This implies that each due date has the same marginal distribution  $F$ . Moreover, we also assume that the marginal distribution  $F$  of job  $j$ 's due date  $D_j$ ,  $j = 1, \dots, n$ , is concave, i.e., that the corresponding density  $f$  is decreasing. We show that under these conditions SERPT with deletions maximizes the expected number of jobs to be completed before their respective due dates. However, SERPT with deletions does not maximize this number stochastically. To illustrate why this is not true we consider the special case where all  $n$  jobs have a common random due date  $D$  with a distribution that is concave. For this case it becomes clear that in order to maximize the probability of all  $n$  jobs completing their processing before the common due date the LERPT policy has to be adopted.

## 2. Minimization of the Expected Waiting Cost and Flow Time.

In this section the SERPT policy is studied in detail. Two possible realizations of the process under this policy are depicted in Figure 1. From Figure 1 it follows that a machine is busy as long as two or more jobs remain to be processed on it. However, an idle period may occur on one of the two machines if only one job remains to be processed on it. Consider the following example (see Figure 1a): Machine 1 (2) is processing its last job, while machine 2 (1) is idle; machine 2 (1) remains idle during the time, say  $J_1$  ( $J_2$ ), that it takes machine 1 (2) to complete the processing of the last job; after leaving machine 1 (2) this job starts its processing on machine 2 (1), which then takes an amount

P

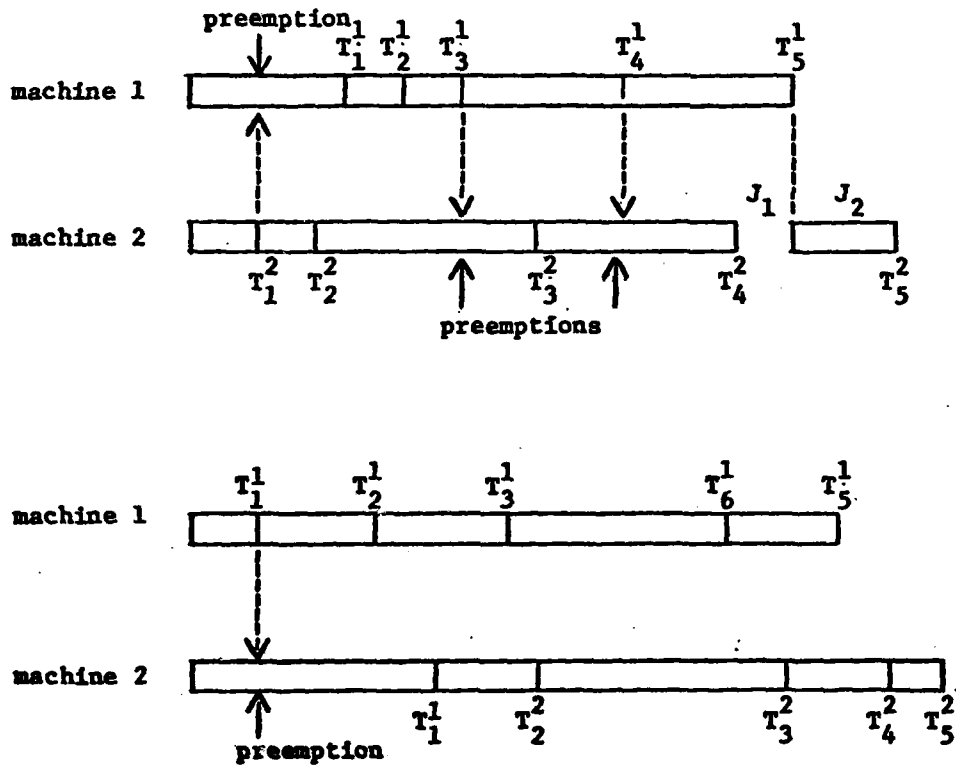


Figure 1

of time  $J_2$  ( $J_1$ ). Let  $T_j^i$ ,  $i = 1, 2$  and  $j = 1, \dots, n$ , denote the time epoch of the  $j$ -th processing time completion on machine  $i$ . It is clear that an idle period occurs on machine 2 (1) if and only if  $T_{n-1}^1 \leq T_{n-1}^2 \leq T_n^1$  ( $T_{n-1}^2 \leq T_{n-1}^1 \leq T_n^2$ ). Let  $k_1, \dots, k_n$ , a permutation of  $1, \dots, n$ , denote the sequence in which the jobs leave the system, i.e., job  $k_1$  is the first job to be completed, job  $k_2$  the second, etc. If  $A \sim B$  denotes that random variables  $A$  and  $B$  have the same distribution, then

$$C_{k_i} \sim \max\left(\sum_{j=1}^i X_j, \sum_{j=1}^i Y_j\right) \quad i = 1, \dots, n-1$$

This implies that the time epoch of the  $i$ -th job completion,  $i = 1, \dots, n-1$ , is a random variable that is the maximum of the independent random variables, each with Erlang distributions. The distribution of the last job completion,  $C_{k_n}$  ( $= C_{\max}$ ), is different. However, we may express  $E(C_{k_n})$  as follows:

$$E(C_{k_n}) = E\left(\max\left(\sum_{j=1}^n X_j, \sum_{j=1}^n Y_j\right)\right) + P(I) \cdot E(\min(J_1, J_2))$$

where  $P(I)$  denotes the probability of an idle period occurring.

**Theorem 1.**

- (i) SERPT stochastically minimizes the  $i$ -th job completion  $C_{k_i}$ ,  $i = 1, \dots, n-1$ .
- (ii) LERPT stochastically minimizes the  $n$ -th job completion  $C_{k_n}$ .
- (iii) SERPT stochastically minimizes the total waiting cost of all jobs,  $\sum_{j=1}^n g(C_j)$ , when the waiting cost function  $g(t)$  is increasing concave in  $t$ .

**Proof:** (i) This follows immediately from the fact that

$$C_{k_i} \sim \max\left(\sum_{j=1}^i X_j, \sum_{j=1}^i Y_j\right) \quad \text{for } i = 1, \dots, n-1$$

- (ii) That LERPT minimizes the makespan stochastically has been shown by Pinedo

and Ross. See the remark following their Theorem 2.

(iii) First we show that SERPT stochastically minimizes the "Flow Time truncated at T," which is defined as  $\sum C_j \cdot I_j(T) + T \cdot \sum (1 - I_j(T))$ , where  $I_j(T) = 1$  if  $C_j < T$  and 0 otherwise.

When one imagines machine 1 (2) to act like a Poisson process with rate  $\lambda$  ( $\mu$ ) until the (n-1) event occurs, it becomes clear that  $T_j^i$ ,  $i = 1, 2$  and  $j = 1, \dots, n-1$ , is not affected by the policy used. It is also clear from Figure 1 that  $T_n^i$ ,  $i = 1, 2$ , does depend on the policy. Let  $\bar{X}$  ( $\bar{Y}$ ) denote the duration of the last job execution on machine 1 (2), i.e., the amount of processing machine 1 (2) still has to do after  $T_{n-1}^1$  ( $T_{n-1}^2$ ). We now show that SERPT minimizes the Flow Time truncated at T for any realization of  $T_j^i$ ,  $i = 1, 2$  and  $j = 1, \dots, n-1$ , and  $\bar{X}$  and  $\bar{Y}$ . From Figure 1 it follows that the waiting cost incurred up to  $\times \max(T_{n-1}^1, T_{n-1}^2)$  is minimized by SERPT. The waiting cost incurred after  $\max(T_{n-1}^1, T_{n-1}^2)$  is positive only when  $T > \max(T_{n-1}^1, T_{n-1}^2)$ . Under SERPT  $\bar{X}$  and  $\bar{Y}$  correspond to the same job, namely job  $k_n$ . Under other policies  $\bar{X}$  and  $\bar{Y}$  may correspond to different jobs. It can be easily verified that the waiting cost incurred after  $\max(T_{n-1}^1, T_{n-1}^2)$  when  $\bar{X}$  and  $\bar{Y}$  correspond to the same job is never larger than the cost incurred after  $\max(T_{n-1}^1, T_{n-1}^2)$  when  $\bar{X}$  and  $\bar{Y}$  correspond to different jobs, even though in the first case an idle period may occur on one of the machines and in the second case no idle period occurs. So the Flow Time truncated at T is minimized by SERPT for any realization of  $T_j^i$ ,  $i = 1, 2$  and  $j = 1, \dots, n-1$  and  $\bar{X}$  and  $\bar{Y}$ . SERPT therefore minimizes the Flow Time truncated at T also stochastically.

The Flow Time truncated at T can be viewed as a cost function  $g(t)$ , equal to  $t$  for  $t \geq T$ . This function is increasing concave. Instead of a single waiting cost function, consider now  $m$  waiting cost functions  $g_i(t)$ ,  $i = 1, \dots, m$ , defined as follows:

$$g_i(t) = \alpha_i \cdot t \quad \text{for } 0 \leq t \leq T_i$$

$$g_i(t) = \alpha_i \cdot T_i \quad \text{for } T_i \leq t.$$

SERPT minimizes each of these cost functions stochastically and hence also

$g(t) = \sum_{i=1}^m g_i(t)$ , where  $g(t)$  is a piecewise linear concave function with  $m + 1$  pieces. Now, any concave function can be approximated by such a piecewise linear concave function, where possibly one of the  $T_i$  has to be chosen  $\infty$ . A standard continuity argument completes the proof of the Theorem.  $\square$

### 3. Stochastic Open Shops with Random Due Dates.

Now we assume that job  $j$ ,  $j = 1, \dots, n$ , has a random due date  $D_j$  and that the due dates  $D_1, \dots, D_n$  are exchangeable, i.e.,  $D_{i_1}, \dots, D_{i_n}$  has the same joint distribution for all permutations  $i_1, \dots, i_n$  of  $1, \dots, n$ . We also assume that the common marginal distribution  $F$  of  $D_j$ ,  $j = 1, \dots, n$ , is concave, which is equivalent to  $D_j$ 's density function  $f$  being decreasing. Our goal is to find in the class of preemptive policies the policy that maximizes the expected number of jobs that complete their processing before their respective due dates.

We first determine the optimal policy in the class of policies which allow preemptions only at the time epochs when the status of a job changes, i.e., either when a due date occurs, or when a machine finishes with the processing of a job or when a job leaves the system. Next we explain why there is no advantage to preempt at any other time epoch, even when allowed to do so. Suppose time  $t$  is a decision moment when the status of one of the jobs changes. Considering only jobs with due dates after time  $t$ , let  $n_{12}$  denote the number of jobs that still need processing on both machines and let  $n_1$  ( $n_2$ ) denote the number of jobs that already have completed their processing on machine 2 (1), but still have to be processed on machine 1 (2). Because the processing time distributions are

exponential, we denote the state  $S$  of the system at time  $t$  by  $(n_{12}, n_1, n_2)$ . The random variable  $N^1(S)$  denotes the number of jobs to be completed after time  $t$  before their respective due dates when the decision-maker acts according to policy  $\pi^1$ . In the subsequent lemma and theorem we repeatedly use the following approach when comparing  $N^1(S)$  with  $N^2(S)$  or with  $N^1(S')$ . We condition on the first event after time  $t$ , which may be a due date occurring or a machine finishing with the processing of a job, and compare  $N^1(S)$  with  $N^1(S')$  (or with  $N^2(S)$ ) under the condition of a particular event occurring first. Let  $\pi^*$  denote the SERPT with deletions policy.

Lemma:

$$E(N^*(k-1, \ell+1, m+1)) \leq 1 + E(N^*(k, \ell, m))$$

Proof: Observe that in state  $(k-1, \ell+1, m+1)$  there is one more job in the system than in state  $(k, \ell, m)$ . Thus one more job has left the system in state  $(k, \ell, m)$ . Let this job be called job 0 and assume that it left the system before its due date in order to justify the first term on the R.H.S. Now the following inequalities can be verified:

- (i)  $E(N^*(0, 1, 1)) \leq 1 + E(N^*(1, 0, 0))$
- (ii)a  $E(N^*(0, \ell+1, 1)) \leq 1 + E(N^*(1, \ell, 0))$
- (ii)b  $E(N^*(0, 1, m+1)) \leq 1 + E(N^*(1, 0, m))$
- (iii)  $E(N^*(0, \ell+1, m+1)) \leq 1 + E(N^*(1, \ell, m))$
- (iv)  $E(N^*(k-1, 1, 1)) \leq 1 + E(N^*(k, 0, 0))$
- (v)a  $E(N^*(k-1, \ell+1, 1)) \leq 1 + E(N^*(k, \ell, 0))$
- (v)b  $E(N^*(k-1, 1, m+1)) \leq 1 + E(N^*(k, 0, m))$
- (vi)  $E(N^*(k-1, \ell+1, m+1)) \leq 1 + E(N^*(k, \ell, m))$

(i) Let  $G$  denote the marginal distribution of the remaining time it takes for a specific due date to occur when we are at decision moment  $t$ . Since  $F$  is con-

cave,  $G$  is concave. Condition on the time it takes machine 1 (2) to complete a job and denote this by  $u$  ( $v$ ). Now

$$E(N^*(0,1,1)) = \bar{G}(u) + \bar{G}(v)$$

while

$$1 + E(N^*(1,0,0)) = 1 + \bar{G}(u+v)$$

and since the distribution  $G$  is concave, (i) follows. The proofs of (ii), ..., (vi) are by induction on  $k, \ell$  and  $m$ . The details can be found in Pinedo [5].

We are now ready for the main result of this section.

Theorem 2: SERPT with deletions maximizes the expected number of jobs that complete their processing before their respective due dates.

Proof: We first show that SERPT with deletions is optimal in the class of policies that allow preemptions only at time epochs when the status of a job changes. It suffices to show that at any such decision moment, in any state, it is no worse to use SERPT with deletions ( $\pi^*$ ), than to take an action not prescribed by this policy (i.e., starting a job that still needs processing on both machines before a job that only needs processing on one machine) and use SERPT with deletions from the next decision moment on. Call this last policy  $\pi'$ . In order to prove optimality of  $\pi^*$  it suffices to show the following inequalities for every  $k, \ell$  and  $m$ :

$$E(N^*(k, \ell, m)) \geq E(N'(k, \ell, m)).$$

Suppose that under  $\pi'$  a job has been started on machine 1 which still needs processing on both machines while under  $\pi^*$  a job has been started on machine 1 that only needs processing on machine 1. Condition on the first event to happen. The only event that makes a difference at the next decision moment is machine 1

finishing with the processing of a job. In case this event occurs  $N'(k, \ell, m)$  becomes  $N^*(k-1, \ell, m+1)$  while  $N^*(k, \ell, m)$  becomes  $1 + N^*(k, \ell-1, m)$ . From Lemma 2 we have

$$E(N^*(k-1, \ell, m+1)) \leq 1 + E(N^*(k, \ell-1, m))$$

This shows that SERPT with deletions is optimal in the class of policies that allow preemptions only at the time epochs that the status of a job changes. Now it remains to be shown that if the decision-maker is allowed to preempt between decision moments it is not optimal to do so. If a random variable  $Y$  has a distribution that is concave, then the random variable  $Z = (Y|Y>t)$  has a concave distribution for any value of  $t$  as well. Assume the decision-maker is allowed to preempt at any of the time epochs  $\Delta, 2\Delta, 3\Delta, \dots, t_1, t_1+\Delta, \dots, t_2, \dots$ , where  $t_i, i = 1, 2, \dots$  is one of the original decision moments. Then a standard induction argument, that starts at the end of the process, shows that the decision-maker never should preempt between the decision moments (for a similar induction argument see the proof of Theorem 3 in Pinedo [6]). This completes the proof of the theorem.  $\square$

Theorem 3 states that the number of jobs to complete their processing before their respective due dates is maximized *in expectation* by SERPT with deletions. However, SERPT with deletions does not maximize this number *stochastically*. Consider the case where all jobs have a common random due date  $D$ , which is a special case of  $D_1, \dots, D_n$  being exchangeable. In this case the SERPT with deletions policy is equivalent to the SERPT policy. In the next theorem we show that SERPT does not maximize the number of jobs to complete their processing before  $D$  *stochastically*. Furthermore, we give an easy proof for the fact that SERPT maximizes this number in expectation when the due date distribution is concave.

Theorem 3:

- (i) SERPT maximizes the probability of completing  $j$  jobs,  $j = 1, \dots, n-1$  before a common due date with an arbitrary distribution.
- (ii) LERPT maximizes the probability of completing all  $n$  jobs before a common due date with an arbitrary distribution.
- (iii) SERPT maximizes the expected number of jobs to be completed before a common due date, when the due date distribution is concave.

Proof: (i) Follows immediately from Theorem 1 (i).

(ii) Follows immediately from Theorem 1 (ii).

(iii) The probability that job  $j$  with completion time  $C_j$  finishes before its due date is  $1 - F(C_j)$ . The objective to be minimized is  $\sum F(C_j)$ . This is equivalent to the objective dealt with in Theorem 1 (iii) and according to Theorem 1 SERPT minimizes this objective in expectation when  $F$  is concave.  $\square$

4. Remarks

For the reader who is familiar with the notation developed by Graham et al. [3] for deterministic scheduling problems it is clear that the models discussed in this paper are stochastic counterparts of the deterministic models  $O_2 | \text{pmtn} | \sum C_j$  and  $O_2 | \text{pmtn}, d_j = d | \sum U_j$ .

The problems discussed in this paper become more complicated when there are more than two machines. It is difficult to formulate an optimal policy for these problems. This jump in complexity when going from two to three machines is a very common phenomenon in both deterministic and stochastic scheduling.

### References

- [1] Chin and Achagbue, private communication, (1982)
- [2] Emmons, H., "The Two-Machine Job Shop with Exponential Processing Times," in *Symposium on the Theory of Scheduling and its Applications*, S. E. Elmaghraby, ed., Springer Verlag (1973).
- [3] Graham, R., E. L. Lawler, J. K. Lenstra, and A. K. G. Rinnooy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey," Ann. of Disc. Math., 5, pp. 287-326, (1979).
- [4] Lawler, E. L., J. K. Lenstra and A. H. G. Rinnooy Kan, "Minimizing Maximum Lateness in a Two Machine Open Shop," Math. Oper. Res., 6, pp. 153-158, (1981).
- [5] Pinedo, M. L., "On Flow Time and Due Dates in Stochastic Open Shops," PDRC Technical Report 82-05, Georgia Institute of Technology, (1982).
- [6] Pinedo, M. L., "Comparisons of Deterministic and Stochastic Scheduling Problems with Release Dates and Due Dates," Oper. Res., in editorial process, (1982).
- [7] Pinedo, M. L. and S. M. Ross, "Minimizing the Expected Makespan in Stochastic Open Shops," Adv. in Appl. Probl., to appear, (1982).
- [8] Pinedo, M. L. and L. Schrage, "Stochastic Shop Scheduling: A Survey," in Proceedings of NATO-School on Deterministic and Stochastic Scheduling, eds. Dempster, Lenstra and Rinnooy Kan, (1982).