

AD-A108 088

GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBUTION F/G 12/1
STOCHASTIC SHOP SCHEDULING: A SURVEY, (U)

SEP 81 M PINEDO, L SCHRAGE

N00014-80-K-0709

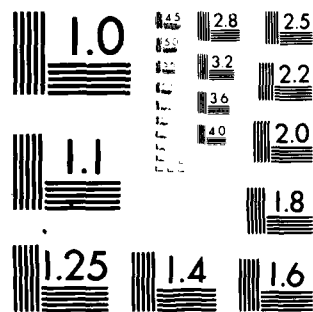
UNCLASSIFIED

PORC-81-12

NL

1-1
K
4-10-80

END
DATE
FILMED
* 82
NTIC

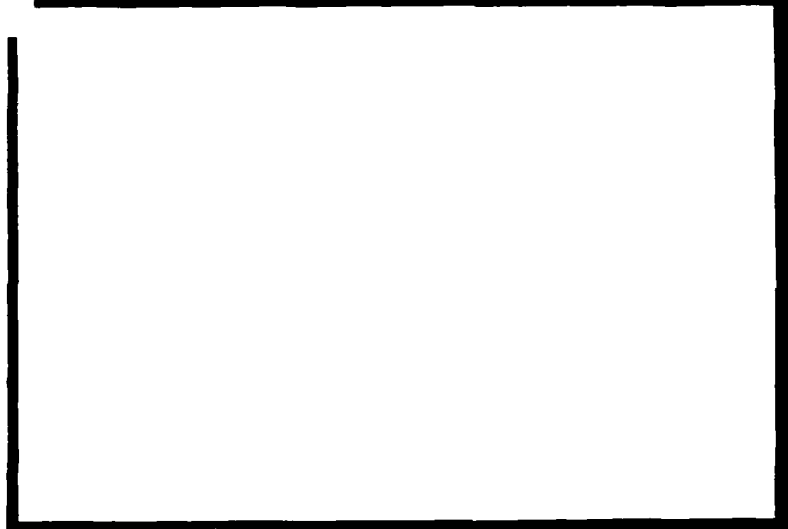
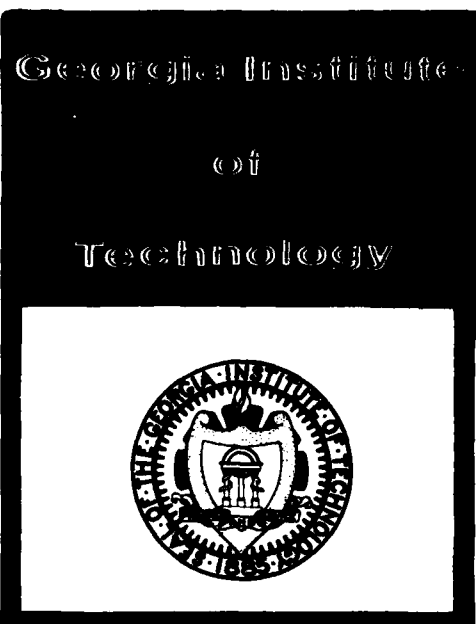


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

9
AD A108088

LEVEL

(11)



DTIC
SELECTED
DEC 3 1981
S A D

This document has been approved
for public release and sale; its
distribution is unlimited.

PRODUCTION and DISTRIBUTION RESEARCH CENTER

DTIC FILE COPY

412669

81 10 27 272

SCHOOL OF INDUSTRIAL
AND SYSTEMS
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
ATLANTA, GEORGIA 30332

5813

Submitted For
1981
Added on file

11/81
1981

A

STOCHASTIC SHOP SCHEDULING : A SURVEY

by

Michael Pinedo[†]

and

Linus Schrage⁺⁺

Contract N00014-80-K-0709
NR-047-221

PDRC 81-12

11/81
1981

[†]Georgia Institute of Technology
School of Industrial and Systems Engineering
Atlanta, Georgia 30332

⁺⁺Linus Schrage
University of Chicago
Chicago, Illinois 60637

This document has been approved
for publication and sale; its
distribution is unlimited.

This work was supported in part by the Office of Naval Research.

STOCHASTIC SHOP SCHEDULING : A SURVEY

Michael Pinedo
Georgia Institute of Technology

Linus Schrage
University of Chicago

ABSTRACT

In this paper a survey is made of some of the recent results in stochastic shop scheduling. The models dealt with include:

- (i) Open shops.
- (ii) Flow shops with infinite intermediate storage (permutation flow shops).
- (iii) Flow shops with zero intermediate storage and blocking.
- (iv) Job shops.

Two objective functions are considered: Minimization of the expected completion time of the last job, the so-called makespan and minimization of the sum of the expected completion times of all jobs, the so-called flow time. The decision-maker is not allowed to preempt. The shop models with two machines and exponentially distributed processing times usually turn out to have a very nice structure. Shop models with more than two machines are considerably harder.

1. INTRODUCTION AND SUMMARY

In this paper an attempt is made to survey the recent results in stochastic shop scheduling. Four shop models are considered; a short description of these follows.

- (i) *Open Shops*. We have n jobs and m machines. A job requires an execution on each machine. The order in which a job passes through the machines is immaterial.
- (ii) *Flow Shops with Infinite Intermediate Storage*. We have n jobs and m machines. The order of processing on the different machines is the same for all jobs; also the sequence in which the

jobs go through the first machine has to be the same as the sequence in which the jobs go through any subsequent machine, i.e. one job may not pass another while waiting for a machine. A flow shop with these restrictions is often referred to as a *permutation* flow shop.

(iii) *Flow Shops with Zero Intermediate Storage and Blocking.* This shop model is similar to the previous one. The only difference is that now there is no storage space in between two successive machines. This may cause the following to happen: Job j after finishing its processing on machine i cannot leave machine i when the preceding job (job $j-1$) still is being processed on the next machine (machine $i+1$); this prevents job $j+1$ from starting its processing on machine i . This phenomenon is called blocking.

(iv) *Job Shops.* We have n jobs and m machines. Each job has its own machine order specified.

Throughout this paper will be assumed that the decision-maker is not allowed to preempt, i.e. interrupt the processing of a job on a machine. For results where the decision-maker is allowed to preempt, the reader should consult the references. In this paper two objectives will be considered, namely (i) minimization of the expected completion time of the last job (the so-called makespan) and (ii) minimization of the sum of the expected completion times of all jobs (the so-called flow time).

This survey is organized as follows: In Section 2 we give a short description of the most important results in *deterministic* shop scheduling (without proofs). The purpose of this section is to enable the reader to compare the results for the stochastic versions of the different models, presented in Section 3, with their deterministic counterparts. For the stochastic models in Section 3 we will not present any rigorous proofs either. However, we will provide for each model heuristic arguments that may make the results seem more intuitive. In Section 4 we discuss the similarities and differences between the deterministic and the stochastic results.

The notation used in this paper is the one developed by Graham et al (5). For example, p_{ij} represents the processing time of job j on machine i . When this processing time is a random variable it will be denoted by p_{ij} . A second example: $02 | p_{ij} \sim \exp(1) | E(C_{\max})$ represents a two machine open shop where the processing times of each job on the two machines are random variables, exponentially distributed with rate one and where the objective to be minimized is the expected makespan ($E(C_{\max})$).

2. DETERMINISTIC SHOP MODELS

This section consists of three subsections: In the first subsection we deal with open shops, in the second one with flow shops and in the last one with job shops.

2.1 Deterministic Open Shops

Consider the two machine case where the makespan has to be minimized. In (5) this problem is referred to as $O_2||C_{\max}$. Gonzalez and Sahni (4) developed an algorithm that finds an optimal sequence in $O(n)$ time. We present here a much simpler method that appears to be new.

Theorem 2.1.1. Let $p_{hk} = \max(p_{ij}, i=1,2, j=1,\dots,n)$. The following schedule minimizes the makespan: If $h=2(1)$, job k has to be started at $t=0$ on machine 1(2); after finishing this processing on machine 1(2) job k 's processing on machine 2(1) has to be postponed as long as possible. All other jobs may be processed in an arbitrary way on machines 1 and 2. Job k may only be started on machine 2(1) either when no other job remains to be processed on machine 2(1) or when only one other job needs processing on machine 2(1) but this job is just then being processed on machine 1(2).

The reader should have little difficulty in proving this theorem. Gonzalez and Sahni (4) showed that the open shop problem with more than two machines is NP-complete.

2.2. Deterministic Flow Shops

Consider first the two machine flow shop with infinite intermediate storage between the machines. We are interested in minimizing the makespan. This problem is usually referred to as $F_2||C_{\max}$. Johnson (7) developed the well-known rule for obtaining the optimal sequence in this problem.

Theorem 2.2.1. The sequence, that puts the jobs with $p_{1j} \leq p_{2j}$ first, in order of nondecreasing p_{1j} and puts the remaining jobs afterwards, in order of nonincreasing p_{2j} , is optimal.

When there are more than two machines in series, the problem is NP-complete (see Gary et al (3)). Research in this area is still going on, focussing mainly on enumerative methods. One special case, however, is easy: Consider the case where the processing times of job i , $i=1,\dots,n$ at all m machines is p_i . In practice such a situation would occur in a communication channel where messages do not change in length when they pass from one station to the next. For this special case Avi-Itzhak (1) established the following theorem.

Theorem 2.2.2. When $p_{1j} = p_{2j} = \dots = p_{mj}$, for $j=1, \dots, n$ any sequence is optimal.

Reddi and Ramamoorthy (14) considered the flow shop with zero intermediate storage and blocking. This problem is not covered in the survey paper of Graham et al (5). We will refer to this flow shop problem as $F|blocking|C_{max}$. Reddi and Ramamoorthy (14) found that $F2|blocking|C_{max}$ can be formulated as a Travelling Salesman Problem with a special structure, a structure that enables one to use an $O(n^2)$ algorithm.

2.3. Deterministic Job Shops

Consider the two machine case $J2||C_{max}$. In this two machine model one set of jobs has to be processed first on machine 1 and after that on machine 2. This set of jobs will be referred to as set A. A second set of jobs has to be processed first on machine 2 and after that on machine 1. This set of jobs will be referred to as set B. Jackson (6), using Johnson's algorithm for $F2||C_{max}$, for obtaining the optimal schedule in $J2||C_{max}$.

Theorem 2.3.1. The following schedule is optimal: All the jobs of set A (B) are to be processed on machine 1 (2) before any job of set B (A) is to be processed on machine 1 (2). The jobs of set A (B) are to be processed on machine 1 (2) in the following order: The jobs with $p_{1j} \leq p_{2j}$ ($p_{2j} \leq p_{1j}$) first in order of nondecreasing p_{1j} (p_{2j}) and the remaining jobs afterwards in order of nonincreasing p_{1j} (p_{2j}). The order in which the jobs of set A (B) are processed on machine 2 (1) does not affect the makespan.

Job shops with more than two machines are NP-complete, even when all processing times are equal to one. But a considerable amount of effort has been dedicated to the research in enumerative methods (see McMahon and Florian (8)).

3. STOCHASTIC SHOP MODELS

This section consists of four subsections: In the first subsection we deal with stochastic open shops. In the second one with stochastic flow shops with infinite intermediate storage between the machines. In the third one we consider stochastic flow shops with zero intermediate storage and blocking. In the last subsection stochastic job shops are considered.

3.1. Stochastic Open Shops

In this subsection we assume that there are two machines available to process n jobs. Each job has to undergo operations on

both machines, the order in which this happens being immaterial. Every time a machine finishes an operation the decision-maker has to decide which job will be processed next on the machine just freed. A policy prescribes the decision-maker which actions to take at the various decision moments; such an action at a decision moment depends on the state of the system at that moment. Observe that a policy only has to instruct the decision-maker what to do as long as there are still jobs which have not yet undergone processing on either machine. This is true for the following reason: When machine 1 (2) becomes free, the decision-maker otherwise only can choose from jobs which have to be processed only on machine 1 (2) and the sequence in which these jobs will be processed on machine 1 (2) does not affect the makespan. Clearly

$$C_{\max} \geq \max \left(\sum_{j=1}^n p_{1j}, \sum_{j=1}^n p_{2j} \right)$$

When one machine is kept idle for some time in between the operations of two jobs, the makespan may be strictly larger than the R.H.S. of the above expression. We may distinguish between two types of idle periods, see Figure 1.

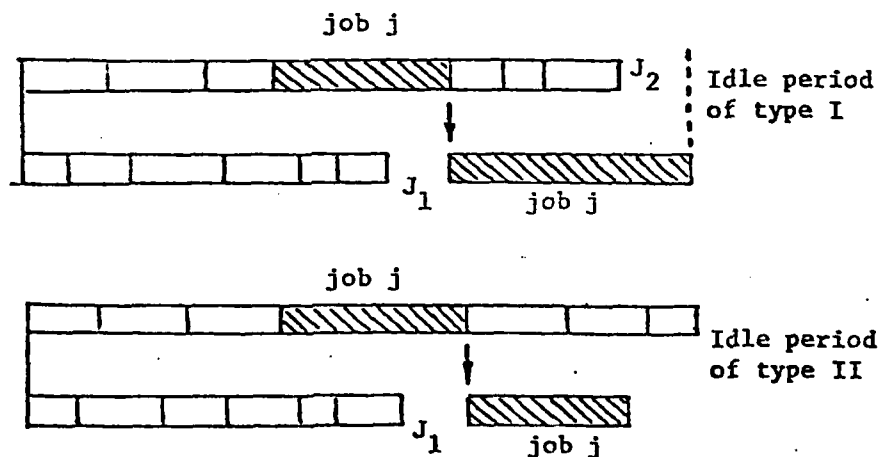


Figure 1

In an idle period of type II, a machine is kept idle for some time, say J_1 , then processes its last job, say job j , and finishes processing job j while the other machine is still busy processing other jobs. It is clear that, although a machine has been kept idle for some time, $C_{\max} = \max \{ \sum_{j=1}^n p_{1j}, \sum_{j=1}^n p_{2j} \}$. In an idle period of type I, a machine is kept idle for some time, say J_1 , then processes the last job, say job j , and finishes processing this job some time, say J_2 , after the other machine has finished all its jobs. Now

$$C_{\max} = \max \left\{ \sum_{j=1}^n p_{1j}, \sum_{j=1}^n p_{2j} \right\} + \min (J_1, J_2)$$

It can be verified easily that only one job can cause an idle period and an idle period has to be either of type I or type II. As the first term on the R.H.S. of the above expression does not depend on the policy, it suffices to find a policy that minimizes $E(\min (J_1, J_2))$.

We will consider now a special case of the two machine open shop model, namely the model $O_2 | p_{ij} \sim \exp(\mu_j) | E(C_{\max})$, where the operations of job j on the two machines are independent and exponentially distributed, both with rate μ_j . From the explanation above it appears intuitive that, in order to minimize the probability that an idle period of type I occurs, jobs that have not received any processing at all should have higher priority than the jobs that already have received processing on the other machine. Moreover among the jobs that have not yet received any processing at all, those with smaller expected processing times should be processed more towards the end. In fact, for $O_2 | p_{ij} \sim \exp(\mu_j) | E(C_{\max})$ the following theorem has been shown in Pinedo and Ross (13).

Theorem 3.1.1. The policy that minimizes the expected makespan is the policy which, whenever any one of the two machines is freed, instructs the decision-maker:

- (i) When there are still jobs which have not yet received processing on either machine, to start among these jobs the one with the largest expected processing time and
- (ii) when all jobs have been processed at least once to start any one of the jobs still to be processed on the machine just freed.

Consider now the model $O_2 | p_{ij} \sim G_i | E(C_{\max})$. In this model we have n identical jobs and two machines. These two machines have different speeds. The distribution of the processing time of a job on machine i , $i=1,2$, is G_i . We assume that G_i , $i=1,2$, is New Better than Used (NBU), i.e.

$$\bar{G}_i(x+y)/\bar{G}_i(x) \leq \bar{G}_i(y)$$

$$x \geq 0, y \geq 0.$$

Under this assumption the following theorem can be proven (see Pinedo and Ross (13)):

Theorem 3.1.2. The makespan is stochastically minimized if the decision-maker starts, whenever a machine is freed, when possible with a job which has not yet been processed on either machine.

The proof of this theorem is a proof by induction, on which we shall not elaborate here. However, the special case $O2|p_{ij} \sim \exp(1)|E(C_{\max})$ can be analyzed further. For this model the following closed form expression for $E(C_{\max})$ under the optimal policy can be obtained:

$$E(C_{\max}) = 2n - \sum_{k=n}^{2n-1} k \binom{k-1}{n-1} \left(\frac{1}{2}\right)^k + \left(\frac{1}{2}\right)^n$$

This expression is obtained by calculating the probability of each job causing an idle period of type I.

Up to now the only objective under consideration has been minimization of expected makespan. Our second objective is minimization of expected flow time, i.e. $E(\Sigma C_j)$. Consider the model $O2|p_{ij} \sim \exp(\mu_i)|E(\Sigma C_j)$: Again we have n identical jobs on two machines with different speeds. For the case where the processing times on machine i , $i=1,2$, are exponentially distributed with rate μ_i , we have the following theorem (see Pinedo (9)):

Theorem 3.1.3. The flow time is stochastically minimized if the decision-maker starts, whenever a machine is freed, when possible with a job that already has been processed on the other machine.

3.2. Stochastic Flow Shops with Unlimited Intermediate Storage Between Machines

In this subsection we consider m machines and n jobs. The n jobs are to be processed on the m machines with the order of processing on the different machines being the same for all jobs. Each job has to be processed first on machine 1, after that on machine 2, etc. At $t=0$ the jobs have to be set up in a sequence, in which they have to traverse the system. We want to determine the job sequence that minimizes either $E(C_{\max})$ or $E(\Sigma C_j)$.

We will consider first the case $m=2$. In Figure 2 is depicted a realization of the process. Intuitively we may expect that in order to minimize the expected makespan, jobs with shorter expected processing times on machine 1 and larger expected processing times on machine 2 should be scheduled more towards the beginning of the sequence, while jobs with larger expected processing times on machine 1 and shorter expected processing times

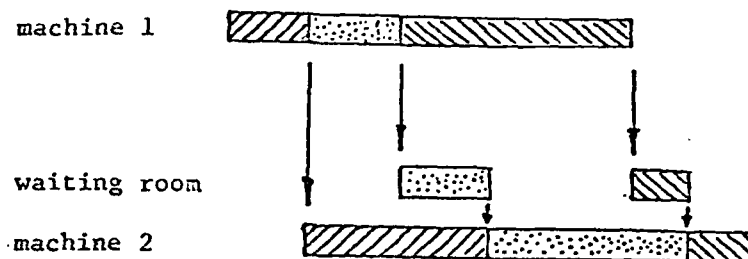


Figure 2

on machine 2 should be scheduled more towards the end of the sequence. In the deterministic version of this problem the optimal sequence is determined by Johnson's rule. Bagga (2) considered $F2|p_{ij} \sim \exp(\mu_{ij})|E(C_{\max})$ i.e. the case where the processing times are exponentially distributed, and proved, through an adjacent pairwise switch argument, the following theorem.

Theorem 3.2.1. Sequencing the jobs in decreasing order of $\mu_{1j} - \mu_{2j}$ minimizes the expected makespan.

This theorem implies that when the processing time of job j on machine 1 (2) is zero, i.e. $\mu_{1j} = \infty$ ($\mu_{2j} = \infty$), it has to go first (last). If there is a number of jobs with zero processing times on machine 1, these jobs have to precede all the others in the sequence. The sequence in which these jobs go through machine 2 does not affect the makespan. A similar remark can be made if there is more than one job with zero processing time on machine 2.

One special case of the flow shop model is of particular importance, namely the case where the processing times of a job on the different machines are independent draws from the same distribution, i.e. $F|p_{ij} \sim G_j|E(C_{\max})$ and $F|p_{ij} \sim G_j|E(C_j)$. Of this case one can easily find examples in real life: Consider a communication channel, where messages do not lose their identity when they pass from one station to the next. From Theorem 3.2.1 follows that for the model $F2|p_{ij} \sim \exp(\mu_j)|E(C_{\max})$ any sequence will be optimal. Weber (15) considered $F|p_{ij} \sim \exp(\mu_j)|E(C_{\max})$, the case for an arbitrary number of machines. With regard to this model he showed the following theorem.

Theorem 3.2.2. The distribution of the makespan does not depend on the sequence in which the jobs traverse the system.

In Theorem 2.2.2 was stated that in the case where the G_j , $j=1, \dots, n$, are deterministic, not necessarily identical, the makespan does not depend on the job sequence either. This property which holds for the exponential and deterministic distributions does not hold for arbitrary distributions. One can easily find counterexamples. In (10) Pinedo considered other examples of $F|p_{ij} \sim G_j|E(C_{\max})$. Before discussing the results presented in (10) we need two definitions:

Definition 1. A sequence of jobs $j_1, j_2, \dots, j_n$ is a SEPT-LEPT sequence if there exists a k such that

$$E(p_{ij_1}) \leq E(p_{ij_2}) \leq \dots \leq E(p_{ij_k})$$

and

$$E(p_{ij_k}) \geq E(p_{ij_{k-1}}) \geq \dots \geq E(p_{ij_n}).$$

(observe that both the SEPT and the LEPT sequences are SEPT-LEPT sequences.)

Definition 2. Distribution G_k and G_l are said to be nonoverlappingly ordered if $P(p_{ik} \geq p_{il})$ is either zero or one. This implies that the probability density functions do not overlap.

Based on these two definitions we can present the following theorem concerning $F|p_{ij} \sim G_j|E(C_{\max})$.

Theorem 3.2.3. For n jobs with nonoverlapping processing time distributions, any SEPT-LEPT sequence minimizes the expected makespan.

Note that this theorem does *not* state that SEPT-LEPT sequences are the only sequences that minimize $E(C_{\max})$. However it is important to observe that $E(C_{\max})$ *does* depend on the sequence and that there *are* sequences which do not minimize $E(C_{\max})$. The next theorem, also concerning $F|p_{ij} \sim G_j|E(C_{\max})$, gives us some idea of how the variance in the processing time distributions affect the job sequences that minimize $E(C_{\max})$.

Theorem 3.2.4. Let $n-2$ jobs have deterministic processing times, not necessarily identical, and let 2 jobs have nondeterministic processing time distributions. Then, any sequence that schedules either one of the stochastic jobs first in the sequence and the other one last minimizes the makespan stochastically.

Based on Theorems 3.2.3 and 3.2.4 and some computational results the following rule of thumb for $F|p_{ij} \sim G_j|E(C_{\max})$ was stated in (10): Schedule jobs with smaller expected processing times and larger variances in the processing times more towards the beginning and towards the end of the sequence and schedule jobs with larger expected processing times and smaller variances

more towards the middle of the sequence. This implies that the optimal sequences have a unimodal form, both as a function of the expectations of the processing time distributions and as a function of the variances of the processing time distributions. Because of this form these optimal sequences may also be referred to as "bowl"-sequences.

Observe that for the problem $F|p_{ij} - G_j|E(\Sigma C_j)$ when the processing time distributions of the jobs are nonoverlapping the SEPT sequence is the only optimal sequence.

Instead of different jobs on identical machines we will consider now the case of identical jobs on different machines, i.e. the processing times of the jobs on a machine are independent draws from the same distribution. The objective now is to find the optimal *machine* sequence (the machine sequence that minimizes the expected makespan) instead of job sequence. This model, which also can be viewed as a tandem queueing model where n customers are waiting at time $t=0$, will be referred to as $F|p_{ij} - G_i|E(C_{\max})$. It can be shown easily that interchanging machines and jobs, i.e. transforming $F|p_{ij} - G_j|E(C_{\max})$ into $F|p_{ij} - G_i|E(C_{\max})$, results in a problem with exactly the same structure. So for $F|p_{ij} - G_i|E(C_{\max})$ Theorems 3.2.3 and 3.2.4, after replacing the words "jobs" for "machines", also hold. One should observe now that the optimal machine sequences stated in these theorems not only minimize $E(C_{\max})$, but minimize $E(C_j)$ for all $j=1, \dots, n$. So these machine sequences also minimize $E(\Sigma C_j)$ (for $F|p_{ij} - G_j|E(\Sigma C_j)$ with nonoverlapping processing time distributions SEPT was the only job sequence that minimized $E(\Sigma C_j)$).

3.3. Stochastic Flow Shops with Zero Intermediate Storage Between Machines

The model in this subsection is rather different from the model in the preceding subsection as now there is no intermediate storage space between the machines. This may have the following consequences: When job j has finished its processing on machine i but cannot be further processed because job $j-1$ is still being processed on machine $i+1$, job j will be held on machine i . However, as long as machine i is holding job j , job $j+1$ may not start its processing on machine i , i.e. job $j+1$ may not leave machine $i-1$. This phenomenon is called blocking. Then models will therefore be referred to as $F|blocking|E(C_{\max})$ and $F|blocking|E(\Sigma C_j)$.

Again, we consider first the case $m=2$. It is clear that whenever a job starts on machine 1, the preceding job starts on machine 2. Let the total time during $[0, C_{\max}]$ that only one machine functions be denoted by I , which is equivalent to the total time during $[0, C_{\max}]$ that one machine is not busy

processing a job. A machine is idle when either a machine is empty or when a job in the first machine is being blocked by a job in the second. During the time period that job j occupies machine 1, there will be some time that only one machine is processing a job: In case $p_{1j} > p_{2j-1}$ machine 1 will keep on processing job j the moment job $j-1$ leaves machine 2. When $p_{1j} < p_{2j-1}$ machine 2 will still be processing job $j-1$ after job j has finished on machine 1. Minimizing $E(C_{\max})$ is equivalent to minimizing $E(I)$ which is equivalent to maximizing the total expected time that both machines are busy processing jobs. Based on this analysis the following result was shown in (10).

Theorem 3.3.1. Minimizing $E(C_{\max})$ in $F2|blocking|E(C_{\max})$ is equivalent to maximizing the total distance in the following deterministic Travelling Salesman Problem. Consider a travelling salesman who starts out from city 0 and has to visit cities $1, 2, \dots, n$ and return to city 0, while maximizing the total distance travelled, where the distance between cities k and l is defined as follows:

$$d_{k0} = 0$$

$$d_{0l} = 0$$

$$d_{kl} = E(\min(p_{1k}, p_{2l})). \quad k \neq 0, l \neq 0.$$

Observe that when the processing times on the two machines are exponentially distributed the distance matrix of the TSP has a very nice structure.

Consider now the model $F2|blocking, p_{ij} = G_j|E(C_{\max})$ where the processing times of a job on the different machines are independent draws from the same distribution. We will say that G_i is stochastically larger than G_k , $G_i \geq_{st} G_k$, when $P(p_{ij} > t) > P(p_{ik} > t)$ for all t . Again, by minimizing $E(I)$ the following theorem can be shown, see (10).

Theorem 3.3.2. When $G_1 \geq_{st} G_2 \geq_{st} \dots \geq_{st} G_n$ and n is even job sequences $n, n-2, n-4, \dots, 4, 2, 1, 3, 5, \dots, n-3, n-1$ and $n-1, n-3, \dots, 5, 3, 1, 2, 4, \dots, n-4, n-2, n$ minimize $E(C_{\max})$. When n is odd job sequences $n, n-2, n-4, \dots, 3, 1, 2, 4, \dots, n-3, n-1$ and $n-1, n-3, \dots, 4, 2, 1, 3, 5, \dots, n-4, n-2, n$ minimize $E(C_{\max})$.

Note that the sequences stated in this theorem are SEPT-LEPT sequences and therefore "bowl"-sequences. This theorem gives us some indication of how the optimal job sequence is influenced by the expected processing times.

Now we will discuss the influence of the variance in the processing times given that the expected values of the processing

times of all jobs are equal, say μ . Consider the following special case: Let the probability density functions of the processing times be symmetric around the mean μ . This implies that the random variables have an upper bound 2μ . We will say that the processing time of job j is more variable than the processing time of job k , $G_j >_v G_k$, when $G_j(t) \geq G_k(t)$ for $0 \leq t \leq \mu$ and (because of symmetry) $G_j(t) \leq G_k(t)$ for $\mu \leq t \leq 2\mu$. Distributions which satisfy these symmetry conditions are:

- (i) The Normal Distribution, truncated at 0 and at 2μ
- (ii) The Uniform Distribution.

The probability density functions of these distributions are depicted in Figure 3.

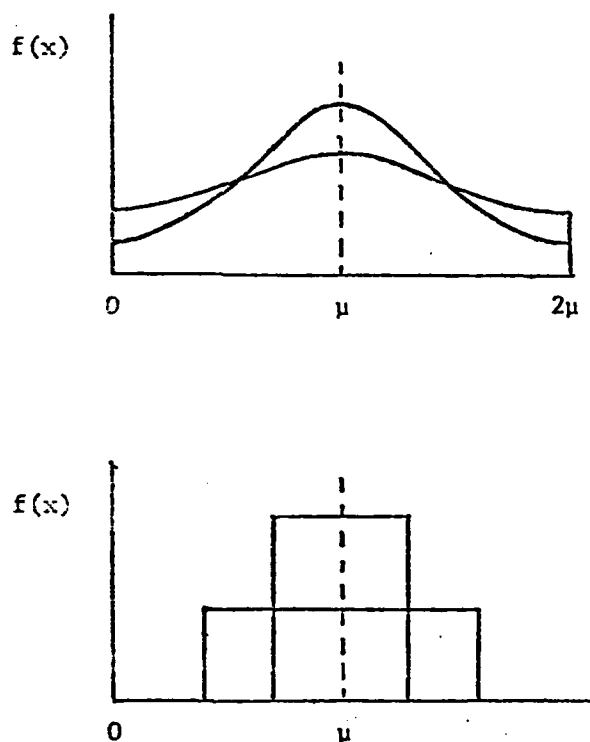


Figure 3

Theorem 3.3.3. When $G_1 <_v G_2 <_v \dots <_v G_n$ and when n is even job sequences $n, n-2, n-4, \dots, 4, 2, 1, 3, 5, \dots, n-3, n-1$ and $n-1, n-3, \dots, 5, 3, 1, 2, 4, \dots, n-4, n-2, n$ minimize $E(C_{\max})$. When n is odd job sequences $n, n-2, n-4, \dots, 3, 1, 2, 4, \dots, n-3, n-1$ and $n-1, n-3, \dots, 4, 2, 1, 3, \dots, n-4, n-2, n$ minimize $E(C_{\max})$.

So from Theorems 3.3.2 and 3.3.3 we observe that here too "bowl"-sequences are optimal and therefore the rule of thumb stated in subsection 3.2 would be valid here, too.

In (10) Pinedo also considered $F|blocking, p_{ij} \sim G_j|E(C_{max})$. For this problem a theorem very much like Theorem 3.2.3 could be proven.

Theorem 3.3.4. For n jobs with nonoverlapping processing time distributions a job sequence minimizes $E(C_{max})$ if and only if it is SEPT-LEPT.

So this theorem, too, emphasizes the importance of "bowl"-sequences. In (10) the author was unable to present a theorem similar to Theorem 3.2.4. However, the following conjecture was stated.

Conjecture. Let $n-2$ jobs have identical deterministic processing times, say with unit processing times and let two jobs have nondeterministic processing times with symmetric probability density functions and mean one. Then, any sequence which schedules either one of the stochastic jobs first and the other one last minimizes $E(C_{max})$.

Observe that for $F|blocking, p_{ij} \sim G_j|E(C_{max})$, when the processing times of the jobs are nonoverlappingly distributed, the SEPT sequence is the only optimal sequence.

Consider now again the case of n identical jobs and m different machines. This implies that the processing times of the jobs on the different machines are independent draws from the same distribution. Again, we would like to know the optimal order in which to set up the machines in order to minimize $E(C_{max})$. This model will be referred to as $F|blocking, p_{ij} \sim G_i|E(C_{max})$. In subsection 3.2 it was mentioned that, in the case of infinite intermediate storage, interchanging jobs and machines results in a model with exactly the same structure. With blocking, however, interchanging machines and jobs does change the structure of the model significantly. In (11) Pinedo showed the following result with regard to $F|blocking, p_{ij} \sim G_i|E(C_{max})$.

Theorem 3.3.5. The expected makespan of n jobs in a system with $m-2$ identical deterministic machines with unit processing times and 2 nonidentical stochastic machines, both with mean one and symmetric density functions, is minimized if one of the stochastic machines is set up at the beginning of the sequence and the other at the end of the sequence.

This theorem appears to be the perfect dual of the conjecture stated before. The next theorem, however, will illustrate the

difference between $F|\text{blocking}, p_{ij} \sim G_j | E(C_{\max})$ and $F|\text{blocking}, p_{ij} \sim G_i | E(C_{\max})$.

Theorem 3.3.6. The expected makespan of n jobs in a system with $m-2$ identical machines with distributions G_1 and two identical machines with distribution G_2 , where G_2 is nonoverlappingly larger than G_1 , is minimized, when one of the two slow machines is set up at the beginning of the sequence and the other one at the end of the sequence.

This theorem is quite different from Theorem 3.3.4. Based on these last two theorems, some other minor results and extensive simulation work, it appeared that the optimal machine sequences are not "bowl"-sequences but so-called "sawtooth"-sequences. These may be described as follows: Suppose we have $m-1$ identical machines with distributions G_1 and m identical machines with distribution G_2 (for a total of $2m-1$ machines), where G_2 is nonoverlappingly larger than G_1 , then we conjecture that the optimal machine sequence puts a slow machine at the beginning of the sequence, followed by a fast machine in the second place, a slow machine in the third place, etc. This sequence has the shape of a "sawtooth". Suppose now we have $m-1$ identical deterministic machines with unit processing times and m identical machines with mean one and symmetric density function, then we conjecture that the optimal sequence puts a stochastic machine at the beginning of the sequence, followed by a deterministic machine in the second place, a stochastic machine in the third place, etc. This sequence, too, has the shape of a "sawtooth".

3.4. Stochastic Job Shops

Very little work has been done on stochastic job shops. The main reason is that these models are even harder than the stochastic flow shops with infinite intermediate storage between machines; these flow shops are just very special cases of job shops. The only result known up to now concerns $J_2 | p_{ij} \sim \exp(\mu_{ij}) | E(C_{\max})$. In this two machine model one set of jobs has to be processed first on machine 1 and after that on machine 2. This set of jobs will be referred to as set A. A second set of jobs has to be processed first on machine 2 and after that on machine 1. This set of jobs will be referred to as set B. In (12) Pinedo showed that Bagga's Theorem for $F_2 | p_{ij} \sim \exp(\mu_{ij}) | E(C_{\max})$ can be generalized into a theorem for $J_2 | p_{ij} \sim \exp(\mu_{ij}) | E(C_{\max})$. However, for $J_2 | p_{ij} \sim \exp(\mu_{ij}) | E(C_{\max})$ we cannot speak anymore of an optimal sequence, we have to speak of an optimal policy which instructs the decision-maker in any state what action to take.

Theorem 3.4.1. The optimal policy instructs the decision-maker, whenever machine 1 (2) is freed, to start processing of the

remaining jobs of set A (B) that did not yet undergo processing on machine 1 (2) the one with the highest value of $\mu_{1j} - \mu_{2j}$ ($\mu_{2j} - \mu_{1j}$). If no jobs of set A (B) remain that did not yet undergo processing on machine 1 (2), the decision-maker may start any one of the jobs of set B (A) that already have finished their processing on machine 2 (1).

4. CONCLUSIONS

It is clear that the stochastic shop scheduling problems are not all that easy. The two machine models with exponential processing times are in general tractable, just like the two machine models with deterministic processing times. For the deterministic versions of $F2||C_{\max}$ and $J2||C_{\max}$ the algorithms are $O(n \log n)$; for the versions of $F2||E(C_{\max})$ and $J2||E(C_{\max})$ with exponential processing times the algorithms are also $O(n \log n)$. For the deterministic version of $O2||C_{\max}$ the algorithm is $O(n)$. The version of $O2||E(C_{\max})$ with exponentially distributed processing times is harder. For the special case $O2|p_{ij} \sim \exp(\mu_j)|E(C_{\max})$ the algorithm is already $O(n \log n)$. We have not yet been able to determine the complexity of the more general case $O2|p_{ij} \sim \exp(\mu_{ij})|E(C_{\max})$. Models with three machines or more appear to be very hard. Only results of a qualitative nature were obtained (e.g. "bowl"-sequences, "sawtooth"-sequences). This jump in complexity when we go from two machines to three occurs also when the processing times are deterministic; non-preemptive shop scheduling with three or more machines are consistently NP-complete.

REFERENCES

- (1) Avi-Itzhak, B., "A Sequence of Service Stations with Arbitrary Input and Regular Service Times": 1965, Man. Sci. 11, pp. 565-573.
- (2) Bagga, P.C., "N-Job, 2 Machine Sequencing Problem with Stochastic Service Times": 1970, Opsearch 7, pp. 184-199.
- (3) Garey, M.R., Johnson, D.S. and Sethi, R., "The Complexity of Flowshop and Jobshop Scheduling": 1976, Math. Operations Res. 1, pp. 117-129.
- (4) Gonzalez, T., and Sahni, S., "Open Shop Scheduling to Minimize Finish Time": 1976, J. Assoc. Comput. Mach. 23 pp. 665-679.
- (5) Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinnoog Kan, A.H.G., "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey": 1979, Ann. Discr. Math. 5, pp. 287-326.

- (6) Jackson, J.R., "An Extension of Johnson's Results on Job Lot Scheduling": 1956, Naval Res. Logist. Quart. 3, pp. 201-203.
- (7) Johnson, S.M., "Optimal Two and Three Stage Production Schedules with Setup Times Included": 1954, Naval Res. Logist. Quart. 1, pp. 61-68.
- (8) McMahon, G.B. and Florian, M., "On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness": 1975, Operations Res. 23, pp. 475-482.
- (9) Pinedo, M.L., "Minimizing the Expected Flow Time in a Stochastic Open Shop with and without Preemptions": 1981, in preparation.
- (10) Pinedo, M.L. "Minimizing the Makespan in a Stochastic Flow Shop": 1981, to appear, Operations Research.
- (11) Pinedo, M.L., "On the Optimal Order of Stations in Tandem Queues": 1981, to appear, Proceedings of the Applied Probability-Computer Science: The Interface, Conference at Boca Raton, Florida.
- (12) Pinedo, M.L. "A Note on the Two Machine Job Shop with Exponential Processing Times": 1981, to appear, Naval Research Logistics Quarterly.
- (13) Pinedo, M.L. and Ross, S.M., "Minimizing the Expected Makespan in a Stochastic Open Shop": 1981, submitted to Journal of Applied Probability.
- (14) Reddi, S.S. and Ramamoorthy, C.V., "On the Flowshop Sequencing Problem with No Wait in Process": 1972, Operational Res. Quart. 23, pp. 323-330.
- (15) Weber, R.R., "The Interchangeability of Tandem $M|1$ Queues in Series": 1979, J. of Applied Prob. 16, pp. 590-695.
- (16) Wisner, D.A., "Solution of Flowshop-Scheduling Problem with No Intermediate Queues": 1972, Operations Res. 20, pp. 639-697.

CORRECTIONS

- P. 2 line 47 variables, exponentially distributed with rate one and where the
- P. 6 line 23 shop model, namely the model $O2|p_{ij} \sim \exp(\mu_j)|E(C_{\max})$, where the
- P. 6 line 23 shop model, namely the model $O2|p_{ij} \sim \exp(\mu_j)|E(C_{\max})$, where the
- P. 7 line 13 $\exp(1)|E(C_{\max})$ can be analyzed further. For this model the fol-
- P. 7 line 14 lowing closed form expression for $E(C_{\max})$ under the optimal pol-
- P. 7 line 23 Up to now the only objective under consideration has been
- P. 7 line 24 minimization of expected makespan. Our second objective is min-
- P. 7 line 15 istry can be obtained:
- P. 8 line 41 distribution, i.e. $F|p_{ij} \sim G_j|E(C_{\max})$ and $F|p_{ij} \sim G_j|E(C_j)$. Of
- P.10 line 7 Because of this form these optimal sequences may also be referred
- P.11 line 22.5 $d_{k0} = 0$
- P.11 line 24.5 $d_{0l} = 0$
- P. 6 line 23 shop model, namely the model $O2|p_{ij} \sim \exp(\mu_j)|E(C_{\max})$, where the
- P. 14 line 51 *Theorem 3.4.1.* The optimal policy instructs the decision-maker,
- P. 14 line 52 whenever machine 1 (2) is freed, to start processing of the
- Theorem 3.3.3.*
- P. 12 line 43 *Theorem 3.3.3.*
- P. 12 line 43 *Theorem 3.3.3.* When $G_1 \leq_v G_2 \leq_v \dots \leq_v G_n$ and when n is even
- P. 4 line 22 for $F2||C_{\max}$, obtained an optimal schedule for $J2||C_{\max}$
- P. 12 line 43 *Theorem 3.3.3.* When $G_1 \leq_v G_2 \leq_v \dots \leq_v G_n$ and when n is even
- P. 10 line 44 machine $i-1$. This phenomenon is called blocking. These models
- P. 16 line 14 in preparation.
- P. 16 line 14 Georgia Institute of Technology, Technical Report.

DATE
FILMED
8