USING SECOND MOMENT INFORMATION IN STOCHASTIC SCHEDULING

Marilyn J. Maddox
Ford Motor Company
Alpha Simultaneous Engineering
Suite 100
15303 Commerce Drive
Dearborn, MI 48120

John R. Birge
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 93-10

February 1993

Using Second Moment Information in Stochastic Scheduling

Marilyn J. Maddox
Ford Motor Co.
Alpha Simultaneous Engineering
Suite 100
15303 Commerce Drive S.
Dearborn, MI 48120
313-248-1280
FAX: 313-390-1774

John R. Birge
Industrial and Operations Engineering Department
University of Michigan
1205 Beal Ave.
Ann Arbor, MI 48109
313-764-9422
FAX: 313-764-3451

Using Second Moment Information in Stochastic Scheduling

Abstract

We present a scheduling heuristic for the stochastic job shop. In a stochastic job shop, operation processing times may vary randomly, or machines may fail at random intervals, or both. Dispatching rules based on substituting expected values for random quantities are often used for scheduling in this uncertain environment. The scheduling heuristic we propose performs restricted dynamic updating of an initial schedule for the shop, using limited distribution information about the random quantities. In particular, we allow general dependence among the stochastic quantities and require at most two moments of each distribution function. Our objective is to minimize expected tardiness. We investigate the benefits of the scheduling heuristic versus a dispatching rule using a simulation study.

Keywords: Scheduling, Stochastic programs, Simulation

In the static job shop problem, *n* jobs must be scheduled for processing on *m* machines. Each job consists of a number of operations; each operation runs on a specific machine or group of machines. The order of machines a job must visit is known. In the deterministic job shop problem, machines are continuously available and processing times for all operations are deterministic. In stochastic versions of the problem, operation processing times may be random or the machines may be subject to random disruptions or both. We consider a stochastic version of the job shop problem. We assume that preemption of one operation by another is not allowed. In addition, we assume that setup times are negligible in comparison to operation processing times and that material handling time between operations is insignificant. Basically, ignoring our consideration of stochastic elements, these assumptions define a simple job shop as described in Conway et. al. (1967).

We concentrate on minimizing the expected tardiness in a job shop. We chose to minimize tardiness because reducing tardiness is a goal in many actual job shops. Smith et. al. (1986) report that tardiness is the single most important objective to practitioners today. In addition, since tardiness costs are nonstationary in time, results for tardiness may be extended to more general functions than results for stationary objectives such as completion time.

A schedule that is optimal with respect to minimizing expected tardiness may perform quite poorly given the actual situation in the job shop. Ideally, we desire a schedule that performs well in a variety of situations. One option is to select a schedule that does well in the worst case scenario. This minimizes the maximum cost of any scheduling problem. However, it does not insure good average performance. Nevertheless, in many cases, managing the worst case scenarios is important. In this paper, we describe a heuristic scheduling procedure which combines the average and worst-case concepts by selecting schedules which minimize over the set of possible schedules the maximum expected schedule cost over distributions with known limited information.

This paper is organized as follows. The remainder of this section describes the stochastic job shop problem and reviews the relevant literature. The second section describes the heuristic scheduling algorithm. The third section presents a simulation study which compares the expected tardiness of schedules developed using the heuristic with schedules developed under a dispatching rule. The section concludes with observations from the study and indicates directions for future research.

1. Background

1.1 Stochastic Job Shop Problem

Under the scenario of random processing times or machine disruptions, the stochastic job shop problem can be modelled as a stochastic program with general recourse. This means the optimal schedule may vary with the realized outcomes of the random components. In this model, the decision of which job to schedule next can depend on any decisions made and events that have occurred in the past, i.e., only nonanticipative decisions are allowed. Thus, the sequence of operations actually processed on a machine depends on the events which occurred during processing. This flexibility may make the general recourse schedule extremely complicated to implement. Simplification is possible by restricting consideration to simple recourse solutions, although the solution obtained is no longer guaranteed to be optimal.

In simple recourse, decisions are independent of all but the initial conditions. A simple recourse schedule for the job shop is completely specified by the processing order on each machine. Simple recourse schedules are inflexible because they do not permit scheduling changes once processing begins. In reality, it is often desirable to update the schedule periodically during processing, using current information on the job shop status. Our heuristic performs limited updating of an initial schedule based on current shop status.

Although most job shops are stochastic, scheduling models usually ignore uncertainty (Graves (1981)). In these models, expected values are substituted for their actual random counterparts. Research on stochastic problems in various applications shows that this procedure of substituting expected values yields disappointing results, see Igelmund (1983b), Kallberg et.al. (1982), Mittenthal (1986), and Wets (1983). In practice, the schedules produced by these deterministic models must be adapted to constantly changing circumstances. In very uncertain environments, a priori schedules are abandoned altogether. Instead, each time a machine becomes idle, a dynamic priority rule is used to choose one operation to process next among the operations queued at the machine. The priority rule assigns an index, based on local operation information, to each operation in the queue. Priority rule scheduling ignores future implications of current decisions. In addition, it reduces the possibility of closely coordinating a job shop with its associated manufacturing processes.

The heuristic described here creates an initial schedule for the job shop before production begins. As processing proceeds, this schedule is updated based on current information on the job shop status. Thus, the heuristic incorporates the strengths of both simple recourse scheduling and dynamic priority rule scheduling. It uses up-to-date information about the job shop status and analyzes the effects of current decisions on the future schedule.

1.2. Literature Review

Much of the existing literature on the stochastic job shop problem is devoted to special cases, either restricting the number or type of machines or requiring explicit distributions for the random elements. The processing times may be stochastic or the machines may be subject to random breakdowns or both.

Many of the stochastic scheduling articles focus on the single machine problem. Both the stochastic job and stochastic machine cases are represented. Some early work on a stochastic job problem was done by Blau (1973). More recently, Glazebrook (1983, 1984, 1987a, b) considered both stochastic job and stochastic machine scenarios. Mittenthal (1986), Mittenthal and Birge (1987), and Birge et. al.

(1987) used stochastic programming to model the stochastic single machine problem. Most of these papers require complete knowledge of the distribution function of the random elements. Independence among random elements is also a common assumption.

There is little published work on the multi-machine, stochastic job shop problem. Assuming independent, exponential job processing times, Weiss and Pinedo (1980) found rules for processing *n* jobs on non-identical processors to minimize stationary objectives.

Igelmund and Radermacher (1983a,1983b) and Möhring, Radermacher and Weiss (1984,1985) studied a stochastic, resource constrained, project scheduling problem. They assumed that the durations of project activities, equivalent to operations in job shop scheduling, were random. They also assumed that the joint distribution of the durations of the activities was known. They considered simple recourse solutions as well as schedules created from a combination of several simple recourse solutions. The authors proved various analytical properties for these classes of schedules.

Solel (1986) presented a general precedence constrained stochastic scheduling model which encompasses the models of Möhring et. al. and Weiss and Pinedo. She modelled the problem as a Piecewise Deterministic (Markov) process and, assuming that the joint distribution of the job durations was known, obtained existence results and characterized the optimal strategy when the objective was to minimize the expectation of a cost function. The cost functions she considered had to be linearly bounded and monotonically increasing as a function of the completion times of the jobs.

All of the models discussed so far explicitly take uncertainty into account. Job shop rescheduling is a heuristic for the stochastic problem that ignores randomness during the planning stage. In job shop rescheduling, a schedule for the job shop is initially constructed using deterministic methods. When a disruption occurs, the job shop is rescheduled from that point onward. The basic dilemma in this heuristic is how to reschedule quickly enough that work flow in the job shop is not disturbed. Of issue are the frequency of rescheduling, whether the whole job shop should be rescheduled, and which deterministic scheduling heuristic should be used. Rescheduling has been studied by Muhleman et. al. (1982), Bean et. al. (1986), Bean and Birge (1985), and Birge (1985), among others.

The stochastic scheduling problem we study is more general than most of the models discussed in the stochastic scheduling literature. We consider multiple machines and assume only limited information about the distributions of the random elements. In particular, we allow general dependence among the stochastic quantities and require at most the first two moments of each distribution function. Due to the computational difficulty of solving general recourse stochastic programs, and the complexity of the resulting solutions, we consider a heuristic solution procedure in this paper.

2. Heuristic Scheduling Procedure

The first step in the heuristic algorithm is to create an initial schedule for the job shop prior to actual processing. This schedule dictates the first operation to load on each machine. The remaining steps of the scheduling heuristic control sequencing once production begins. Each time an operation finishes processing, the heuristic updates the current schedule and selects an operation to put on the machine next. The heuristic stops when all operations have finished processing.

This section consists of two subsections. The first subsection describes a method for creating an initial schedule for the job shop. The second subsection discusses a method for updating this schedule as production progresses.

2.1 Initial Schedule

The heuristic algorithm begins by creating an initial schedule for the job shop. In realistic scheduling environments, initial schedules which balance several objectives may be desired. Here, we schedule with the objective of minimizing expected tardiness. The initial schedule may be created by substituting expected values for random quantities and using a deterministic scheduling method. For example, a deterministic dispatching rule may be used. On a small problem, a branch and bound algorithm discussed in Maddox (1988) may be used to create an initial schedule which has the minimum worst-case expected tardiness among all such schedules. Evaluating the true expected tardiness of the initial schedule is difficult because of the stochastic nature of the problem and the number of possible alternatives.

2.2 Updating the Schedule

When an operation finishes processing, the heuristic updating procedure determines the next operation to place on that machine. The heuristic chooses either the operation specified in the current schedule or selects a replacement operation. Since the procedure sequentially updates the initial schedule, there is always a complete schedule available. This permits evaluation of the future effects of current decisions. If the updating is too extensive, the heuristic essentially becomes a dispatching rule, since the future sequencing information it utilizes is unreliable. If the information the algorithm uses is incorrect, the algorithm could potentially create schedules with higher tardiness than an algorithm which ignores such information. On the other hand, not allowing enough revisions may make the heuristic algorithm too inflexible, given the stochastic nature of the job shop. When an operation finishes processing, the key question is what operation to put on the machine next. Hence, in our heuristic algorithm we only permit updates of the next operation to be placed on the machine.

We refer to the machine involved in the current decision as the current machine, M. Any machines identical to the current machine are identified as the current machine group, N. M is included in machine group N.

The process of deciding which operation to load on machine M involves four steps:

1. Identify the set of potential "replacement operations" for operation I, the operation scheduled next on machine M in the current schedule.
2. For each replacement operation, develop a replacement schedule in which the replacement operation is scheduled next on machine M.
3. Choose the ''best'' replacement operation from the set of potential replacements.
4. Load either operation I or the ''best'' replacement operation on machine M.

The first three steps are described in more detail below.

Step 1: Identify Potential Replacement Operations

The number of potential replacement operations affects both the speed and the effectiveness of the algorithm. If a large number of potential replacements are identified, evaluating which one is ''best'' may require a significant amount of time. On the other hand, limiting the number of replacements limits the flexibility of the algorithm.

The set of potential replacements could include all unprocessed operations on machine group N. However, unless the current job shop conditions are drastically different from the expected ones, operations currently sequenced "farther" from the current time T are less likely to be good replacements than operations scheduled closer to T. Technological precedence constraints are also likely to prohibit moving operations not currently scheduled "near" T.

Of particular interest as replacements are operations that are now scheduled to start before or soon after the expected completion time of operation I in the current schedule. When the variance of operation I increases, the set of replacement operations should increase as well. Therefore, we included an operation in the set of replacement operations if it ran on machine group N, and it was expected to start before the expected completion time of operation I plus a constant times the standard deviation of operation I's processing time. The constant is a parameter of the algorithm.

Step 2: Develop Replacement Schedules

Once an operation has been identified as a potential replacement, the heuristic develops a schedule with that operation placed next on machine M. Let J denote a replacement operation. If J is originally scheduled on machine M the operations scheduled between operations I and J are bumped back one position in the replacement schedule. If operation J is scheduled on another machine in machine group N, the two operations are interchanged in the replacement schedule.

The positions of the predecessor and successor operations of operation J should also be updated in the replacement schedule. This increases the impact of the schedule change. The predecessor and successor operations must be resequenced carefully, since incorrectly placing them may cause unwanted delays and machine idle time. Determining their new positions is difficult, however, since the exact completion times of the unprocessed operations are unknown.

In the heuristic algorithm, if replacement operation J is now sequenced before an operation H which it was originally scheduled after, the algorithm checks to see if any successors of H are scheduled before successors of J on a machine. If any such successors are found, interchanges are made to rectify the situation. In the simulation, a version of the heuristic with no successor updates is tested as well, to judge the viability of this approach.

Step 3: Choose the "Best" Replacement Operation

Ideally, once each potential replacement schedule has been developed, the schedule with the lowest expected tardiness is selected as the replacement schedule. If the expected tardiness of this replacement schedule is less than that of the current schedule, the replacement operation is loaded on the current machine and the replacement schedule becomes the current schedule. Unfortunately, the expected tardiness of a schedule cannot be calculated exactly. The tardiness random variable has a complicated distribution since it is a function of the maximum of a sum of possibly dependent random variables. It is possible, however, to bound the expected tardiness of a schedule. Birge and Maddox (1992) and Maddox(1988) derive an upper bound on the expected tardiness of a schedule. A lower bound on the expected tardiness of a schedule can be obtained by substituting expected values for the random processing times and determining the resulting tardiness.

These bounds bracket expected tardiness in some interval. We use this interval to determine the operation to schedule. Using the lower bound assumes a "best-case" distribution. The upper bound assumes a "worst-case" distribution. The schedule with either the minimum best-case or worst-case expected tardiness becomes the current schedule. The operation dictated by this schedule is loaded onto machine M.

Developing and bounding the expected tardiness of each replacement schedule may be time consuming. An alternative is to choose one replacement operation and build only one replacement schedule. This requires a procedure for determining the ''best'' replacement from the set of potential replacement operations. The goal is to determine the operation which is most likely to have the schedule with the smallest bound on expected tardiness. One possible way to pick the operation is to use a dispatching rule. Dispatching rules which perform well in studies minimizing tardiness are most appropriate. In the version of the heuristic tested in the simulation, a replacement schedule is built for each replacement operation.

3. Simulation

3.1 Simulation Description

The purpose of the simulation was to compare the average tardiness performance of the heuristic algorithm and a dynamic dispatching rule. Dynamic dispatching rules are often used to schedule stochastic, multiple machine job shops. The heuristic algorithm tested was described in the previous section. The initial schedule was generated assuming operation processing times equalled their expected values. The Apparent Tardiness Cost (ATC) dispatching rule (Vepsalainen and Morton (1987)), described below, was used to generate the initial schedule. Each time an operation completed processing, a set of potential replacement operations was identified and two replacement schedules were developed for each. In one replacement schedule, only the position of the potential replacement operation was altered. In the other, the position of the replacement operation and the positions of all of its technological successors were updated. The length of the time interval in which replacements are chosen is a parameter of the simulation. The length is a function of the standard deviation of operation I's processing time.

Under each set of experimental conditions, two forms of the heuristic were tested. In one, the schedule with the minimum worst-case expected tardiness, among the replacement schedules and the current schedule, was chosen as the new current schedule. The bounding procedure discussed in Birge and Maddox (1992) and Maddox (1988), utilizing two moments of each processing time distribution, was used to determine the value of each schedule. In the other version of the heuristic, the schedule with the minimum best-case expected tardiness was chosen. Note that both of these approaches recognize the stochastic nature of the problem but make different assumptions about the *distribution* of processing times. The choice of best-case versus worst-case depends on the decision-maker's risk aversion. The ATC dispatching rule was chosen for testing against the heuristic because of its superior performance in deterministic studies. The ATC of operation j of job i is

$$\frac{1}{E(\xi_j)}\exp(-[\frac{T_i-t-\overline{\xi_j}-\sum_{q=j+1}^{l_i}(W_q+\overline{\xi_q})}{kp}]^{+}),$$

where

t = current time
W_q = expected wait time for remaining operation q
l_i = number of last operation in job i
p = expected average processing time of waiting operations
k = lookahead parameter.

The expected wait time for operation q is set equal to $b*\overline{\xi_q}$, where b is a parameter of the simulation. In our simulation, $b=2$ and $k=3$. Vepsalainen and Morton (1987) suggest $k=3$ as a reasonable value for dynamic job shops.

Here, the expected processing time of an operation was used in calculating its ATC. Initially, the ATC of each operation without a predecessor was calculated. The one with maximum ATC was loaded on machine M. This process continued by selecting the operation with maximum ATC from the set of operations whose predecessors had been scheduled, until all operations completed processing.

The job shop selected for study is part of an automobile manufacturing facility consisting of 14 machine groups and 26 machines. Ten machine groups contain a single machine, two groups consist of three machines, one machine group contains four machines and the last machine group includes six machines. The machines are all subject to random breakdowns, with a mean time of 2400 minutes between breakdowns. The time to repair each machine is 60 minutes. Means and variances for operation processing times are calculated from this machine information. Fifteen jobs require scheduling on these machines. In total, ninety-three operations must be scheduled. The due dates for the jobs are based on total work content.

The ATC dispatching rule and the two versions of the heuristic were simulated under four parameter combinations of due date tightness and variance in the operation processing times. The due dates were set to 2 or 2.21 times the expected processing times. When the due dates were set to 2 times the expected processing times, there were no replications with zero tardiness. When the due dates were set to 2.21 times the expected processing times, the ATC dispatching rule produced schedules with zero tardiness in two thirds of the replications. Two variance levels were tested in the simulation. In one case, the variance in the processing times was calculated assuming 2400 minutes between breakdowns and 60 minute repair times. In the other case, the variance in the processing times was calculated with double or triple the repair times.

3.2 Simulation Results

The simulation consisted of three experiments. Each experiment used a different distribution to generate the operation processing times. The three distributions tested were: a lognormal distribution with independent processing times, a bi-modal distribution with independent processing times, and a bi-modal distribution with totally correlated processing times.

The results of the simulation study using lognormal operation processing times appear in Table 1 .The ATC dispatching rule is identified by ATC in the table. In Table 1, BC,c=x refers to the version

Parameter Combination	Scheduling Procedure	Median Tardiness	Mean Tardiness	Confidence Level	# Better Than BC	Tardiness Range
tight due date low variance	BC,c=2	27.277	200.26	99	-	22-2256
	WC, c=2	54.485	222.26	50	1	22-2261
	WC, c=0	54.485	354.12	60	2	22-4256
	WC, c=10	57.510	353.94	50	1	50-4109
	ATC	76.291	213.86	-	1	23-1861
tight due date high variance	BC, c=2	25.971	412.71	99	-	21-5346
	WC, c=2	54.893	443.15	50	0	24-5351
	ATC	78.173	367.83	-	1	51-4021
loose due date low variance	BC, c=2	0.000	129.1	99	-	0-1729
	WC, c=2	8.818	136.03	60	0	0-1730
	ATC	16.011	110.06	-	1	0-1285
loose due date high variance	BC, c=2	0.000	339.37	99	-	0-4818
	WC, c=2	7.897	348.2	50	0	0-4820
	ATC	14.030	261.19	-	1	0-3446

Table 1: Simulation Results with Lognormal Processing Times

of the heuristic which chooses the schedule with the minimum best-case expected tardiness at each decision point. Replacements are chosen from the interval of length equal to the expected processing time plus c times the standard deviation of operation I. WC,c=x refers to the version of the heuristic which chooses the schedule with the minimum worst-case tardiness at each decision point. Table 1 lists the median and mean sample tardiness for each scheduling procedure under each parameter combination. Under each parameter combination, the scheduling procedures are entered in order of median tardiness. The table also lists the confidence level that the probability of a negative difference between the tardiness of schedules created by adjacent scheduling procedures is greater than the probability of a positive difference. It includes, for each parameter combination, the number of replications (out of 15 total for each combination) in which a schedule created by the WC,*c=x* or ATC procedures has lower tardiness than the corresponding schedule created by the BC,*c=2* algorithm. The final column in Table 1 summarizes the tardiness range, rounded to the nearest integer, of the schedules obtained under each parameter combination.

Initially, using tight due dates and low variance, the WC version of the heuristic was run with different *c* values to judge the effect of *c* on the results. Table 1 shows that the median and mean tardiness values were similar under the various choices of *c*. Setting *c=10* gives the worst results for the WC algorithm. Increasing the number of potential replacements, while allowing additional flexibility in the algorithm, does not necessarily produce schedules with lower tardiness, because inaccurate future information is being used in decision making. Since the version of the algorithm with *c=2* performed best in these tests, the remainder of the simulation runs were completed with *c=2*.

Note that ranking the scheduling algorithms by sample mean expected tardiness produces a different order. This is primarily because of an extremely large expected tardiness value, an order of magnitude greater than any other, on one of the replications. The ATC dispatching rule always produced the schedule with the lowest tardiness on this case. Removing this replication, the order of the means is the same as the order of the medians.

Altering the due date tightness and variance levels does not influence the ranking of the scheduling procedures. Under every parameter combination with lognormal processing times, the BC version of the heuristic produces schedules with the lowest median expected tardiness. In every case, there is 99% confidence that the BC algorithm is more likely to produce lower expected tardiness schedules than the other algorithms. Comparing the WC,*c=2* and ATC procedures, the highest confidence that the probability of a negative difference in tardiness is greater than the probability of a positive difference in tardiness occurs when the due dates are tight and the variance is low.

In Maddox (1988), Chapter 4, for two test problems, the simple recourse schedule with minimum best-case expected tardiness was tested against the simple recourse schedule with minimum worst-case tardiness. There, when a lognormal distribution was used to generate the processing times, it was concluded that the worst-case schedule performed well only in extreme situations. In light of this, it is not surprising that the BC version of the heuristic performs better in these tests, in terms of average tardiness, than the WC version of the heuristic. Given the small number of replications in the current study, statistics on the behavior of the algorithms in extreme situations is inappropriate. The cost of greatly increasing the number of replications in this study was prohibitive. We conclude, however, that the behavior of the BC and WC heuristic algorithms is similar to the behavior of the optimal simple recourse schedules.

The use of lognormal processing times is not likely to be representative of a shop in which machines generally perform in a fixed amount of time unless a tool breaks or another disruption occurs. In this case, the distribution of each operation processing time is more likely to follow a bi-modal distribution. The distribution achieving the worst-case expected tardiness is also a bi-modal distribution (see Birge and Maddox, 1992). For this reason, bi-modal distributions were also tested. The first set of results considers only independent processing times. This represents the situation in which processing is only loosely connected among machines. It is not the worst-case distribution because the worst-case distribution has many correlated processing times. The results of this experiment appear in Table 2. In

Parameter Combination	Scheduling Procedure	Median Tardiness	Mean Tardiness	# Best (of 30)	Tardiness Range
tight due date low variance	BC, c=2 WC, c=2 ATC	120 60 120	168.4 85.2 118.1	0.5 24 4.5	30-240 120-540 30-240
tight due date high variance	BC, c=2 WC, c=2 ATC	330 60 120	479.1 301.0 225.5	0.5 17.5 11.5	120-2190 30-1080 30-750
loose due date low variance	BC, c=2 WC, c=2 ATC	0 18 0	18.8 33.9 10.7	12 0 18	0-216 0-234 0-156
loose due date high variance	BC, c=2 WC, c=2 ATC	0 18 0	168.2 159.7 92.2	10.83 1.33 17.83	0-1215 0-1146 0-474

Table 2: Simulation Results for Bi-Modal, Independent Processing Times

these results, many of the values were the same because the set of possible processing times was limited. Instead of calculating confidence levels for achieving lower tardiness, we simply list the number of times out of 30 replications in which each method achieved the lowest overall tardiness. Ties were split equally among the tying methods.

Note the difference in the results of Table 2 compared to Table 1. Here, tight due dates favor the WC rule with ATC again producing lower mean values when variance is high. For looser due dates, ATC is preferred. This observation is consistent with the small problem results in Maddox (1988). Tight due dates tend to favor the forward looking but pessimistic WC rule over the dynamic ATC rule and the optimistic BC rule. This characteristic appears even when the processing times are independent.

The next set of results considers completely correlated processing times. This situation may occur, for example, when a batch of parts or its pallet has a defect that delays all processes. It is also representative of a shop of tightly coupled machines, where a breakdown in one area quickly shuts down other areas of the shop. Completely correlated processing times are not necessarily the worst-case distribution because that distribution may impose anti-correlations in order to achieve higher overall expectations. It is probably the worst-case in most practical situations.

The results for this experiment appear in Table 3. Here, there are three possible scenarios

Parameter Combination	Scheduling Procedure	Median Tardiness	Mean Tardiness	# Best (of 3)	Tardiness Range
tight due date low variance	BC, c=2	120	2804	0	120-5469
	WC, c=2	60	2503	3	60-4930
	ATC	120	2673	0	120-5046
tight due date high variance	BC, c=2	120	12139	0	120-22371
	WC, c=2	60	11360	3	60-20784
	ATC	120	11491	0	120-20877
loose due date low variance	BC, c=2	0	2399	1.5	0-5232
	WC, c=2	18	2094	0	0-4269
	ATC	0	2128	1.5	0-4179
loose due date high variance	BC, c=2	0	11840	0.5	0-23091
	WC, c=2	18	10847	1.0	0-20244
	ATC	0	10872	1.5	0-20007

Table 3: Simulation Results for Bi-Modal, Correlated Processing Times

(corresponding to no disruptions, only long jobs disrupted and all jobs disrupted). In this case, the WC heuristic obtains minimum expected tardiness when due dates are tight. ATC performs well in the worst outcomes although in the middle outcomes (only long jobs disrupted) ATC always yielded the highest tardiness.

The computational results demonstrate that the heuristic algorithm with limited updating is a valuable scheduling method in some scenarios. The worst-case algorithm appears preferable when processing times are bi-modal and particularly when processing times are correlated. The best-case algorithm appears best in expectation in the uni-modal cases. The dynamic scheduling approach, as represented by ATC, appears best when processing time variation is high. These results are consistent with tests on small models in Maddox (1988).

Two alternative versions of the heuristic were tested in this simulation. The difference between them was the criterion used to pick the ''best'' schedule to use each time an operation finished processing. Both versions could be improved by resequencing the predecessors of a potential replacement operation in the replacement schedule. In the current versions of the heuristic, the schedule for each potential replacement operation is developed and compared with the current schedule. An alternative is to pick a single replacement, develop the corresponding replacement schedule and compare it with the current schedule. This would decrease the computation time of the heuristic significantly but it could also decrease the performance of the algorithm. The relative value of these changes depends on the specific implementation.

Additional tests that examine the effect of using different objective functions for choosing the current schedule would also be valuable. One possibility is to use the probability that tardiness is greater than some threshold value as the objective. This objective may more accurately reflect the costs of tardiness as premium freight charges are incurred. Initial bounds for this quantity appear in Maddox and Birge (1991).

Acknowledgements

The authors gratefully acknowledge the assistance of Sharon Fox in performing the computational tests.

J. R. Birge was supported in part by Office of Naval Research Grant N00014-86-K-0628 and the National Science Foundation under Grants ECS-8815101 and DDM-9215921.

References

Bean, J.C., and J.R. Birge (1985), "Match-up Real-time Scheduling," Technical Report 85-22, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan.

Bean, J.C., J.R. Birge, C.E. Noon, and J. Mittenthal (1986), "Match-up Scheduling with Multiple Resources, Release Dates and Disruptions," Technical Report, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan.

Birge, J.R. (1985), "Real-time Adaptive Scheduling in Flexible Manufacturing, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan.

Birge, J.R., J.B.G. Frenk, J. Mittenthal, and A.H.G. Rinnooy Kan (1987), "Single Machine Scheduling Subject to Stochastic Breakdowns," working paper.

Birge, J.R., and M.J. Maddox (1992), "Bounds on Expected Project Tardiness," working paper.

Birge, J.R., and R.J-B. Wets (1986), "Designing Approximation Schemes for Stochastic Optimization Problems, in particular, Stochastic Programs with Recourse," *Mathematical Programming Study*, 27, 54-102.

Blau, R.A. (1973), "N-Job, One Machine Sequencing Problems Under Uncertainty," *Management Science*, 20, 1, 101-109.

Conway, R.W., W.L. Maxwell, and L.W. Miller (1967), *Theory of Scheduling*, Addison-Wesley Publishing Company, Inc.

Glazebrook, K.D. (1983), "On Stochastic Scheduling Problems with Due Dates," *International Journal of Systems Science*, 14, 11, 1259-1271.

Glazebrook, K.D. (1984), "Scheduling Stochastic Jobs on a Single Machine Subject to Breakdowns," *Naval Research Logistics Quarterly*, 31, 251-264.

Glazebrook, K.D. (1987a), "Sensitivity Analysis for Stochastic Scheduling Problems," *Mathematics of Operations Research*, 12, 2, 205-223.

Glazebrook, K.D. (1987b), "Evaluating the Effects of Machine Breakdowns in Stochastic Scheduling Problems", *Naval Research Logistics*, 34, 319-335.

Graves, S. C. (1981), "A Review of Production Scheduling," *Operations Research*, 29, 646-675.

Kallberg, J., R. White, and W. Ziemba (1982), "Short Term Financial Planning Under Uncertainty," *Management Science*, 28, 670-682.

Igelmund, G., and F.J. Radermacher (1983a), "Preselective Strategies for the Optimization of Stochastic Project Networks Under Resource Constraints," *Networks*, 13, 1-28.

Igelmund, G., and F.J. Radermacher (1983b), "Algorithmic Approaches to Preselective Strategies for Stochastic Scheduling Problems," *Networks*, 13, 29-48.

Maddox, M.J. (1988), "Scheduling a Stochastic Job Shop to Minimize Tardiness Objectives," Ph.D. Dissertation, The University of Michigan, Ann Arbor, Michigan.

Maddox, M.J., and J.R. Birge (1991), "Bounds on the Distribution of Tardiness in a PERT Network", working paper.

Malcolm, D.G., J.H. Roseboom, C.E. Clark, and W. Fazar (1959), "Application of a Technique for Research and Development Program Evaluation," *Operations Research*, 7, 5, 646-669.

Mittenthal, J. (1986), "Single Machine Scheduling Subject to Random Breakdowns," Ph. D. Dissertation, The University of Michigan, Ann Arbor, Michigan.

Mittenthal, J., and J.R. Birge (1987), "Recourse Bounds in Stochastic Single Machine Scheduling," Technical Report #37-87-133, Decision Sciences and Engineering Systems Department, Rensselaer Polytechnic Institute.

Möhring, R.H., F.J. Radermacher, and G. Weiss (1984), "Stochastic Scheduling Problems I - General Strategies," *Zeitschrift für Operations Research*, 28, 7A, 193-260.

Möhring, R.H., F.J. Radermacher, and G. Weiss (1985), "Stochastic Scheduling Strategies II - Set Strategies," *Zeitschrift für Operations Research*, 29, 3A, 65-104.

Muhlemann, A.P., A.G. Lockett, and C.K. Farn (1982), "Job Shop Scheduling Heuristics and Frequency of Scheduling," *International Journal of Production Research*, 20, 227-241.

Smith, M.L., R. Ramesh, R.A. Dudek, and E.L. Blair (1986), "Characteristics of U.S. Flexible Manufacturing Systems - A Survey," in *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, K.E. Stecke and R. Suri (eds.), Elsevier Science Publishers B.V., Amsterdam, 477-486.

Solel, E. (1986), "A Dynamic Approach to Stochastic Scheduling via Stochastic Control," Ph.D. Dissertation, Dalhousie University, Halifax, Nova Scotia.

Vepsalainen, A.P.J., and T.E. Morton (1987), "Priority Rules for Job Shops with Weighted Tardiness Costs," *Management Science*, 33, 8, 1035-1047.

Weiss, G., and M. Pinedo (1980), "Scheduling Tasks with Exponential Service Times on Nonidentical Processors to Minimize Makespan or Flow Time," *Journal of Applied Probability*, 17, 187-202.

Wets, R. J-B. (1983), "Stochastic Programming: Solution Techniques and Approximation Schemes," in *Mathematical Programming: The State of the Art 1982*, A. Bache, N. Groëche, and L. Krote, (eds.), Springer-Verlag, Berlin, 566-603.