ACCURATE MYOPIC HEURISTICS

FOR TARDINESS SCHEDULING

Working Paper No. 370

Thomas E. Morton*
Ram Mohan Rachamadugu**
Ari Vepsalainen***

February 1984

*Carnegie-Mellon University, Pittsburgh, Pennsylvania.
**The University of Michigan, Ann Arbor, Michigan
***Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania

ACCURATE MYOPIC HEURISTICS FOR TARDINESS SCHEDULING

by

Thomas E. Morton*

Ram Mohan Rachamadugu**

Ari Vepsalainen***

February 1984

*Carnegie-Mellon University, Pittsburgh, Pennsylvania.
**The University of Michigan, Ann Arbor, Michigan.
***Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania.

Abstract

Weighted tardiness is easily one of the top three criteria in scheduling, (one reported survey [7] suggests it is the most important practical criterion) yet little has been done for real world problems of this type, due to its inherent complexity. (In fact, simple, intuitive, extremely accurate heuristics for even the stylized one machine problem have not been previously available.) Good heuristics are thus essential; one major approach is to design excellent heuristics for simplified versions of real problems, and then to modify these to build and test heuristics for more complicated situations. We follow this approach. We have built and tested such a heuristic for the stylized one machine problem. It is one of the simplest ever proposed, has good intuitive appeal, and has been found in testing to be extremely accurate. The heuristic has also been modified and tested for parallel machines, and flow shops. All these results are reported briefly here. A very brief summary of further research for proportionate flow shops, bottleneck shops and job shows, currently underway, is also given, followed by a sketch of future research directions. (This article is a condensation of three technical reports. [18], [22], [25].)

ACCURATE MYOPIC HEURISTICS FOR TARDINESS SCHEDULING

1. Introduction and Summary

In a survey, Panwalker, Dudek, and Smith [7] report that the proportion of schedulers in industry who ranked meeting due dates or minimizing penalty costs as the most important criterion was larger than for any other criterion. The class of such criteria would include mainly average tardiness, weighted average tardiness, maximum tardiness and weighted percentage tardy. It can be argued that weighted average tardiness is the most important of these. (In any event the performance of our heuristics developed by us seems to be highly correlated for these three, as will be apparent below in the flow shop results.) Thus the design of good heuristics for real world problems with a weighted tardiness criterion would seem to be very important, especially if these heuristics also turn out to be robust. Yet relatively little has been done for tardiness in realistic environments since Carroll introduced COVERT in 1965 [4].

One major approach to developing good heuristics for complex environments is to design excellent heuristics for the stylized one machine problem, and then to modify these to build and test heuristics for more realistic situations. We follow this approach. (Other major approaches such as search techniques or relaxation procedures can still often benefit by the addition of simple heuristics for guidance.)

Even for the stylized one machine problem, simple, intuitive, extremely accurate heuristics have not been available. In Sections 2 and 3 we discuss 5 earlier heuristics for the simple problem: WSPT, EDD, Schild and Fredman [24], Montagne [15], Baker [2]. Next we present a linear and an exponential version of a new myopic heuristic which behaves asymptotically correctly

when the due dates are either too tight or too slack and which also approximates the choices of pairwise interchange. (Preliminary study resulted in the selection of the exponential version.) The Schild and Fredman [24] procedure could not be evaluated since it was devised to be an exact procedure, which was later shown to be wrong. Baker's procedure [2] cannot easily be generalized to the weighted case, and thus was also not evaluated. WSPT, EDD, Montagne, and the new MYOPIC were tested against optimal solutions when they could be found, (lower and upper bounds otherwise), for 1,920 problems. The myopic procedure consistently outperformed the other heuristics in an average sense by a wide margin. There was only one setting of problem parameters for which the myopic solution exceeded the optimal solution by more than 0.07 of an average processing time per job.

We turn now to the parallel machine extension presented in Section 4. There do not seem to be any previous heuristics here, even when all machines are identical. However, we can generalize our heuristics and the earlier one machine heuristics to the equal parallel machine case in a rather obvious way. All the simple heuristics provide a dynamic ranking of the jobs. When a machine becomes available, assign the highest ranking job. Due to greater computational complexity, we did not attempt to compute optimal solutions for our benchmark. (A new approach allows some optimal testing for smaller problems; such further testing is currently in progress.) We tested EDD, WSPT, Montagne, Two-pass Montagne, and the Myopic heuristic on 1,280 problems. In general, the Myopic outperformed all others in an average sense with only a few exceptions. In these exceptions, the Myopic cost never exceeded the best cost by more than 0.002; also its parameter value could be adjusted slightly to obtain the best result. In order to provide more conservative benchmarks, a second study was done in which lower bounds were calculated for each problem

by several methods. Average myopic deviations from the "best" lower bound were 0.01, 0.08, 0.16, 0.21 by increasing tardiness factor, for a total of 320 problems. The myopic procedure, in a third set of experiments was followed by pairwise interchange to try to get a different type of benchmark. Not much improvement to the heuristic resulted; as expected from its design, the Myopic heuristic is usually locally optimal or near optimal.

We turn next to flow shops in Sections 5 and 6. It turns out that the Myopic heuristic can usually be applied in a local dispatch fashion with good results. This is consistent with well known results for average lateness [1]. However when a heuristic employs the due date, this unfortunately means that the due date off the current machine (operation due date) must be determined. This requires knowing total processing time and waiting time after this machine for each job (we term these "lead times"). A theorem for preemptive proportionate shops is that if lead times are known exactly, solving the problem locally at each machine will exactly solve the global problem. (A flow shop is said to be proportionate if the ratio of processing time for any 2 jobs is constant across machines.) Thus if shops in practice are "nearly" proportionate, accurate estimation of lead times is central. Lead times in the study were estimated by two methods: 1. A fixed multiple of remaining process time. (The constant was a parameter to be investigated) 2. Iterative estimation of actual lead times. Neither optimums nor lower bounds could be obtained. Seven competing heuristics were evaluated: 1. First Come First Served (FCFS), 2. Weighted Shorted Process Time (WSPT), 3. Earliest (Global) Due Date (EDD), 4. Slack per Remaining Operation (S/OP), 5. COVERT, 6. MYOPIC1, 7. MYOPIC2. A total of 1,280 problems were run. For each problem the standard of comparison was excess above the "best result of the 7 heuristics." To summarize the results, MYOPIC came in first

and second, COVERT was third, WSPT a poor fourth: MYOPIC2 = 0.009; MYOPIC1 = 0.04; COVERT = 0.09; WSPT = 0.37. FCFS was last with 1.36. Next, a second study was run with the same set of problems to check out robustness of the various heuristics to misspecification of the criterion function. The criteria evaluated were weighted tardiness, maximum tardiness, % tardy jobs, work in process (WIP), work in system (WIS). (The first 3 are tardiness measures, the last two inventory measures.) The three tardiness measures behaved very similarly, giving a combined ranking order of MYOPIC2, MYOPIC1, COVERT, WSPT, EDD, S/OP, FCFS (almost identical to the first study). The two shop inventory measures differed from the tardiness rankings (and from each other), with an average ranking of EDD, S/OP, MYOPIC2, MYOPIC1, WSPT, COVERT, FCFS. (The poor showing of WSPT is an average of a very good showing on WIP, and a very poor showing on WIS). Note that MYOPIC2 ranks first and third across tardiness and inventory measures, a truly remarkable achievement. MYOPIC1, which is easier to use, ranks second and fourth. COVERT ranks third and fifth, WSPT ranks fourth and fifth.

In Section 7 we very briefly indicate current other related research on the tardiness problem. For equal parallel machines, the authors have recently discovered a method for determining optimal solutions whose effort is the product of the number of machines, and the effort for one machine with the same total number of jobs. Thus it now becomes feasible to run some smaller problems with the optimum as a benchmark. A proportionate flow shop is one in which any two jobs have the same ratio of processing time on every machine. A number of simple propositions have been proved for the pre-emptive case. These results should help construct good heuristics for the non-preemptive case if there is only "mild" departure from proportionality [25]. Peng Si Ow [20] has recently achieved a heuristic for flow shops with a <u>bottleneck</u>

machine which are better than any of MYOPIC1, MYOPIC2, or COVERT in this situation. Vepsalainen has had good success to date extending the myopic to dynamic job shops. Again MYOPIC2 is usually superior. He has also tested a number of improvements such as anticipated work in queue (AWINQ) modifications.

We finally discuss possible future research in Section 8. Some of these directions include:

1. Add dynamic arrivals and other time constraints to the one machine problem.
2. Add precedence constraints to the one machine problem.
3. Treat uniform (proportionate) parallel processors.
4. Treat general unequal parallel processors.
5. Treat the generalized flow shop.
6. Treat bottleneck job shops.
7. Develop myopic procedures for other criteria.

Plans for each of these possibilities will be briefly sketched.

2. One machine--Heuristic Design

The next two sections represent a summarized version of the research reported in [18]. Lenstra [14] has shown that the simple weighted tardiness problem is NP-complete, which motivates the heuristic approach taken here. Problem definition: n jobs arrive simultaneously with processing time p_i, due date d_i, and weight w_i. We minimize the sum of penalties for jobs. If t_i is the completion time of i, its penalty is $w_i(t_i - d_i)^+$. Turning to past heuristics, the weighted shortest processing time rule (WSPT) is known [2] to be asymptotically optimal as the load increases, and to minimize weighted lateness. The earliest due date (EDD) rule will find, for lightly loaded

shops, a perfect zero tardiness schedule if one is possible [2]. Schild and Fredman combined these two rules to produce a procedure [24] known not to be optimal [8]. There seems to be no computational studies. Montagne's method [15] is to rank jobs by decreasing $w_i/(p_i(\sum p_j - d_i))$, that is, the WSPT measure multiplied by the "tardiness if last." Although this method turned out to test here fairly well, it is not clear how to generalize "last of all" in a dynamic long horizon context. Baker [10] presents a good heuristic with the right asymptotic behavior for the unweighted case. It is appears to be difficult to generalize to the weighted case; we did not test it.

In developing our new myopic heuristic, we desired one with the correct asymptotic behaviors, plus local optimality under adjacent pairwise interchange.

Proposition: For two adjacent jobs in an optimal solution

$$\frac{w_i}{p_i}\left\{1 - \frac{(d_i - t - p_i)+}{p_j}\right\} \geq \frac{w_j}{p_j}\left\{1 - \frac{(d_j - t - p_j)+}{p_i}\right\}$$

where t is the start time for job i.

To form a heuristic, we first replace p_i and p_j by the average $\bar{p}$. (This approximation allows each job to be given a priority in isolation rather than investigating all pairs, reducing effort from $O(n^2)$ to $O(n)$, at some cost of accuracy.) Secondly we allow a family of rules by replacing $\bar{p}$ by $k\bar{p}$, and determining the best k experimentally. For a larger k than the nominal k=1, the priority begins to rise somewhat before it would be necessary for local optimality. It turns out that this look ahead feature helps to approximate 3 and 4 way conflicts. A k=2.0 usually turns out to be about the best. Thus our priority rule becomes

$$R_i = \frac{w_i}{P_i} \left\{1 - \frac{(d_i - t - P_s)^+}{k\bar{p}}\right\}^+$$

The solid heavy line in Figure 1 illustrates the first form of our heuristic priority rule, which we dub "linear" for the way the priority falls off.

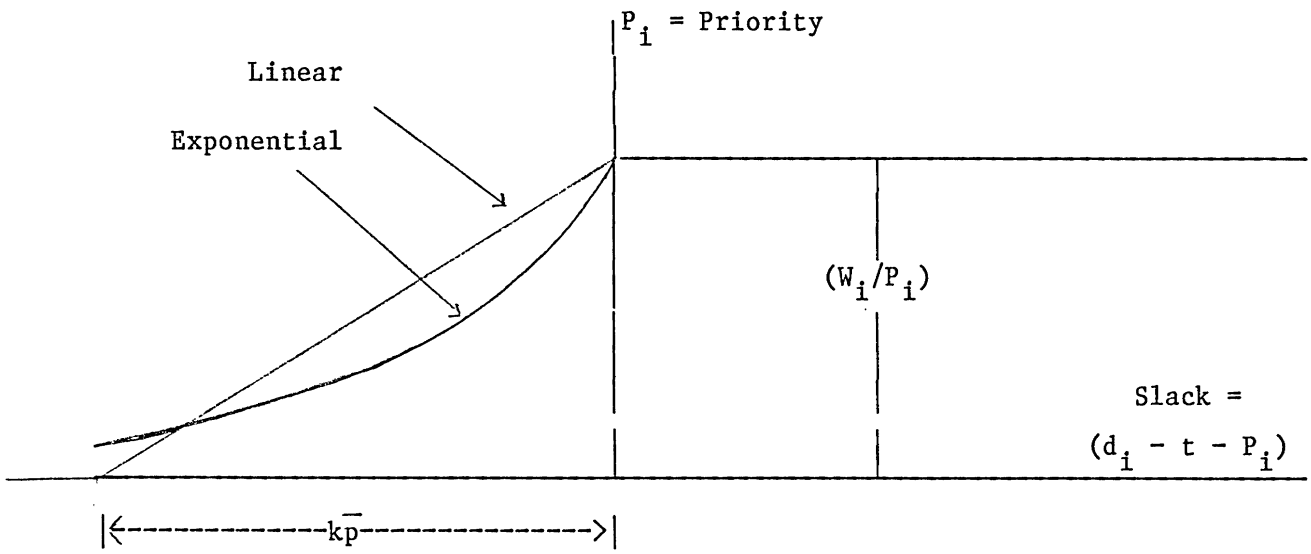

FIGURE 1

Myopic Heuristics

After some initial testing it became apparent that simply assigning 0 priority to all jobs with more than $2\bar{p}$ slack is not good. Jobs should retain some priority based both on due date and weighted processing time. An exponential decay proved best (typical k=0.5).

$$R_i = \frac{w_i}{p_i} \exp\left\{-\frac{k}{\bar{p}}(d_i - t - p_i)^+\right\}$$

The rule is the same for negative slack, but dies off like the dashed line for positive slack. The rule now becomes in this region a due date rule, except that due dates are shifted somewhat forward or backward depending on the WSPT ranking values. We dub this the "exponential" version.

3. One machine--Computational Study.

The original study [18] devotes several pages to previous researchers' experimental design to help motivate ours. However, this discussion is omitted here. Our own measure of the performance of heuristic j is $C_j = (v_j - v^*)/(\bar{w}^* n^* p)$. That is, the normalized excess cost of heuristic j is the excess of its cost over optimal divided by the product of average job weight, number of jobs, and average processing times. This measures gives the average extra tardiness per job for a heuristic as a fraction of a standard processing time. (Percentage errors are very misleading for answers all very close to zero.)

We omit also any discussion here of the carefully designed procedures for obtaining optimal benchmarks where possible, and lower/upper bounds otherwise. (Basically optimal benchmarks for n=10 and n=20 problems, and about 50/50 between optimal and bounds for n=30.) Turning now to the experimental design, some parameters need explanation. The tardiness factor is a measure of % of jobs which on average would be tardy in a FCFS sequence. The due date range expresses "maximum" range in due dates as a fraction of $n\bar{p}$.

We did a factorial design with:

4 tardiness factors:	0.2, 0.4, 0.6, 0.8
2 job time coef. of variation:	0.1, 0.3
1 uniform distribution for w_i/p_i	[0,2]
2 correlations between p_i and d_i	0.0, 0.5
2 due date range factors	0.4, 0.8
3 numbers of jobs	10, 20, 30
20 replications per cell	

leading to 1,920 problems. From prior testing, we selected the exponential version of our myopic heuristic with k=0.5.

Table 1 presents a summary of our results for n=10 and n=20. The case n=30 is more complicated, since only half of the problems could be solved to optimality. The optimal solution comparisons for n=30 are presented in Table 2, while comparison with best lower bounds obtained for the other problems are shown in Table 3. For the same unsolved problems Table 4 compares the heuristic vs. best upper and lower bounds.

In Tables 1, 2, and 3 we see that the myopic heuristic is consistently close to optimal, and superior to other heuristics. It is also extremely consistent over parameter variations. In only one parameter setting did the myopic solution exceed the optimal solution by more than 0.07 of an average processing time. In only one setting was the myopic second best, and then by only 0.006. Table 3 shows that for the unsolved n=30 problems, the myopic outperforms the second best heuristic (Montagne) by a wide margin. Table 4 shows that it is very difficult to improve on the myopic heuristic in these cases, even with extensive computational effort.

What can be said about the other heuristics? EDD is good at a tardiness of 0.2 but deteriorates very rapidly at high tardiness to be <u>by far</u> the worst. WSPT is slightly worse than EDD at low tardiness, but is much better at high tardiness. The Montagne procedure and the Myopic procedure dominate these at all parameter settings. The Montagne procedure is nearly competitive with the Myopic for n=10, but becomes uncompetitive rapidly for higher numbers of jobs.

In summary, it is rather clear from our computational study that the new myopic procedure performs close to optimally throughout a range of parameters, and also outperforms other heuristics, by large margins in many cases. It is simple, easy to implement, and can be used in a dynamic dispatching mode. We shall see that it generalizes readily to more realistic problems.

TABLE 1

ONE MACHINE

Mean Performance for 10 and 20 Job Problems

# Jobs	Tardiness	R = 0.4					R = 0.8				
		OPT Value	EDD Excess	WSPT Excess	MP Excess	MYO Excess	OPT Value	EDD Excess	WSPT Excess	MP Excess	MYO Excess
10	0.2	0.038	0.770	0.047	0.014	0.020*	0.017	0.028	0.107	0.011	0.026
	0.4	0.253	0.515	0.094	0.048	0.029	0.184	0.293	0.232	0.084	0.051
	0.6	0.886	1.006	0.151	0.088	0.027	0.740	0.953	0.376	0.183	0.055
	0.8	2.090	1.427	0.112	0.065	0.015	2.094	1.402	0.202	0.088	0.025
20	0.2	0.024	0.107	0.074	0.027	0.021	0.007	0.024	0.151	0.022	0.014
	0.4	0.403	0.830	0.192	0.125	0.033	0.196	0.513	0.498	0.202	0.047
	0.6	1.319	1.981	0.298	0.194	0.035	1.128	1.697	0.774	0.398	0.054
	0.8**	3.619	2.897	0.337	0.219	0.018	3.513	3.064	0.587	0.220	0.018

OPT: Mean Value of Normalized Optimum
EDD: Earliest Due Date Rule
WSPT: Weighted Shortest Processing Time Rule
MP: Montagne's Procedure
MYO: Myopic Heuristic

*Increasing the value of parameter k in the myopic heuristic yields better results than Montagne's method.

**One problem was not solved to optimality in case of both range factors - 0.4 and 0.8. However, myopic heuristic yielded the best feasible solution for both problems.

TABLE 2

ONE MACHINE

Mean Performance Measure for Solved 30 Job Problems

(n = 30)

R = 0.4

Tardiness	# Fully Solved	OPT Value	EDD Excess	WSPT Excess	MP Excess	MYO Excess
0.2	73	0.027	0.107	0.099	0.035	0.017
0.4	26	0.400	1.125	0.290	0.164	0.027
0.6	8	2.069	2.049	0.439	0.350	0.056
0.8	16	5.186	4.242	0.564	0.315	0.018

R = 0.8

Tardiness	# Fully Solved	OPT Value	EDD Excess	WSPT Excess	MP Excess	MYO Excess
0.2	80	0.001	0.033	0.224	0.020	0.007
0.4	38	0.172	0.521	0.739	0.260	0.048
0.6	10	1.600	2.412	1.215	0.634	0.073
0.8	20	5.380	4.223	0.837	0.352	0.030

TABLE 3

ONE MACHINE

Deviations from the Best Lower Bound
(For Unsolved 30 Job Problems)

Tardiness	Number of Problems	R = 0.4				R = 0.8				
		EDD Excess	WSPT Excess	MP Excess	MYO Excess	Number of Problems	EDD Excess	WSPT Excess	MP Excess	MYO Excess
0.2	7	0.413	0.196	0.150	0.091	0	-	-	-	-
0.4	54	1.457	0.514	0.414	0.250	42	1.157	0.853	0.481	0.209
0.6	72	3.586	1.140	0.940	0.713	70	3.165	1.793	1.188	0.664
0.8	64	5.098	1.056	0.876	0.592	60	4.799	1.452	0.923	0.487

TABLE 4

ONE MACHINE

Comparison of Heuristics and Lower and Upper Bounds
(For Unsolved 30 Job Problems)

Tardiness	R = 0.4			R = 0.8		
	Best Lower Bound	Best Upper Bound	Myopic Value	Best Lower Bound	Best Upper Bound	Myopic Value
0.2	0.071	0.155	0.162	NONE	NONE	NONE
0.4	0.294	0.526	0.544	0.167	0.356	0.376
0.6	1.270	1.909	1.983	0.925	1.522	1.589
0.8	4.549	5.096	5.141	4.504	4.965	4.991

4. Parallel Machines

We modify the problem just given only in that n jobs now arrive simultaneously to m initially available machines. (The modification to allow dynamic arrivals and initial availability is simply the standard dispatch modification.) A much more complete discussion of prior research, lower bound methodology, etc. is given in the original paper [11].

There seem to be no prior study of heuristics for weighted tardiness and parallel machines. A number of related, preemptive, and special case results are given in [13, 10, 9, 19, 23]. One heuristic for the unweighted case is known [3], which unfortunately reduces to EDD for one machine, which as we have seen is very poor.

In extending our heuristic to the parallel machine case, we first note:

Proposition 1: If all jobs have equal length, any two jobs scheduled jth and (j+1)th on two different machines may be pairwise interchanged using the proposition of Section 2.

Proposition 2: A similar remark applies for any two adjacent jobs on the same machine.

These propositions guide our construction of a heuristic. Order the jobs in a list by the exponential version of the one machine myopic heuristic. Assign the first job on the list to the first available machine. Repeat. Note that our heuristic is a single-pass procedure; partitioning by machine and sequencing on machine are simultaneous; it is a dispatch procedure. (Re-sequencing after assignment was found to give only very small improvement, and destroyed the dispatch feature.)

We adapt WSPT, EDD, and Montagne from Section 2 in a similar way to give competing heuristics. We also test a two-pass version of Montagne heuristic: 1. basic heuristic 2. re-sequence by Montagne on each machine.

We turn now to the experimental design, which is very similar to that for the one machine case.

4 tardiness factors:	0.2, 0.4, 0.6, 0.8
2 job time coef. of variation:	0.1, 0.3
1 uniform distribution for w_i/p_i	[0,2]
Correlation between p_i and d_i	0
2 due date range factors	0.4, 0.8
jobs per machine	15, 30
machines	2,5
replications	20

Thus there were a total of 1,280 problems tested. The normalization applied to costs to compare heuristics is the same as given in the last section. Preliminary tests suggested the exponential myopic be used with k=1.0. (We could have used k = 0.5 as in single machine case. However, better results can be achieved by slightly increasing the value of k for larger number of machines.

In Tables 5, 6, 7, and 8 we see that the myopic heuristic is consistently superior to other heuristics. In the only two parameter settings which were exceptions, the myopic heuristic was second by no more than 0.002. The two pass version of Montagne's method was almost always second. Our heuristic scored dramatically better than the second best for large range due dates (R = 0.8) less dramatically better for small ranges (R = 0.4). Our advantage also increases moderately by tardiness factor. It seems relatively insensitive to numbers of jobs and machines. EDD deteriorates very rapidly with increasing tardiness factor, as before; WSPT is superior only to EDD as before.

In the few situations in the above experiment where the procedure did not do as well, it seemed possible to fine tune the parameter. This was done successfully, and is reported in the original paper [22].

TABLE 5

PARALLEL MACHINES
Comparison of Heuristics

Number of machines: 2
Number of jobs per machine: 15

Tardiness \ Due Date Range	0.4					0.8				
	EDD	WSPT	M1	M2	H1	EDD	WSPT	M1	M2	H1
0.2	0.1135	0.0944	0.0874	0.0581	0.0382	0.0163	0.1320	0.1001	0.0230	0.0090
0.4	1.0085	0.4552	0.4364	0.4058	0.3286	0.6017	0.5637	0.4844	0.3511	0.1896
0.6	2.5070	1.2997	1.2724	1.2257	1.1121	2.1958	1.4814	1.3412	1.1669	0.9121
0.8	5.0060	3.0120	2.9728	2.9428	2.8488	5.0261	3.1517	3.0061	2.9084	2.7178

TABLE 6

PARALLEL MACHINES
Number of machines: 2
Number of jobs per machine: 30

Tardiness \ Due Date Range	0.4					0.8				
	EDD	WSPT	M1	M2	H1	EDD	WSPT	M1	M2	H1
0.2	0.1195	0.1245	0.1142	0.0638	0.0306	0.0050	0.2231	0.1737	0.0296	0.0025
0.4	1.7513	0.7901	0.7572	0.6968	0.6432	0.9343	0.8766	0.7581	0.5231	0.2914
0.6	4.9485	2.4212	2.3061	2.2864	2.2345	4.1909	2.6885	2.4477	2.1832	1.7778
0.8	9.4954	5.4868	5.3947	5.3070	5.1162	9.6686	5.9662	5.6993	5.4797	5.0396

EDD Earliest Due Date Rule
WSPT Weighted Shortest Processing Time Rule
M1 Montagne's method
M2 Montagne's method, two-pass version
MY Myopic heusitic
* Increasing the value of the parameter yields better results than the EDD rule

TABLE 7

PARALLEL MACHINES
Comparison of Heuristics

Number of machines: 5
Number of jobs per machine: 15

Tardiness \ Due Date Range	0.4					0.8				
	EDD	WSPT	M1	M2	H1	EDD	WSPT	M1	M2	H1
0.2	0.1297	0.0865	0.0847	0.0565	0.0346	0.0032	0.1080	0.0999	0.0166	0.0051*
0.4	0.9839	0.4413	0.4365	0.3998	0.3089	0.6117	0.5332	0.5100	0.3500	0.1894
0.6	2.5311	1.3220	1.3100	1.2596	1.1152	2.2319	1.4712	1.4244	1.1920	0.8566
0.8	4.9591	2.9784	2.9561	2.9122	2.7647	4.9877	3.1411	3.0750	2.9233	2.6845

TABLE 8

PARALLEL MACHINES
Number of machines: 5
Number of jobs per machine: 30

Tardiness \ Due Date Range	0.4					0.8				
	EDD	WSPT	M1	M2	MY	EDD	WSPT	M1	M2	MY
0.2	0.1300	0.1179	0.1144	0.0681	0.0285	0.0001	0.2032	0.1874	0.0255	0.0011*
0.4	1.7140	0.7327	0.7224	0.6526	0.5507	0.9190	0.9130	0.8709	0.5308	0.2707
0.6	4.8413	2.3901	2.3714	2.2680	2.2111	3.9717	2.6173	2.5265	2.0972	1.6149
0.8	9.5990	5.5407	5.5031	5.3813	5.1435	9.3606	5.7673	5.6383	5.2774	4.7507

EDD Earliest Due Date Rule
WSPT Weighted Shortest Processing Time Rule
M1 Montagne's method
M2 Montagne's method, two-pass version
MY Myopic heusitic
* Increasing the value of the parameter yields better results than the EDD rule

We turn now to a second study, in which we calculated several lower bounds on each problem, and recorded the best one. The myopic heuristic was then compared to the lower bound. Table 9 shows this comparison for 320 problems. Each line of the table represents the average of 20 replications; parameter variations are shown in the table. The myopic scores very well. Average deviations from the best lower bound were 0.01, 0.08, 0.16, 0.21 by increasing tardiness factor. Since no high-computational upper bounds have been obtained which are not much closer to the heuristic than to lower bounds, these results are very satisfying. In Table 10 we turn to that issue. For the same 320 problems a phase 2 pairwise interchange was added to the myopic. Note that there was very little improvement.

5. Flow Shops--Heuristic Design

We omit details of earlier studies, details of the experimental design, and fine detail of analysis; these may be found in [4]. It turns out that the myopic heuristic can be applied in a local dispatch fashion with good results, both here in flowshops, and in current research by Vepsalainen in job shops. This is consistent with many early studies validating the local use of WSPT for weighted average lateness [11, 21]. However, in the current case, the operation due date is needed off the local machine. This is the global due date minus the lead time after this machine, defined as the actual sum of remaining processing time and waiting. Thus accurate estimation of the lead-time becomes essential. We first summarize some results for the preemptive proportionate flow shop which support this idea.

Proposition 1. If all the LT are known exactly for the optimal solution, then optimal local assignments will also optimize the global problem.

Proposition 2. Proposition 1 is relatively insensitive to small errors in the leadtime.

TABLE 9

PARALLEL MACHINES

Deviation from the Best Lower Bound
the Myopic Heuristic

Tardiness Factor = 0.2

		R = 0.4		R = 0.8	
n	n/m	Deviation from lower bound	Best lower bound	Deviation from lower bound	Best lower bound
2	15	0.0119	0.0113	0.0069	0.0029
2	20	0.0225	0.0229	0.0051	0.0009
4	10	0.0184	0.0187	0.0128	0.0029
4	14	0.0231	0.0138	0.0063	0.0000

Tardiness Factor = 0.4

2	15	0.0493	0.2603	0.0637	0.1621
2	20	0.1038	0.2742	0.1096	0.1514
4	10	0.0416	0.1660	0.0582	0.0755
4	14	0.0679	0.1758	0.0662	0.0690

Tardiness Factor = 0.6

2	15	0.1012	0.9009	0.1263	0.8226
2	20	0.2812	1.2383	0.2363	0.9586
4	10	0.0933	0.7628	0.1007	0.5444
4	14	0.1616	0.8224	0.1636	0.6652

Tardiness Factor = 0.8

2	15	0.2083	2.7561	0.1867	2.5254
2	20	0.3547	3.2259	0.2576	3.0736*
4	10	0.1053	1.8846	0.1394	1.8294*
4	14	0.2440	2.2868	0.1668	2.3890*

m = machines n = jobs

TABLE 10

PARALLEL MACHINES

Deviation of the Myopic Heuristic from the Best Lower Bound after Adjacent Pairwise Interchange

Tardiness Factor = 0.2

		R = 0.4		R = 0.8	
n	n/m	Deviation from lower bound	Best lower bound	Deviation from lower bound	Best lower bound
2	15	0.0084	0.0113	0.0053	0.0029
2	20	0.0204	0.0229	0.0035	0.0009
4	10	0.0158	0.0187	0.0179	0.0029
4	14	0.0200	0.0138	0.0038	0.0000

Tardiness Factor = 0.4

2	15	0.0457	0.2603	0.0600	0.1621
2	20	0.1007	0.2742	0.1025	0.1514
4	10	0.0392	0.1660	0.0480	0.0755
4	14	0.0621	0.1758	0.0555	0.0690

Tardiness Factor = 0.6

2	15	0.0995	0.9009	0.1217	0.8226
2	20	0.2808	1.2383	0.2328	0.9586
4	10	0.0925	0.7628	0.0962	0.5444
4	14	0.1601	0.8224	0.1613	0.6652

Tardiness Factor = 0.8

2	15	0.2079	2.7561	0.1867	2.5254
2	20	0.3530	3.2259	0.2566	3.0736*
4	10	0.1052	1.8846	0.1383	1.8294*
4	14	0.2435	2.2868	0.1658	2.3890*

Lead times in this study were estimated by two methods: 1. A fixed multiple of remaining process time. (The constant is a free parameter). 2. Iterative estimation of leadtime. In the iterative approach, the problem is first run with fixed multiple estimates. All the actual resulting leadtimes are saved and used in a second pass as the estimates. The process is repeated (5 or 6 times usually) until there is no further improvement (there will still be some "bounce" at termination).

The original report discusses a number of related previous heuristics using the due date: 1. global due date 2. operation due date (allocate 1/m of remaining time to each machine to set these) 3. static slack, 4 static slack per operation, 5. static slack/remaining operation time. Dynamic versions of these rules are also discussed. Most of these rules do poorly in the weighted tardiness case. Some of the more elaborate historical procedures do use some estimate of lead times. These include: 1. critical ratio, 2. dynamic composite, 3. COVERT. COVERT seems to perform well [22]. good results for unweighted average tardiness. (We made the obvious modification to COVERT for the weighted case to make our comparison fair.) COVERT is <u>somewhat</u> similar to the linear version of our MYOPIC heuristic, but it is not designed to be locally optimal as ours is. COVERT has maximum priority when the "crash" slack is zero, decreasing down to zero priority when a "generous" slack is greater or equal to zero. MYOPIC (linear) has maximum priority when the <u>true</u> slack is zero, decreasing to zero priority about two processing times prior to due date. COVERT has also a free parameter in defining "generous," which we set at 0.5 as recommended [11].

MYOPIC1 and MYOPIC2 then implement the exponential rule (k=0.5) developed for the one machine case using the operation due date, fixed estimate, iterated estimate, respectively in a dispatch mode. In the current

job shop research, Vepsalainen starts with a similar procedure tested in a typical set of dynamic job shops.

6. Flow Shops--Computational Study

We did not attempt to compute optimal solutions, or even lower bounds, given the computational complexity. The stopping rule for the MYOPIC iterative procedure was based on: 1. convergence of leadtime estimates; 2. improvement in objective function; 3. number of iterations. We tested the following heuristics: FCFS, WSPT (local), EDD (global), S/OP, (weighted) COVERT, MYOPIC1, MYOPIC2.

The layouts investigated are as follows. A 4 machine shop could be increasing speed from first to last, constant, or decreasing. Either 20 or 60 jobs were used giving 6 cases. An 8 machine shop could be increasing or alternating with 60 jobs giving 2 cases. Thus there were a total of 8 layouts.

Processing times for a job are a random multiple of a base time for that job divided by the speed of the machine. The multiple was either uniform (0.75, 1.25) or uniform (0.25, 1.75). (Smaller range is approximately proportionate shop, larger range is approximately random.) The base for each job was drawn also from a uniform distribution (5, 25). There are two side load types. Either all jobs start at the first machine, or else 20% start at later machines for 4 machine cases, 10% for 8 machine cases. The tardiness factor is set at 0.3 or 0.6. The due date range was 0.6 to 1.6. Due dates are set depending on the tardiness factor, as explained in the report on single machine case [2]. Weights for jobs are uniform from 1.0 to twice the base job time, making WSPT indices flatter, and thus making problems more difficult. There are 10 replications per parameter set with a factorial design, giving a total

of 1,280 problems run. We use the same normalized measure of excess weighted tardiness discussed earlier. Since optimal solutions are not available (or lower bounds), we use for each problem the excess of a given rule over the best of the seven heuristics.

Table 11 gives these results. To summarize rankings by overall average normal excess over the "best of 7": MYOPIC2 = 0.009; MYOPIC1 = 0.04; COVERT = 0.09; WSPT = 0.37; EDD = 0.61; S/OP = 0.62; FCFS = 1.36. Thus the myopic with iterated lead time is superior (but somewhat expensive). One can use the cheaper constant MYOPIC1 at a normalized cost of 0.03 COVERT (just as complicated as MYOPIC1) costs an additional 0.05. At 0.28 more, the additional simplicity of WSPT seems perhaps bought at a high price. The other three are not really in the running. Careful investigation of the table reveals that the qualitative behavior of the four best heuristics vis-a-vis each other remains remarkably constant over layout type, tardiness factor, due date range. Behavior of all the rules by parameter variations are very consistent with results for the one machine case. Sensitivity of results to experimental design is fully discussed in the original paper.

We turn to a second study which investigated robustness of these results to misspecification of the criterion. The same set of problems and heuristics were tested against three tardiness measures and two inventory measures. The results are shown in Table 12. Overall rankings are very similar for weighted tardiness, maximum tardiness, and % tardy, with average ranking of MYOPIC2, MYOPIC1, COVERT, WSPT, EDD, S/OP, FCFS. The two inventory measures, work in process (WIP) and work in system (WIS) were different in ranking from tardiness measures, and fairly different from each other. (WIS charges inventory at least until the due date, and thus does not particularly reward earliness as does WIP). The ranking for WIP is WSPT, (MYOPIC1, MYOPIC2, EDD) nearly

FLOW SHOPS

Table 11. Comparison of Heuristics

Rule	4 machines, 20 jobs				Total Average	Excess over Best Rule
	τ = 0.3		τ = 0.6			
	R = 0.6	R = 1.6	R = 0.6	R = 1.6		
FCFS	.836	1.269	2.264	2.395	1.691	1.267
WSPT	.261	.625	1.017	1.242	.786	.362
EDD	.322	.054	1.904	1.661	.985	.561
S/PT	.361	.075	1.921	1.658	1.004	.580
COVERT	.163	.071	1.002	.784	.505	.082
MYOPIC1	.121	.033	1.004	.738	.474	.051
MYOPIC2	.104	.021	.912	.679	.431	.007

Rule	4 machines, 60 jobs				Total Average	Excess over Best Rule
	τ = 0.3		τ = 0.6			
	R = 0.6	R = 1.6	R = 0.6	R = 1.6		
FCFS	1.171	1.970	3.547	3.997	2.672	2.065
WSPT	.335	.850	1.565	1.919	1.167	.560
EDD	.349	.026	3.053	2.571	1.500	.893
S/PT	.376	.039	2.948	2.472	1.459	.852
COVERT	.198	.043	1.513	1.069	.706	.099
MYOPIC1	.132	.014	1.455	1.015	.654	.047
MYOPIC2	.112	.010	1.383	.971	.619	.012

Rule	8 machines, 60 jobs				Total Average	Excess over Best Rule
	τ = 0.3		τ = 0.6			
	R = 0.6	R = 1.6	R = 0.6	R = 1.6		
FCFS	.601	.753	1.455	1.588	1.129	.746
WSPT	.254	.441	.707	.883	.571	.188
EDD	.368	.218	1.251	1.240	.769	.386
S/PT	.437	.264	1.265	1.253	.805	.422
COVERT	.195	.210	.706	.744	.464	.081
MYOPIC1	.170	.166	.698	.714	.437	.053
MYOPIC2	.143	.128	.631	.658	.390	.007

τ = tardiness factor
R = due date range

FLOW SHOPS

Table 12. Heuristics Across Criteria

4 machines, 20 jobs

Rule	Norm. Weight. Tardiness	4 machines, Max Norm. Tardiness	20 jobs Portion of Tardy Jobs	WIP	WIS
FCFS	1.129	.326	53.6%	2.24	3.11
WSPT	.571	.145	50.3%	1.91	2.84
EDD	.769	.176	56.5%	1.94	2.48
S/PP	.805	.193	67.1%	2.00	2.51
COVERT	.464	.115	52.3%	2.04	2.70
MYOPIC1	.437	.108	48.5%	1.98	2.62
MYOPIC2	.390	.100	45.5%	1.93	2.57

4 machines, 60 jobs

Rule	Norm. Weight. Tardiness	4 machines, Max Norm. Tardiness	20 jobs Portion of Tardy Jobs	WIP	WIS
FCFS	2.672	.367	47.8%	5.09	8.12
WSPT	1.167	.134	44.7%	3.72	6.99
EDD	1.500	.151	43.2%	3.69	5.58
S/PP	1.459	.183	51.4%	3.80	5.52
COVERT	.706	.083	38.0%	4.45	6.53
MYOPIC1	.654	.071	34.6%	3.72	5.68
MYOPIC2	.619	.070	33.3%	3.79	5.80

8 machines, 60 jobs

Rule	Norm. Weight. Tardiness	4 machines, Max Norm. Tardiness	20 jobs Portion of Tardy Jobs	WIP	WIS
FCFS	1.691	.227	49.5%	4.52	6.33
WSPT	.786	.085	46.2%	3.80	5.69
EDD	.985	.097	47.5%	3.84	4.94
S/PP	1.004	.123	57.1%	3.96	4.99
COVERT	.505	.055	42.1%	4.24	5.49
MYOPIC1	.474	.049	39.9%	3.87	5.03
MYOPIC2	.431	.048	37.4%	3.86	5.07

tied, S/OP, COVERT, FCFS. The ranking for WIS is (EDD, S/op) nearly ties, MYOPIC1, MYOPIC2, COVERT, WSPT, FCFS. Notice that the myopic rules are much more robust across differing criteria than COVERT or WSPT. In fact, if we average the inventory measures, MYOPIC2 averages first and third, a truly robust achievement. By comparison, COVERT ranks third and fifth. (COVERT behaves especially badly for larger problems in either WIP or WIS).

These results are very encouraging for further development of myopic heuristics in other realistic environments.

7. Other Current Related Research

Turning first to continuing efforts on the parallel machine problem, the authors have recently devised a method for determining optimal solutions which needs no more alternatives to be investigated than the one machine case. (Although there is more computation per alternative.) For either equal or uniform machines there is a 1-1 correspondence between the order in which jobs are assigned to the first finishing machine, and the resulting schedule. Ties are not a problem, given specification of a tiebreaking mechanism. Thus there are only n! schedules to consider as in the one machine case. Lower bounds are obtainable as a linear assignment problem; branch on the first job rather than the last. Thus it now becomes feasible to run some smaller problems using the optimum solution for comparison.

The authors have recently completed a study giving a number of simple propositions which hold true in any pre-emptive proportionate flow shop. Some of these results were necessary for the optimal lead time decomposition theorem given in Section 5. A rough statement of the propositions: 1. Permutation schedules constitute the set of dominant schedules for any regular measure of performance. 2. The completion time of any job off any machine is

a simple expression involving total actual processing for the job plus waiting time for all previous jobs at the bottleneck machine. 3. Any permutation schedule minimizes makespan. 4. Several "tardiness" results. 5. A "no passing" result. 6. A "no waiting" result. The importance of these results is partly that they can help form good heuristics for the non-preemptive case for shops which are even roughly proportional [12].

In a nearly proportionate flow shop, permutation schedules should, as a result, be nearly optimal. Thus lead times can be calculated directly rather than by iteration, and non-bottleneck machines may often be ignored. Peng Si Ow [12] has recently obtained theoretical and experimental results, using these ideas, when there is a bottleneck machine. She compares her heuristic with earlier heuristics and our myopic heuristics, and often can achieve considerable improvement.

Vepsalainen is also currently extending the myopic procedures presented here for the dynamic job shop. He has run several computational studies similar to those given here for the flow shop, with similar results. MYOPIC2, MYOPIC 1, COVERT remain the best performers, in that order. He is also currently investigating "semi-myopic" rules which modify priorities somewhat by considerations at the next machine(s). His work is in the same spirit as the AWINQ, but crafted carefully into the current myopic heuristics. This work forms the last two chapters of his thesis.

8. Potential Future Research

Adding time constraints to the one machine problem would actually make optimum benchmarks easier to obtain, but would seem to require complicating the heuristics somewhat. Job start constraints (dynamic arrivals) only require restricting the heuristic evaluation each time a job finishes to those

currently available. (It has already been tested somewhat in the dynamic job shop work.) Job ending constraints (late absolute due dates) could be handled easily with two passes (which unfortunately destroys a lot of the dispatch advantage). The first pass is done as before, ignoring ending constraints. Second pass: find the earliest late due date violated, and move that job forward in priority until it just beats its deadline; then find the second date violated, etc. If the second job runs into the first in the process, both must be moved back, etc. It is easy to see a feasible schedule will be created if there is one. Turning to scheduled downtimes: machine periods of unavailability are easy for regular situations. If a machine operates 40 hours of 168 each week, treat it as a machine 40/168 as fast operating full time.

Adding precedence constraints to the one machine problem also makes benchmarks easier to obtain. The analogous result to EDD for precedence constraints is known, and is exact. Define the initial set of a job as all its predecessors. The extended EDD rule is: find the job with the earliest due date. Process first all its predecessors in any feasible order, then the job itself. Now repeat for the unprocessed part of the network. The analogous result to WSPT for precedence constraints is an excellent heuristic given and tested in [23]. The extended WSPT rule is: calculate the aggregate weight and the aggregate processing time for each (job, initial set). The job and initial set with highest aggregate weight per unit aggregate processing time should be run first. In this subproblem, the job must be run last. In the initial set pick the job with highest ranking in the same way; repeat. Eventually the whole subproblem will be complete. Repeat the process on the rest of the problem. We now have all the elements: due date, processing time, weight for use of our myopic heuristic. Calculate a priority for every

job, to determine which (job, initial set) to go first. In that subproblem put the job last, and repeat on the sub-problem etc. Putting these ideas together with those of the last paragraph it should be possible to provide accurate answers to very complex one machine tardiness problems. There are many such problems in real life, for example, planning one's own activities. In addition, there is reason to hope that complex job shops handling projects instead of individuals jobs, could often usefully concentrate on the bottleneck machine, which could be analyzed using these ideas.

Turning to future research on parallel machines, the uniform (proportionate machine) case is a rather easy extension. Weights and due dates are the same. Use the weighted average processing time on all machines for the formula. Compute the priorities each time a job is to be assigned, and assign to the machine that finishes it first. Note that the load is kept rather even, so that makespan is handled as well as a secondary objective. The general unequal case with processing times p_{ij} is more difficult. The following sorts of heuristics might be considered. In heuristic type 1, for each assignment calculate the priorities as before. All jobs within a certain cutoff tolerance priority deviation from the highest are considered to be scheduled next. For these jobs, try scheduling each on each machine, and schedule that job on that machine which completes it earliest. This rule should perform well on tardiness, but only fair on makespan. In heuristic type 2, we consider makespan to be capturing total processing capability of the machines. Suppose w_i now represents the dollar value of completing job i. Suppose based on historical data, machine j can process d_j dollars on average per hour. Then the value per hour of assigning job i to machine j is $(w_i/p_{ij}) - \lambda d_j$ λ is a parameter set between 0.0 and 1.0 depending on how busy the machine is. Select jobs within a priority tolerance as before. For

every such job i and machine j calculate the net per hour value of the assignment as stated. However, if a machine becomes scheduled more than T hours into the future it is temporarily "closed" and only remaining machines compete. (Without the tardiness objective the same makespan procedure could be used with all jobs checked instead of those within a tardiness tolerance.)

The generalized flow shop (sets of parallel machines in series) should be straightforward to study. Treat each parallel center as a single machine with a single priority list (as for the parallel case), which then reduces the problem to an ordinary flow shop. (In the case that each center is nearly uniform, simply replacing each center by a single faster machine is postulated to be an excellent approximation in many cases.)

Turning finally to other criteria, the authors have devised a methodology for transforming general convex cost criteria as a function of lateness into myopic priority functions similar to those discussed here. This means that such mixed criteria as 1. (weighted) earliness + (weighted) tardiness, 2. (weighted$_1$lateness + (weighted) tardiness, can be studied. The first is already under investigation [17]. More complicated mixed criteria such as including setup costs are much more difficult, however some preliminary heuristics have been designed.

As a longer range goal, the authors are working toward robust heuristics for criteria which are combinations of lateness, tardiness, makespan (efficiency), and setups, in job shops with such real constraints as precedence, availability of (machines, tools, people), shifts, maintenance. Such heuristics, to be useful must also given though to dovetailing with lotting procedures.

References

[1] Baker, K. R. "Introduction to Sequencing and Scheduling," John Wiley & Sons Inc., New York, 1974.

[2] Baker, K. R. and Bertrand, J. W. M. "A Dynamic Priority Rule for Scheduling against Due-Dates," Journal of Operations Management, Vol. 3, No. 1, pp. 37-42, November 1982.

[3] Bernardo, J. J. and Lin, K. S. "Scheduling Independent Jobs On Parallel Processors," Research Paper.

[4] Carroll, D. C. "Heuristic Sequencing of Single and Multiple Components," Ph.D. Thesis, Sloan School of Management, M.I.T., 1965.

[5] Conway, R. W., Maxwell, W. L., and Miller, L. W. Theory of Scheduling, Addison-Wesley Inc., Reading, Masachusetts, 1967.

[6] Dannenbring, D. G. "An Evaluation of FlowShop Sequencing Heuristics," Mangement Science, Vol. 23, 1977.

[7] Dudek, R.A., Panwalkar, S. S. and Smith, M. L. "Sequencing Research and the Industrial Scheduling Problem" in "Symposium on the Theory of Scheduling and Its Applications" held at North Carolina State University, 1972 edited by S. E. Elmaghraby, Springer-Verlag, New York 1973.

[8] Eastman, W. L. "Comments on a Paper by Schild and Fredman," Management Science, Vol. II, pp. 756-755, 1965.

[9] Elmaghraby, S. E. and Park, S. "On the Scheduling of Jobs on a Number of Identical Machines," AIIE Transactions, Vol. 6, pp. 1-13, 1976.

[10] Barnes, J. W. and Brennan, J. J. "An Improved Algorithm for Scheduling Jobs on Identical Machines," AIIE Transactions, Vol. 9, pp. 25-31, 1977.

[11] Fisher, M. L., "A Dual Algorithm for the One Machine Sequencing Problem," Mathematical Programming, Vol. II, pp. 229-251, 1976.

[12] Fox, Robert 'OPT'.

[13] Lawler, E. "On Scheduling Problems with Deferral Costs," Management Science, Vol. 11, No. 2, October 1966, pp. 280-288.

[14] Lenstra, J. K., Sequencing by Enumerative Methods, Mathematisch Centrum, Amsterdam, 1977.

[15] Montagne, E. R. Jr., "Sequencing With Time Delay Costs," Arizona State University Industrial Engineering Research Bulletin, pp. 20-31. January 1969.

[16] Morton, T. E. and Dharan, B. G. "Algoristics for Single Machine Sequencing with Precedence Constraints," Management Science, Vol. 26, No. 10, June 1978, pp. 1011-1020.

[17] Morton, T. E. and Ow, Peng Si, "Single Machine Early/Tardy Problem," GSIA Working Paper, Carnegia-Mellon University, Pittsburgh, 1984.

[18] Morton, T. E. and Rachamadugu, R. V., "Myopic Heuristics for the Single Machine Weighted Tardiness Problem," Technical Report #CMV-RI-TR-83-21, Robotics Institute, Carnegie Mellon University, Pitsburgh, November 1982.

[19] Nunni Khoven, T. S. and Emmons, H. "Scheduling on Parallel Machines to Minimize Two Criteria Related to Job Tardiness," AIIE Transactions, Vol. 19, pp. 288-296, 1977.

[20] Ow, Peng Si "Focused Scheduling in Proportionate Flowshops," GSIA Working Paper, Carnegie Mellon University, Pittsburgh, 1983.

[21] Rachamadugu, R. V. and Morton, T. E. "Scheduling Jobs in a Proportionate Flowshop" presented at TIMS/ORSA Conference in San Diego, October 1982.

[22] Rachamadugu, R. V. and Morton, T. E. "Myopic Heuristics for the Weighted Tardiness Problem on Parallel Processors," Technical Report #CMV-RI-TR-83-17, Robotics Institute, Carnegie Mellon University, Pittsburgh, August 1983.

[23] Rost, J. G. "Scheduling with Deadlines and Loss Functions on K Parallel Machines," Management Science, Vol. 11, No. 3, January 1965.

[24] Schild, A. and Fredman, I. J., "On Scheduling Tasks with Associated Linear Loss Functions," Management Science, Vol. 7, pp. 280-285, 1961.

[25] Vepsalainen, A., Rachamadugu, R. V. and Morton, T.E. "Lead Time Iteration in Flow-Shop Scheduling," GSIA Working Paper, Carnegie Mellon University, Pittsburgh, April 1982.