Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

DYNAMIC RESCHEDULING OF A JOB SHOP: A SIMULATION STUDY

Anand S. Kunnathur
P.S. Sundararaghavan

ISOM Department
The University of Toledo
Toledo, OH 43606-3390, U.S.A.

Sriram Sampath

Cap Gemini America
25800 Science Park Drive, Suite 180
Beachwood, OH 44122

ABSTRACT

The development of a rule based Expert System (ES), driven by a discrete event simulation model, that performs dynamic shop scheduling is described. Based on a flowtime prediction heuristic that we have developed and base-line runs to establish the efficacy of scheduling strategies such as SPT, Critical Ratio, Total Work etc, a re-scheduling based dispatching strategy is investigated in a dynamic job shop environment. The results are discussed and analyzed.

1 INTRODUCTION

Global competition has enhanced the significance of manufacturing effectiveness. Better manufacturing schedules provide competitive advantage through reduced production cost by way of lesser inventories and increased productivity. Moreover, increasing global competition has driven many US companies into investing heavily in Advanced Manufacturing Technologies, like Flexible Manufacturing Systems (FMS) and Computer Integrated Manufacturing (CIM). These manufacturing systems have created a whole new set of operational problems which emphasize the importance of superior job scheduling strategies.

In this increasingly competitive manufacturing environment, simulation has gone from a tool of "last resort" to being viewed as an invaluable design and problem solving methodology (Shannon 1988).

Artificial Intelligence (AI), the technology that attempts to preserve domain intelligence (Knowledge Base) in order to use the same for decision making in the future, has matured enough to redirect the research in scheduling. There are several capabilities of AI that make this technology particularly suitable for scheduling (Shaw, Park, Raman 1990).

In this paper we report the development of a generic rule based Expert System (ES) which is driven by a front end simulation engine and which uses several heuristics developed by us. We also report on their performance.

The rescheduling heuristic developed in this work uses a flowtime prediction strategy. We have also investigated the effectiveness of this strategy to set due-dates for newly arriving jobs. The organization of the paper is as follows.

The next section reviews the relevant literature on expert system shells in general and specific ES shells developed for scheduling. Section 3 presents the mechanics of the ES shell, detailed problem description and notation. The shop and job characteristics are also described in detail. SIMAN-V and C's role in the model is described. The objective measures that have been used in the model are also explained.

Section 4 describes the simulation model and the scheduling rules that have been used to obtain the baseline data used in the ES shell. Section 5 describes the studies that have been conducted using the ES shell and includes a review of relevant job shop scheduling literature. A flowtime prediction heuristic, based on dynamic shop conditions, is developed in Section 6. Various scenarios are discussed where such prediction is useful and possible.

Section 6 also describes our re-scheduling experiments in detail. The re-scheduling heuristic that has been developed in this work is elaborated. Results from the heuristic are compared and analyzed with the baseline results.

2 LITERATURE REVIEW

Expert Systems Applications developed probably constitutes the major segment of Artificial Intelligence research that has been conducted in the past two decades. We discuss the job scheduling related ES in this review. Job Shop scheduling has conventionally been looked upon as a computationally complex mathematical problem. Recent research on the real-world domain of scheduling demonstrates that the computational complexity issue is a minor part of the job shop scheduling problem in the real world and that expert schedulers do exist. (Mckay, Buzacott, Charness and Safayeni 1989). Mathematically, a dynamic shop scheduling problem is "*np*-complete", (Morton, Pentico 1994) which justifies heuristic methods.

The first real prototype expert system for scheduling was ISIS, developed by Fox and Smith (1984). It uses backward interval scheduling and beam search, with an evaluation function defined by estimation of constraint violations. The OPIS system (Ow, Smith and Howie 1988) employs an opportunistic approach to improve upon ISIS. It selects the most appropriate strategy for scheduling opportunistically; the resulting flexibility achieved in problem solving results in better performance.

OPAL is a multi-knowledge-based commercial scheduling software developed by Bensana, Bel and Dubois in 1988. Shaw (1988) has developed a knowledge-based scheduling approach based on the problem-solving techniques developed in artificial intelligence. Wysk, Wu and Yang (1986) use multipass expert system to decide the appropriate scheduling rules based on information such as the current system status, scheduling objective and management goals.

Shaw, Park and Raman (1990) use a machine learning approach to perform intelligent scheduling. Based on simulation runs they determined the most effective dispatching rule for a set of system attributes. O'Keefe and Rao (1992) use a look-ahead simulation strategy to evaluate the performance of candidate dispatching rules in an FMS cell.

3 RULES BASED ES SHELL FOR SCHEDULING

3.1 General Shop Conditions

The ES shell developed in this work resides on a discrete event simulation model of a production shop. Most of the researchers in the past have used systems involving shops having anywhere between five and twelve machines. Conway (1965) performed an elaborate study on flow shop with nine machines and 9000 jobs simulated over 30 different rules. Muhlemann *et al.* (1982) conducted a study on a job shop consisting of 18 machines. Ahmed and Fisher (1992) used a five machine job shop to conduct their study on due-dates. Fry *et al.* (1988) used a ten machine job shop to conduct their study on processing time dispatching rules. Ragatz and Mabert (1994) describe a nine machine shop in the explorative study on due date assignment rules. The simulation model that we have developed is tested on a typical job-shop with no break-downs.

3.2 Dynamic Shop Design

The dynamic shop modeled in the discrete event simulation engine consists of ten machines and uses warm-up period to make the shop representative. The job arrivals are modeled with exponential inter-arrival time and the process times have been assumed to be normal with relatively low variance.

Table 1: Processing-time Distribution at Various Machines

Mac->	1	2	3	4	5	6	7	8	9	10
Mean	15	17	12	18	19	17	14	12	18	9
Std.D	2	2.5	1	3	2.5	2	1.5	0.5	2.75	1

Table 1 gives the processing time distribution at each of the machines. The inter-arrival time of jobs is exponential with an average of 6/hour. In the Job Shop Model mode every job can have anywhere between 2 and 8 operations, i.e., the number of operations per job is drawn from a discrete probability distribution with a minimum of 2 and a maximum of 8. The visitation sequence is randomly selected (the next machine in the sequence is drawn from a discrete probability distribution with a minimum of 1 and a maximum of 10; every unvisited machine has an equal probability of being selected) with no machine being visited twice by the same job.

Due Date assignment is done internally using the Total-Work (TWK) rule for all the runs. The TWK (Total Work) rule is as follows:

$$d_j = A_j + k \sum_{i \in mj} p_i \qquad (1)$$

where d_j - Due Date of job j

A_j - Arrival time of j

p_i - Process time of the j in machine i.

k - Parameter than reflects the amount of expected delay a job will experience. The parameter k is 3,4,5 or 6 with equal probability.

m_j - Set of machines visited by job j

3.3 Objective Measures Studied

The following objective measures have been fused in the ES shell: **Average Flowtime, Average Tardiness and Percentage Tardy**. With the arrival rates and processing times assumed, the Dynamic Job Shop has an average machine utilization of 75.7%, with a maximum of 95% and a minimum of 45%.

3.4 EXPERT SYSTEM SHELL DESIGN

SIMAN-V™ has been used as the simulation modeling tool. All the experiments were done on a Gateway ® 2000 Pentium-66Mhz. The *jobs* in the dynamic shop are *entities* created in the SIMAN-V simulation model. SIMAN-V allows memory pointers to be accessed by C. The entire rule based scheduling module is written in C and compiled in WATCOM® C compiler. The C object module is linked with SIMAN-V runtime libraries to generate a final run-time module.

Entities (memory variables) are created by the discrete event simulation model based on the arrival distribution. The created entities are assigned job attributes, viz., process times, due date, predicted flow time and the machining sequence. The entities enter the shop where they are processed by the machines. The entity leaving the machine after completion of processing triggers the ES shell which schedules the next entity to the free machine. The entity gets processed by the appropriate machine as per the machining sequence. Upon completion of all required processing the entity leaves the system after updating the statistics.

The rule based scheduling shell dispatches jobs depending on the system characteristics and objective measures selected in the user interface by the user. The system currently uses simple IF..THEN..ELSE rules to schedule jobs. Experimental data from past simulation runs serve as a knowledge base to guide decision making when the ES is used to make accept/reject decisions and perform adaptive re-scheduling.

4 TESTING AND ENHANCEMENTS

4.1 Dispatch Heuristics Tested

The ES shell with the front end simulation engine is generic and hence can simulate a variety of job shops. This section describes the baseline or one-pass heuristics tested in the scheduling engine. The baseline runs have been conducted in order to test the ES shell. The results from the baseline runs have been analyzed here prior to the development of heuristics to address the objectives.

One-pass heuristics are simple heuristic procedures that build up a single complete solution a step at a time (Morton, Pentigo 1994). These are very useful primarily because these serve as simple solutions which meet specific shop objectives. These are iteratively used to build more sophisticated multi-pass heuristics or search heuristics that fit well in a rule based ES development paradigm.

- We have experimented with a number of one-pass heuristics after a thorough search of job shop scheduling literature. (Muhlemann 1982, Morton, Pentigo 1994) For convenience we have named these one-pass heuristics as *pure strategies*. Performance of objective measures from these pure strategies have served as fundamental experimental data to drive the ES shell. We have classified these pure strategies into ***simple dispatch heuristics*** and ***simple bottleneck heuristics***.

Simple dispatch heuristics profess a scheduling strategy that is based on the simple premise: *highest priority first*. Whenever a machine becomes available, we simply schedule the highest priority job currently available at that machine. Dispatch heuristics that have been used in this work:. *Static* priorities tried were First In First Out (FIFO), and *dynamic* priorities like Minimum Slack First, Slack Per Operation, Minimum Total Work First were also tried. *Local* priorities like Shortest Processing Time first and *forecast* priorities like Critical Ratio and Least Work Remaining have also been tried.(Standard definitions have been used.)

4.2 Bottleneck Heuristics Tested

Bottleneck heuristics are dispatch strategies where the bottleneck machine is identified from the set of machines available in the shop and the scheduling strategy at the bottleneck machine is different from the other machines.

A machine is defined as a *bottleneck* of the shop at time *t* if it has the maximum work-load among all machines at time *t*. In a dynamic environment the bottleneck tends to be dynamic depending on the jobs that are currently available in the system. In the simulation model the work-load is calculated for all machines whenever a new job enters the system or whenever a job leaves the system after completion of processing. The machine with the maximum work-load is deemed as the *bottleneck*. Scheduling at the bottleneck machine is different from the non-bottleneck machines. Because of the superior performance of SPT in comparison to the rest of the simple dispatch heuristics when it comes to flowtime criterion, all bottleneck machines have been scheduled on SPT. The non-bottleneck machines have been scheduled using four different options (LWR, EDD, TWK, Least Slack/opn)

4.3 Results from the Baseline Runs

The simulation model was validated based on the input parameters and the performance of objective measures. Using the run length analysis tools, viz., SIMAN output analyzer, available in ARENA, the Graphical User Interface that resides on SIMAN-V, the run length and warm-up period were determined after extensive experimentation. Using a moving average analysis of flowtime the initial condition bias (warm-up) was determined to be 20 days. The total run length was 80 days. Therefore data was collected and analyzed for 60 days.

The results from the baseline runs are tabulated in Table 2 and 3. The results of a paired difference t-test conducted between results obtained using SPT vs EDD, and SPT vs FIFO indicate clearly that there is a statistical difference between all the objective measure values between various dispatching rules.

Both simple dispatch and simple bottleneck heuristics are reported in Table 2. As can be seen from the Tables, SPT performs best on flowtime, followed by Bottleneck - SPT/Non-Bottleneck - LWR rule. However TWK rule performs the best when it comes to percentage tardy as objective measure. On the percent tardy criterion there is hardly any statistical difference between Bottleneck - SPT/Non-Bottleneck - LWR rule, Bottleneck - SPT/Non-

Bottleneck - TWK rule and SPT rule.

EDD, Slk/Opn and Critical ratio perform equally well when it comes to Average tardiness criterion. The most interesting result of the whole exercise is that all Bottleneck based rules perform well for all the objective measures taken together. As identified by other scheduling literature, SPT is probably the best when it comes flowtime criterion.

Table 2: Performance of dispatching rules

Characteristics	SPT	EDD	FIFO	TWK	MinSlk	LWR
Flowtime	314.84	333.88	353.69	325.42	333.88	350.35
Average Tardiness	110.11	59.57	84.55	123.18	59.57	137.00
Percentage Tardy	17.64	36.62	41.35	12.61	36.62	18.58

Table 3: Performance of dispatching rules

Characteristics	Slk/Op	C.Rat.	B-SPT N-EDD	B-SPT N-LWR	B-SPT N-TWK	B-SPT N-S/o
Flowtime	340.31	337.67	325.33	317.49	318.92	331.31
Average Tardiness	57.74	59.80	84.64	105.35	113.36	78.28
Percentage Tardy	35.28	37.55	23.15	19.48	16.34	22.18

Fry, Philipoom and Blackstone (1988) present a list of truncated SPT rules which attempt to capture the effectiveness of SPT in meeting the flowtime criterion but perform well with tardiness criterion. Our bottleneck based rules seem to do exactly that when one looks at all the criterion in total. Truncated rules are difficult to implement whereas bottleneck based rules are simple and easily implementable. We have seen what rules are appropriate for each of the objective measures. In the next section we describe studies conducted in dynamic job shop re-scheduling using the experimental data gathered. We first develop a flowtime prediction heuristic for the purpose of dynamic rescheduling.

5 DYNAMIC RESCHEDULING

5.1 Flowtime Prediction

Predicting flowtime in a dynamic environment is a difficult task primarily because of the fact that there is no definite method to estimate the waiting time of the job at various queues in a shop. However, knowing an estimate of flowtime before the acceptance of an order is important to give a delivery commitment to the customer. In this increasingly customer-driven market the probability of success of a firm in the market place is directly dependent on its ability to give and adhere to delivery commitments. Conventionally, researchers in dynamic scheduling have tried to address delivery commitments through improved due-date setting. We examine relevant literature on due-date setting in this section.

Eilon and Chowdhury (1976) used expected waiting times and current queue lengths to more accurately estimate job flow time. Weeks (1979), in a simulation study of a 24 machine job shop having labor and machine constraints concludes that rules based on the number of jobs in the system, provide good due date performance.

Ragatz and Mabert (1994) conducted a simulation analysis of due-date assignment rules. They report on a number of performance measures including: standard deviation of lateness, mean tardiness, and mean absolute missed due dates and made several interesting comments about the role of shop characteristics and status in affecting performance. We have developed a flowtime prediction heuristic in this paper, which uses both characteristic and shop status information.

Eilon and Chowdhury (1976) used expected waiting times and current queue lengths to estimate job flow time. Our flowtime prediction heuristic uses expected waiting times and current queue lengths. However our estimation uses the job's processing time, along with queue lengths. Ragatz and Mabert (1982) demonstrated a methodology to include both job characteristics and shop status information for setting due dates.

5.2 Flowtime Prediction Heuristic

Weeks (1979), in a simulation study of a 24 machine dual constrained (machine and labor) shop, used a rule, based on the number of jobs in the system, to provide good due date performance. Even though Week's heuristic used information on total shop congestion as a basis for developing flow time estimates, it ignored the job characteristics. Ragatz and Mabert (1994) in their study found the Weeks rule to be performing rather poorly in comparison to other rules which use both job and shop characteristics.

The flowtime prediction heuristic developed in this work uses both job and shop characteristics. The heuristic assumes that the dispatching rule is SPT. (SPT has performed the best when it comes to mean flowtime criterion.) The heuristic is as follows:

5.3 Heuristic I

Suppose a new job arrives at time t:

Step-1. The percentile to which the simulated processing time of job belongs (based on the processing time distribution at each work-station) is determined at each work center.

Step-2. Using the actual queue size at each work-station at that point of time in the simulation run the potential location of the new job at various queues is ascertained, using the information in *step 1*, ignoring all other jobs that may be received after the receipt of this job.

Step-3. Using the information in *step 2* the projected waiting time of the job at each work-station is determined and summated over all the work-stations that the job would visit. Mathematically the waiting time at time t of the job at various queues can be expressed as,

$$\text{Waiting time } W^t = \sum_{j=1}^{n} (m \times \overline{P}j) \qquad (2)$$

where, m - number of *jobs* in front of the current *job*.

$\overline{P}j$ - Mean processing time at machine j.

n - Number of machines visited by the *job*.

Step-4. The waiting time determined in *step 3* is added to the TWK (total work) of the job and the whole value is multiplied by an experimental factor γ to give the predicted flowtime. Mathematically,

$$\text{Predicted flowtime} = \gamma(W^t + \text{TWK}) \qquad (3)$$

Initial simulation runs using $\gamma = 1$ were performed. A comparison of the average predicted flowtime and average actual flowtime resulted in a value of 1.75 for γ. This value is dependent upon the shop characteristics and reflects the degree of bottlenecking in the shop.

5.4 Flow Time Prediction Results

In order to investigate the effectiveness of the flowtime predicting heuristic the predicted flowtime was compared with actual flowtime. Ten replications of the simulation were conducted with SPT dispatching. A *paired difference t* test to examine whether there is a statistical difference between the distributions of the actual and predicted flowtimes was negative. The results indicate that there is no statistical difference between the two distributions. Therefore the flowtime prediction heuristic is effective in giving an estimate of the actual flowtime. Comparison of the averages indicated that this heuristic could certainly be used to serve as a tool for rescheduling.

5.5 Rescheduling

Historically there have been two approaches to scheduling: *sequencing*, the approach that seeks to establish an order for all open jobs and *dispatching*, the approach that provides a solution by the use of local rules for selection of one job from the list of available jobs at decision epochs. It has been reported that sequencing approach is more efficient than the dispatching approach in a pure static environment. (Matsura, *et al*; 1993) However in a dynamic environment it is practically impossible to adopt a total sequencing approach, simply because the problem cannot be solved satisfactorily. Therefore dispatching is probably the only solution. *Re-scheduling* as has been defined earlier is a goal driven strategy that attempts to involve shop characteristics, shop objectives and dynamic shop status information to perform effective dispatching.

Muhlemann, *et.al* (1982) investigated a rescheduling strategy for a 15 machine job shop. They concluded that frequent scheduling can produce significant improvements on shop performance. Matsura, *et.al* (1993) suggest that a rescheduling (referred as switching strategy by the authors) strategy is very efficient when there are uncertainties in the system. Garg and Wang (1989) demonstrate that a rescheduling based strategy is more effective in meeting shop objectives in a Flexible Manufacturing Cell (FMC).

Rescheduling can be triggered by a change in the shop conditions. The challenging exercise is therefore to identify shop conditions which could serve as a vehicle to trigger rescheduling. In the rescheduling heuristic developed in this work, variation of predicted mean flowtime from the actual mean flowtime serves as the trigger to perform rescheduling. The rescheduling strategy reported in this paper uses the flowtime prediction heuristic described in the previous segment. The rescheduling heuristic is as follows:

5.6 Heuristic II

Suppose a new job arrives at time t:

Let,

k - Jobs completed

n - Jobs in the system (Work in progress)

p - Proportion (w.r.t total number possible in the whole simulation run) of jobs completed in time *t*.

f_1 - Mean actual flowtime of completed jobs

f_2 - Mean predicted flowtime for all work in progress (W.I.P)

f_o^o - Mean adjusted flowtime based on experimental runs

f_3 - Mean flowtime estimate for $(k + n)$ *jobs*

f_o^t - Predicted flowtime mean at time *t*

then,

Step-1. The flowtime for the job is predicted based on the Heuristic-I described earlier and the mean predicted flowtime (f_2) for all work in progress (W.I.P) is recalculated.

Step-2. Whenever a job completes processing (time *t*) the following are computed:

f_3 - Mean flowtime estimate for $(k + n)$ *jobs*

$$= \frac{(f_2 n + f_1 k)}{(k + n)}$$

f_o^t - Mean predicted flowtime at time *t*

$$= f_o^o\,(1-p) + f_1 p$$

Based on straight simulation runs f_o^o is determined as follows:

$$f_o^o = \bar{p} \times \left(\bar{m} + \frac{\sum_{i=1}^{r} p_i q_i}{r} \right)$$

where $\bar{p}$ = Average processing time of all machines in the shop

$\bar{m}$ = Average number of machines visited by any job in the shop

q_i = Average queue length of the i[th] machine based on prior simulation runs

r = The total number of machines in the shop, 10 in this study.

For the shop and job characteristics used in the model f_o^o was determined to be 263.84 minutes.

Step-3. Decision to reschedule: The rescheduling parameter is determined based on the parameters computed in *step 2.* The rescheduling parameter (at time *t*) is given as:

$$R = \frac{(f_3 - f_o^t)}{f_o^t}$$

The decision to reschedule is then based on the following rule:

If $R > \Delta \rightarrow$ RESCHEDULE

$R \leq \Delta \rightarrow$ HOLD OFF

where Δ is an experimental decision variable.

Note: Once a reschedule has been performed further reschedules are possible only after a simulation run for time *T*. This is done to ensure that one does not go overboard in rescheduling every other minute.

Δ and *T* are experimental variables used in this research to study their influence on the performance of the rescheduling heuristic.

5.7 Candidate Rules for Dispatching

The rescheduling heuristic described above describes the process of arriving at a decision to reschedule. An analysis of the baseline runs indicated that bottleneck based rules perform well in meeting overall objective measures. Therefore two out of our three candidate rules for dispatching are bottleneck based rules. The ES shell uses two sets of dispatching rules, one for the *flowtime minimization* criterion and the other for *tardiness minimization* criterion. The candidate rules for each of the criteria are as follows:

(i) Flowtime Minimization:

The following rules are the candidate rules for the flowtime minimization criterion; (a) SPT, (b) Bottleneck Machine - SPT and Non-bottleneck machines - LWR and © bottleneck Machine - SPT and Non-bottleneck machines - TWK. Whenever our heuristic advises a reschedule, the dispatching rule is changed to a different rule (by rotation) within the aforesaid list of candidate rules.

(ii) Tardiness Minimization:

Based on an analysis of the baseline runs, the following dispatch rules perform well, for minimizing average tardiness: (a) EDD; (b) Slk/Opn (ratio of total slack to number of operations remaining) and © Critical Ratio. Therefore these rules are the candidate rules for the average tardiness minimization criterion. Rescheduling is done within this set of candidate rules in rotation.

We experimented with various values for the variables Δ and *T* to investigate their influence in the performance of objective measures. Next, we analyze the performance of the rescheduling heuristic for both criteria.

Table 4: Objective Measures for various values of Δ
(Flowtime minimization criterion; T=4 Days)

Characteristics	Δ=0.05	Δ=0.10	Δ=0.125	Δ=0.15	Δ=0.20
Flowtime	318.96	318.96	315.30	315.30	315.86
Average Tardiness	114.30	114.30	106.73	106.73	108.76
Percentage Tardy	16.78	16.78	16.38	16.38	16.42
No. of Reschedules	10	8	7	5	1

Table5: Objective Measures for various values of Δ
(Average Tardiness minimization criterion; T=4 Days)

Characteristics	Δ=0.05	Δ=0.10	Δ=0.125	Δ=0.15	Δ=0.20
Flowtime	333.08	329.79	329.79	327.90	330.60
Average Tardiness	90.06	76.94	76.94	88.98	90.68
Percentage Tardy	25.35	29.41	29.41	23.24	24.50
No. of Reschedules	11	8	8	6	5

5.8 Rescheduling Results

The performance of the rescheduling heuristic for both the criteria are reported in Tables 4 and 5. The rescheduling decision factor, Δ, directly affects the performance of objective measures for both criteria. Shop performance is very sensitive to Δ. The decision factor has a direct influence on the number of reschedules until a certain value. This is because Δ measures the variation of average flowtime vis-à-vis the average flowtime under SPT scheduling. A lower value for Δ would therefore mean more frequent re-scheduling and lower average flowtime. For larger values for the decision factor, there is hardly any rescheduling.

Table 5 describes similar results as in the case of tardiness minimization. However, it was observed that rescheduling does not seem to be more effective than EDD when it comes to average tardiness minimization criterion. This is attributed the fact that the rescheduling trigger is flowtime based. Further studies can be conducted with a tardiness based rescheduling criterion.

6 CONCLUSION

We have seen in this chapter that a rule based system works well for implementing a rescheduling based strategy. The rescheduling strategy developed in this research performs effectively for the shop characteristics used. We believe that the rescheduling strategy could perform even better for a different set of shop characteristics. However this calls for extensive experimentation with this heuristic. We have seen that an optimal (which is dependent on the system parameters) number of reschedules does result in a superior overall scheduling strategy. The effectiveness of rescheduling in tardiness minimization was not as profound as in flowtime minimization. However this could be attributed to the rescheduling trigger, which in our case is flowtime based. Simulation driven, expert system implemented rescheduling as a strategy, appears to hold a lot of promise. For complete details the reader is referred to Sampath (1996).

ACKNOWLEDGMENTS

The work of Sriram Sampath (1996)during his master' s program was partially supported by the ISOM Academic Challenge Grant of the College of Business Administration at The University of Toledo.

REFERENCES

Ahmed, Imtiaz and W.W. Fisher. 1992. Due Date Assignment, Job Order Release, and Sequencing Interaction in Job Shop Scheduling. *Decision Sciences* 23: 633-647.

Bensana, E., Bel, and D. Dubois. 1988. OPAL: A milti-knowledge-based system for industrial job-shop scheduling. *International Journal of Production Research* 26 (5): 795-819.

Conway, R.W. 1965. Priority Dispatching and Work In

Progress Inventory in a Job Shop. *Journal of Industrial Engineering* 16: 123-130.

Conway, R.W., W.L. Maxwell, and L.W. Miller. 1967 *Theory of Scheduling.* Addison-Wesley.

Eilon, S., and I.G. Chowdhury. Due Dates in Job Shop Scheduling. *International Journal of Production Research* 14 (2): 223-237.

Fox, M.S., and S.F. Smith. ISIS - A Knowledge Based System for Factory Scheduling. *Expert Systems* 1: 25-49.

Fry, T.D., P.R. Philipoom, and J.H. Blackstone. 1988. A simulation study of processing time dispatching rules. *Journal of Operations Management* 7 (4): 77-92.

Garg, A., and H. Wang. On Simulation Run Length: A Step Towards Real Time Control in an FMS. *International Journal of Operations and Production Management* 10 (9): 74-88.

Matsuura, H., H. Tsobone, and M. Kanezashi. 1993. Sequencing, dispatching, and switching in a dynamic manufacturing environment. *International Journal of Production Research* 31 (7): 1671-1688.

McKay, K.E., J.A. Buzacott, N. Charness, and F. Safyeni. Can Expert Systems deal with the job shop scheduling domain. *WATMIMS working paper* No. 89-01.

Morton, T.E., and D.W. Pentigo. 1994. Heuristic Scheduling Systems, John Wiley & Sons.

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 (2): 227-241.

O'Keefe, R.M., and R. Rao. 1992. Part input into a Flexible Flow System: An evaluation of look-ahead simulation and a fuzzy rule base. *The International Journal of Flexible Manufacturing Systems* 4: 113-127.

Ow, P.S., S.F. Smith, and R. Howie. A Co-operative Scheduling System. *Expert Systems and Intelligent Manufacturing*, edited by M.D. Oliff, North Holland: 70-89.

Ragatz, G.L., and V.A. Mabert. (1982) An Investigation of Flow Time predictability in a Single Machine Shop. *Proceedings*, Midwest AIDS: 154-156.

Ragatz, G.L., and V.A. Mabert. 1994. A simulation analysis of due-date assignment rules. *Journal of Operations Management* 5 (1): 27-39.

Sampath, Sriram. 1996. Unpublished M.S. Thesis, College of Business Administration, The University of Toledo, Toledo, OH 43606.

Shannon, R.E. 1988. Knowledge based simulation techniques for manufacturing. *International Journal of Production Research* 26 (5): 953-973.

Shaw, M.J. 1988. Knowledge-based scheduling in flexible manufacturing systems: An integration of pattern-directed inference and heuristic search. *International Journal of production Research* 26 (5): 821-844.

Shaw, M.J., S.C. Park, and N. Raman. 1990. Intelligent Scheduling with Machine Learning Capabilities: The Induction of Scheduling Knowledge. *Carnegie Mellon University — The Robotics Institute-Technical Report.*

Turban, E. *Decision Support Systems: Management Support Systems*, 2nd Edition, Macmillan.

Weeks, J.K. 1979. A Simulation Study of Predictable Due Dates. *Management Science* 25 (4): 363-373.

Wysk, R.A., S.D. Wu, and N. Yang. Multi-pass Expert Control System for Manufacturing Systems. *Proceedings of the Symposium on Real-time Optimization in Automated Manufacturing Facilities*, National Bureau of Standards, Gaithersburg, MD.

AUTHOR BIOGRAPHIES

ANAND S. KUNNATHUR received the master's degree in physics from the University of Delhi, the master's degree in mathematics from York University, Canada, and the Ph.D. degree in management science from the University of Tennessee. His research interests are in the areas of information systems, manufacturing management, and the interface between management science and functional areas of business. He has published in journals such as *Operations Research, TIMS* special issue on "Optimization in statistics," *SIAM Sci. and Stat. Computing, EJOR, Omega, Computers and Industrial Engineering, Information and Management*, and others.

P. S. SUNDARARAGHAVAN received the Bachelor's degree in chemical engineering from the University of Madras, the Post Graduate Diploma in management from the Indian Institute of Management in Calcutta, and the Ph.D. degree in management science from the University of Tennessee. He is a former Chairman of the ISOM Department at The University of Toledo and the founder of The University of Toledo Chapter of the APICS Society. He has published in journals such as *Naval Research Logistics, SIAM Algebraic and Discrete Methods, Decision Sciences, IIE Transactions, OPSEARCH, EJOR, Journal of Operations Research Society*, and others. His current research interests include sequencing and scheduling, simulation, production planning and control, and issues in advanced manufacturing technology adoption.

SRIRAM SAMPATH received the M.S. degree in Manufacturing Management from the College of Business Administration of The University of Toledo. He works for Cap Gemini in the area of manufacturing software development and customer support.