4
8-6-73

258

UCRL-51407

**RESEARCH ON PIPELINE PROCESSORS:
A REVIEW OF LITERATURE ON SCHEDULING**

David G. West

June 14, 1973

Prepared for U.S. Atomic Energy Commission under contract No. W-7405-Eng-48

**LAWRENCE
LIVERMORE
LABORATORY**

University of California/Livermore

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

NOTICE

"This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights."

Printed in the United States of America

Available from

National Technical Information Service

U. S. Department of Commerce

5285 Port Royal Road

Springfield, Virginia 22151

Price: Printed Copy \$ ____*; Microfiche \$0.95

<u>*Pages</u>	<u>NTIS Selling Price</u>
1-50	\$4.00
51-150	\$5.45
151-325	\$7.60
326-500	\$10.60
501-1000	\$13.60

TID-4500, UC-32
Mathematics and
Computers

LAWRENCE LIVERMORE LABORATORY
University of California, Livermore, California, 94550

UCRL-51407

**RESEARCH ON PIPELINE PROCESSORS:
A REVIEW OF LITERATURE ON SCHEDULING**

David G. West

MS. date: June 14, 1973

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

MASTER

NOT REPRODUCED

NOT REPRODUCED

leg

RESEARCH ON PIPELINE PROCESSORS: A REVIEW OF LITERATURE ON SCHEDULING

Introduction

The designing of pipeline processors includes problems in the scheduling of segments of various procedures for different hardware units (pipes) in the computer. This paper is a review of the literature on scheduling published primarily by business, industrial engineering, and operations research sources.

The production job shop seems to be the closest analog for the research problems in the design of a pipeline processor, and therefore it is discussed in detail.

Two ways to model scheduling are defined: static modeling and dynamic

modeling. The literature indicates that the development of feasible solutions to the problems of scheduling has involved the evaluation of various priority rules or combinations of priority rules to determine which is best for a specific application. Descriptions of various priority rules are included, and more complete mathematical explanations are located in the Appendix.

In the last section, the basic elements of heuristic scheduling are described. Because of its complexity, this procedure does not seem to be useful at this time for the scheduling problems in a pipeline processor.

Basic Methods of Modeling

There are two ways to model scheduling and dispatching:

1) Static scheduling

N jobs arrive simultaneously in a shop. No further jobs arrive.

2) Dynamic scheduling

Jobs continuously arrive at random intervals.

Each of these two types of scheduling can be further subdivided by

1) Number of machines

(single stage, $M = 1$; multistage, $M > 1$);

2) Nature of the job rate [Day and

Hottenstein, 1970].

Static Scheduling

The simultaneous start time for N jobs is usually created by holding job orders

over a period of time until N jobs can be released. This levels the shop load

over the planning period. Single-machine research is being used to optimize the more complicated multiple-machine problem [Day and Hottenstein, 1970]. The shortest processing time (SPT) rule is optimal for due date and nondue date criteria [Day and Hottenstein, 1970]. The multiple-machine case can be divided into three machine arrangements — series, parallel, and hybrid. Gantt charts can be used to schedule jobs on any of these arrangements. Optimization has used the make-span criteria in the series flow shop.

There seem to be four approaches used to study static scheduling:

- 1) combinational approaches — changing of one permutation to another by changing priorities of jobs;
- 2) general mathematical programming — i.e., linear, dynamic, convex, quadratic, and integer programming;
- 3) reliable heuristics — branch and bound techniques, "tree" searching;
- 4) Monte Carlo techniques. [Day and Hottenstein, 1970, p. 14]

Dynamic Scheduling

The single-machine case is a single-channel queuing problem. Usual assumptions are

- 1) Poisson arrival rates;
- 2) exponential service times;
- 3) first-come, first-served (FCFS);
- 4) mean service rate λ > mean

arrival rate of jobs.

Machine arrangements for the multiple-machine case are

- 1) Tandem queues;
- 2) Job shop: general network in which each order is processed by a particular subset of machines;
- 3) Parallel-channel queues [Gentry, 1972, p. 14].

The job shop seems to be the arrangement closest to the pipeline processor. It has been analyzed

by considering the job shop as a network of waiting lines with fixed short-run capacity, the short run being interpreted as one 8-hour shift . . . The early research on queuing suggested

that an important variable which could be manipulated in sequencing models was the priority dispatching rule (queue discipline) and much of the research has focused upon comparative studies of dispatching rules through the use of computer simulation [Day and Hottenstein, 1970, p. 16].

A powerful analytical result for studying dynamic job shops is Jackson's decomposition principle [Jackson, 1967]. Assuming that

- 1) job arrival times are exponentially distributed;
- 2) processing times at each machine are exponentially distributed;
- 3) jobs are routed to each machine by a fixed probability transition matrix (probability of a job going from a machine to another machine); and
- 4) the priority rule at each machine is FCFS; then the system can be decomposed into a network of independent individual-machine queuing systems.

Moodie and Roberts [1968] investigated the parallel processor shop to evaluate the performance of the following rules:

- 1) shortest processing time (SPT);
- 2) slack: jobs in order of increasing slack time;
- 3) due date;

4) first-come, first-served (FCFS)—used as a random rule;

5) weighted objective—a combination of rules.

They found that the SPT, slack, and weighted-objective rules were generally superior to the due date and FCFS rules.

Priority Rules

Many priority rules have been proposed, but no rule has been proposed that assures optimality. Priority rules can affect aggregate measures of shop performance [Day and Hottenstein, 1970]. Priority rules are compared on the basis of their effectiveness in meeting a pre-determined criteria [Day and Hottenstein, 1970]. Effectiveness is measured by the value of the criteria functions when a rule is used.

Priority rules can be classified as Local Job Status, Semi-Global, or Global Rules.

Local Job Status Rules have a priority determined only by the immediate jobs at the given machine; examples are Shortest Processing Time and Slack [Gentry, 1972].

Semi-Global Rules use information about some other jobs and machines.

Global Rules use information about all other operations in the schedule and are seldom used, because of their complexity.

The Appendix to this report contains priority rules that are typically used in research on dynamic job shops. Conway [1964] has investigated 92 rules of this type; this is the largest experimental investigation reported to date.

Four categories of job priority rules are

1) Lateness Rules—these rules determine priority according to some increasing function of lateness;

2) Arrival Order Rules—these rules assign priority according to the order in which jobs arrive;

3) Job Property Rules—some property of the job itself is used by these rules to determine priority;

4) Random Rule—job priority is assigned at random. [Conway et al., 1960]

The Lateness and Job Property rules are the primary rules used by industry.

Most of the experimental work done by industry on dynamic job-shop scheduling has concentrated on the effect of priority rules on the mean and variance of

1) Job lateness with respect to due dates;

2) Percent of jobs late;

3) Work-in-process inventory;

4) Order flow time;

5) Waiting time at each machine or in the total shop;

6) Number of orders in the shop;

7) Machine utilization;

8) Set-up time. [Day and Hottenstein, 1970]

Day and Hottenstein [1970, p. 21] commented that

A generalized statement of the relative effectiveness of priority rules in relation to a given criterion is difficult, if impossible, to make. The reason becomes apparent when one considers the many possible types of distributions, transition matrices, parameter settings, and other assumptions which may be used in developing a simulation of a

dynamic job shop. This is precisely the problem one encounters in studying the literature when noting, for example, that in one experiment priority rule X was the most effective rule in relation to criterion Y, whereas in another experiment employing the same set of priority rules, priority rule X was relatively ineffective.

Ranked Positional Weight

The ranked positional weight technique requires a precedence matrix for sequencing operations and a positional weight defined by the sum of the time for all following operations and the time of the operation

itself. The operations are ordered in the sequence of descending positional weight and then assigned to the machines. Sequencing is obtained from the precedence matrix. [Biegel, 1971, pp. 183-190]

Shortest Processing Time Rule (SPT)

The SPT rule requires that jobs requiring the least amount of time (including set-up time) be run first. This rule has proved to be a powerful scheduling rule in the dynamic job shop.

Usually, application of the SPT rule will tend to reduce the total lateness of all jobs. For example, take two jobs having processing times of 1 hour and 10 hours. Scheduling the shorter job first gives the longer job a lateness of 1 hour, while scheduling the longer job first gives the shorter job a lateness of 10 hours. Morse [1958, p. 241] has shown that the expected total waiting time can be minimized by using the SPT rule.

Richard Conway [1965] has compared the SPT rule to 31 other rules. Although the rule was not always the best measure

of work in process inventory, it did dominate the set of rules studied.

Performance under every measure was very good, it was an important factor in each of the rules that exhibit a 'best' performance under some measure and it is simpler and easier to implement than the rules that surpass it in performance . . . should be considered a 'standard' in scheduling research . . . [Conway, 1965, p. 129].

Another advantage of the SPT rule seems to be a lack of sensitivity to changes in various parameters. Conway and Maxwell [1962] tested this lack of sensitivity by introducing errors in estimated processing time and changing job shop size; they believed a better

nonlocal rule could be devised, but that it would have to be very complex.

The SPT rule is used in Johnson's rule to assign jobs to two machines. This rule states [Biegel, 1971, p. 190]:

1) Find the smallest processing time. If it is on Machine A, assign the job to Machine A as the first job. If it

is on Machine B, assign it to Machine B as the last job.

2) Remove the job just assigned from consideration. Go to step 1. If any two jobs have the same time on the same machine, the order of assignment is arbitrary. When all jobs have been assigned, you have a near optimal sequence.

Slack Time Rule (Slack Rule)

Slack time is the excess time available between the due date and the estimated completion date (arrival time plus processing time remaining for the job). Slack time can be considered static or dynamic.

The dynamic Slack Rule was implemented in a job shop at Hughes Aircraft

Company, El Segundo. They found excellent schedule performance in job shops with fixed completion schedules [Bulkin et al., 1966].

Le Grande [1963] found that the slack time rule had the smallest standard deviation of the completion time distribution and percent of late orders.

Composite Rules

Composite rules can be formed by taking linear combinations of other simple rules. The idea is to utilize the best aspects of a rule and try to dampen the less desirable features.

Conway, Maxwell, and Oldziey [n.d.] formulated a composite rule using Slack Rule, Shortest Processing Time Rule, and a Job Status Rule that considered the loading of the next machine group for a job. This rule can be expressed as

$$Z_{i,j}(t) = (O_{i,j} - P_{i,j}) + b \cdot \left(\sum P_{i,j} \right)^r + h \cdot Wnq,$$

where:

$Z_{k,j}(t)$ is the sequencing index of the j th operation of the i th job at a point in time, t , where the $Z_{i,j}(t)$'s for all operations are considered in increasing order;

i is the particular job being considered,

j is the operation being considered,

$O_{i,j}$ is the due date for the j th operation on the i th job,

$P_{i,j}$ is the process time for the j th operation of the i th job,

b is an arbitrary number that varies the weight given to process time,

$\sum P_{i,j}$ is the sum of the process time for all jobs in the queue for the j th operation,

r is an arbitrary exponential weighting factor that varies the weight given to $P_{i,j}$.

h is an arbitrary number that varies the weight given to the relative size of the queue that job i will visit next, and

W_{ij} is the relative size of the next queue to which job i will move. The relative size is the sum of the process times in this queue divided by the sum of the process times of all jobs in all queues in the shop.

Moodie and Roberts [1968] developed a composite rule using a weighted objective. It is stated as:

$$D_{j,k} = \sum_{i=1}^N W_i R_{i,j,k}$$

subject to:

$$\sum_{i=1}^n W_i = 1$$

where:

$D_{j,k}$ = priority index of j on machine k ,

$R_{i,j,k}$ = ratio of the "best" value for rule i divided by rule i value for job j on machine k ,

W_i = weighting of the i th rule, and

n = the number of rules combined.

In their study, this rule along with the slack and shortest processing time rules dominated all others tested. They found that the composite rule showed remarkable consistency for changing shop loads while the individual rules did not display this consistency.

Heuristic Scheduling

Heuristic Scheduling may provide a new and wider framework for attacking scheduling problems. Holloway and Nelson [1971] have done research in the area of heuristic scheduling for a job shop with due dates. The basic elements for heuristically scheduling any machine are as follows:

1) A set of six constraints on scheduled start times for each operation. These semi-global constraints are a function of the preceding operation and whether they are firm or flexible.

2) A hierarchical class structure that determines the order in which sets

of constraints are relaxed during the scheduling process.

3) Modes specifying which classes are available for scheduling.

4) A backup procedure to limit the search for a schedule within a given mode.

5) Priority rules to schedule a particular operation in a given class. [Holloway and Nelson, 1971]

Good results were obtained in using these elements to create a heuristic scheduler. But the solution does not seem amenable to pipeline processor scheduling, as it was complex and took a good deal of computing power.

Appendix:

EXAMPLES OF PRIORITY RULES, FROM DAY AND HOTTENSTEIN [1970]

In order to discuss the calculation of job priority values (each of which is determined for the machine in question) the following notation is introduced:

a = dummy variable.

$A_{j,k}$ = the arrival time of job j at the kth machine specified on its job route.

C_i = a constant ($i = 1, 2, \dots, 12$).

D_j = the due date of job j.

E_j = the arrival time of job j in the shop.

i = a dummy subscript.

j = job number.

k = the kth machine specified in a given job's routing.

N_j = the nth (last) machine specified in the job route of job j.

$p_{j,k}$ = the processing time of job j on the kth machine specified in the job route of job j.

$\pi_{j,k}$ = the priority of job j at the kth machine specified in the job route of job j.

$s_{j,k}$ = the set-up time of job j on the kth machine specified in the job route of job j.

t = the time at which selection from a queue is to be made.

The examples are presented in outline form:

A. Lateness rules

1. Inversely relating priority value to the due date of the job. The job with the earliest due date receives the highest priority.

$$\pi_{j,k} = \frac{C_i}{D_j}.$$

2. Inversely relating priority value to the remaining slack time. Highest priority is given to the job for which the time remaining to due date less the remaining processing time is a minimum.

$$\pi_{j,k} = \frac{C_2}{D_j - \left(t + \sum_{i=k}^{n_j} P_{j,i} \right)}.$$

3. Relating priority value directly to the remaining slack time.

Highest priority is given to the job for which the time remaining to due date less the remaining processing time is a maximum.

$$\pi_{j,k} = D_j - \left(t + \sum_{i=k}^{n_j} P_{j,i} \right).$$

4. Relating priority value directly to the ratio whose numerator

is the slack time and whose denominator is the remaining processing time of the job.

Highest priority is given to the job waiting in queue for which this ratio is smallest.

$$\pi_{j,k} = \frac{D_j - \left(t + \sum_{i=k}^{n_j} P_{j,i} \right)}{\sum_{i=k}^{n_j} P_{j,i}}.$$

5. Relating priority value directly to the slack time per remaining

operation. Highest priority is assigned to the job with the smallest ratio of slack time to the number of operations remaining.

$$\pi_{j,k} = \frac{D_j - \left(t + \sum_{i=k}^{n_j} P_{j,i} \right)}{n_j - \sum_{a=1}^k 1}.$$

B. Arrival order rules

1. Assigning priority on a first-come, first-served (FCFS) basis.

The first arrival in the queue of the machine receives the highest priority: priority values are assigned to jobs in a strictly decreasing sequence as they arrive in the particular queue.

$$\pi_{j,k} = \frac{C_3}{A_{j,k}}.$$

2. Assigning priority on the basis of arrival time in the shop. Of the jobs waiting for service at a machine, highest priority would be assigned to the job with the earliest shop arrival date.

$$\pi_{j,k} = \frac{C_4}{E_j}.$$

C. Rules that determine priority according to some property of the job itself

1. Inversely relating priority value to the processing time of the coming operation. Highest priority is assigned to the job with the shortest processing time on the machine in question. This rule, as mentioned previously, is synonymous with the SPT rule if set-up times are not considered.

$$\pi_{j,k} = \frac{C_5}{P_{j,k}}.$$

2. Relating priority directly to the processing time on the coming operation. Highest priority is given to the job with the longest operation on the machine in question.

$$\pi_{j,k} = C_6 P_{j,k}.$$

3. Inversely relating priority value to the total remaining processing time. Highest priority is given to the job for which the sum of the processing times for all remaining operations is a minimum.

$$\pi_{j,k} = \frac{C_7}{\sum_{i=k}^{n_j} P_{j,i}}.$$

4. Relating priority directly to the total remaining processing time. Highest priority is assigned to the job for which the sum of the processing time for all remaining operations is a maximum.

$$\pi_{j,k} = C_8 \sum_{i=k}^{ni} P_{j,i}.$$

5. Relating priority value directly to the total processing time of all operations on the job route. Highest priority is assigned to the job for which the total processing time of all operations on the job route is a maximum.

$$\pi_{j,k} = C_9 \sum_{i=1}^{ni} P_{j,i}.$$

6. Assigning priority on the basis of the ratio of processing time to the total work content of the job. Highest priority is assigned to the job that has the smallest ratio of processing time (on the machine in question) to the total processing time of all operations on the job route.

$$\pi_{j,k} = \frac{P_{j,k}}{\sum_{i=1}^{ni} P_{j,i}}.$$

7. Relating priority to the queue length of the next machine defined on the job route. Highest priority is given to the particular job that, upon leaving this machine, will go on to the next machine that has the shortest (in the sense of least processing time) critical queue. FCFS is used if no queue is critical. A queue is critical if it has less than a specified number of time units of processing time waiting. Actually, there is a family of rules of this type that can be parameterized by Q , a value below which a queue becomes critical.

8. Relating priority to the work content of the queue of the next machine defined on the job route. Highest priority is assigned to the particular job that will go on for its next operation to the queue with the least work, work being defined as the sum of the imminent processing times in that queue.

9. Inversely relating priority value to the number of remaining operations. Highest priority is assigned to the job with the fewest remaining operations.

$$\pi_{j,k} = \frac{C_{10}}{nj - \sum_{a=1}^k 1}.$$

10. Relating priority directly to the number of remaining operations. Highest priority is assigned to the job with the most remaining operations.

$$\pi_{j,k} = C_{11} \left(n_j - \sum_{a=1}^k 1 \right).$$

11. Letting priority depend upon the (dollar) value of the job. Jobs are divided into high-value and low-value classes, greater priority being assigned to the high-value class of jobs. Within each class, the FCFS rule is observed. Actually, there is a family of rules of this type that can be parameterized by X , the proportion of jobs assigned to the high-value class.

12. Relating priority value directly to the (dollar) value of the job. The priority is taken to be equal to the value of the job.

$$\pi_{j,k} = V_j.$$

13. Assigning priority according to the ratio of delay cost to processing time. Investigated in detail by Carroll [Day and Hottenstein, 1970], this family of priority rules is referred to as COVERT (cost over time) rules. Highest priority is given to the job for which the ratio is smallest.

14. Assigning priority on the basis of the earliest operation due date. Equally spaced due dates are assigned to each operation on the job route at the time the job enters the shop. Highest priority is assigned to the job with the earliest operation due date.

$$\pi_{j,k} = A_{j,1} + (D_j - A_{j,1}) \frac{K}{n_j}.$$

15. Assigning priority on the basis of set-up time. Highest priority is given to the job that requires the least amount of set-up time, relative to the job that has been completed on the machine in question.

$$\pi_{j,k} = \frac{C_{12}}{S_{j,k}}.$$

Bibliography

- Biegel, J. E. Production Control. Prentice Hall, Englewood Cliffs, New Jersey, 1971.
- Bulkin, M. H., J. L. Colley and H. W. Steinhoff, "Load Forecasting, Priority Sequencing, and Simulation in a Job Shop Control System," Management Science 10, No. 2, B-29-B-51 (October, 1966).
- Conway, R. W. An Experimental Investigation of Priority Assignment in a Job Shop. Memorandum RM-3789-PR, The RAND Corporation, Santa Monica, California (February 1964).
- Conway, R. W., "Priority Dispatching and Job Lateness in a Job Shop," J. Industrial Eng. 16, No. 4, 228-237 (July-August, 1965).
- Conway, R. W., B. M. Johnson and W. L. Maxwell, "An Experimental Investigation of Priority Dispatching," J. Industrial Eng. 11, No. 3, 221-229 (May-June, 1960).
- Conway, R. W. and W. L. Maxwell, "Network Scheduling by the Shortest Operation Discipline," Operations Research 10, No. 1, 51-73 (February 1962).
- Conway, R. W., W. L. Maxwell and J. W. Oldziey. Sequencing Against Due Dates. Technical Report No. 6, College of Engineering, Cornell University, N. Y. (n.d.), pp. 1-29.
- Dáy, J. E. and M. P. Hottenstein, "Review of Sequencing Research," Naval and Research Logistics Quarterly 17, No. 1, 11-39 (March 1970).
- Gentry, D. H. An Analysis of the Effects of Selected Scheduling Rules in a Parallel Processor Shop. A. E. C. Report Y-1848 (Category UC-39), Union Carbide Corp., Oak Ridge, Tennessee, Y-12 Plant (25 August 1972).
- Holloway, C. A. and R. T. Nelson, A New Approach to Job Shop Scheduling with Due Dates. Research Paper No. 11, Graduate School of Business, Stanford University (April, 1971).
- Jackson, J. R., "Networks of Waiting Lines," Operations Research 5, 518-521 (August 1957).
- LeGrande, E. L., "The Development of a Factory Simulation System Using Actual Operating Data," Management Technology 3, No. 1, 1-19 (May 1963).
- Maxwell, W. L. and N. Mehra, "Multiple-Factor Rules for Sequencing with Assembly Constraints," Naval Research Logistics Quarterly 15, No. 2, 241-254 (June 1968).
- Moodie, C. L. and S. D. Roberts, "Experiments with Priority Dispatching Rules in a Parallel Processor Shop," International J. Production Research 6, No. 4 303 (1968).
- Morse, P. M. Queues, Inventories, and Maintenance. John Wiley and Sons., Inc., New York, 1958.