Application of GPSS/360 to Job Shop Scheduling

by

Ronald A. Schwarz[1] and Thomas J. Schriber[2]

SUMMARY OF THE PAPER

Pertinent characteristics of the job shop scheduling problem are reviewed. Two GPSS/360 models assessing a total of eight alternative rules for constructing job shop schedules are presented. These rules have been studied by earlier investigators; hence this is a state-of-the-art paper illustrating the approach to an old problem using a relatively new capability, GPSS/360. Selected model output is exhibited, results are summarized, and assembly and execution times are indicated. Several references to appropriate earlier investigations are included. The models shown here are remarkably compact in comparison with earlier simulation models used to investigate the same or similar scheduling rules.

DESCRIPTION OF THE PROBLEM

A "job" is a piece of work whose completion requires that a series of operations be performed on a specified sequence of machines. A job shop is a shop in which the work is initiated and brought to completion. In general, more than one job is being worked on at a time; a stream of jobs is usually moving through the shop at all times. No two jobs are the same, except by chance, i.e. the series of operations and the sequence in which they are to be performed varies from job to job.

The problem lies in scheduling the jobs so that all due dates are met or that total lateness time is minimized. Construction of a schedule involves using a rule to resolve the conflict resulting when two or more jobs simultaneously compete for a given machine. Effectiveness of alternative rules can be assessed by constructing models simulating shop operation, then comparing the schedules which result when the models are computer-implemented.

Some of the rules that might be used to resolve conflicts that arise in constructing schedules are:

1) Job Slack: The job which has the least job slack gets the machine next. Job slack is defined as the due date, minus the current time, minus the total operation time remaining for the job.

2) Job Slack per Operation: The job which has the least job slack per operation gets the machine next. Job slack per operation is job slack divided by the remaining number of operations for that job.

3) Job Slack Ratio: The job which has the least job slack ratio gets the machine next. Job slack ratio is job slack divided by the time remaining until the job's due date.

4) Modified Job Slack Ratio: The job with the smallest modified job slack ratio is put on the machine next. The job slack ratio is modified by a term which takes machine loading into account to estimate the delay associated with each operation.

5) Shortest Imminent Operation: The job which will tie up the machine for the least amount of time gets the machine next.

6) Shortest Imminent Operation with Critical Job Slack Override: This rule results when the Shortest Imminent Operation rule is subject to temporary suspension if a job's slack falls beneath some arbitrarily-defined critical level.

7) First-come, first-served

8) Random

The above set of rules provides varied representation both in time-dependent characteristics and in the scope of information required. Hence, for Rules 5,7, and 8 the priority enjoyed by a job does not depend on the current time, whereas the other rules make use of the clock in assigning priorities. And, although most of the rules use local information, one (Rule 4) is included which relies on global considerations. A further observation is that each of the rules is inflexible except for Rule 6, which depends on operating experience in that the modeler specifies the threshold at which a job's slack becomes critical, causing the otherwise normal Shortest Imminent Operation rule to be subverted at the machine in question. Finally, the last two rules provide benchmarks of a sort against which results produced by the other six rules can be gauged.

There are two job shop problem modes:

1) The static problem is said to result when all jobs are on hand at time zero.

2) The dynamic problem results when some jobs are on hand at time zero but new jobs are admitted to the shop from time to time.

[1]Federal City College, Washington, D. C. 20001.

[2]The University of Michigan, Ann Arbor, Michigan 48104.

The information necessary for processing jobs is contained in a job file. A job file is a record of the jobs on hand, the sequence in which the jobs are to visit the various machines, and the operation times involved on the machines.

FEATURES OF THE MODELS PRESENTED HERE

Two models for the static problem are documented and discussed here. The first model implements only one scheduling rule, Job Slack per Operation, and is instructive in that it reveals the spirit of GPSS/360, illustrates some of the capabilities of the language, and displays the particulars of the scheduling problem. The second model simultaneously assesses the effects of the eight rules listed above by providing eight job shops operating in parallel, each of which processes the same static job file according to one of the eight rules. The second model also provides for random generation of the job files. Model output includes machine utilizations and the hours early or late that each job was completed according to each rule. Job Completion Tables showing the mean, standard deviation, and distribution of job earliness or lateness hours for each rule are also produced.

Rather than providing its own randomly-generated job files, the first model works with job files provided as model inputs. The first model can then be used to process various files generated by the second model, providing a limited check on the validity of the more complicated model. Although not presented here, other one-rule models were also used to investigate the validity of the second model.

The number of jobs per job file and the number of machine groups in the job shop are variables in both models. It is arbitrarily assumed in the job-file generation phase of the second model that each machine is visited exactly one time by each job. The order in which the machines are visited is random. Operation time for each job on each machine is determined by sampling from an operation time population uniformly distributed over the interval of integers from 5 to 15, inclusive.

Job due date is established by multiplying the job's total operation time by two factors: 1) an experimentally-determined factor depending on both the number of jobs in the file and the number of machines in the shop, and 2) a factor drawn randomly from the closed interval between 1.00 and 1.15. The first factor is designed so that each job will have a reasonable chance of being completed by the due date. The model itself was used to determine what the factor should be. The second factor is introduced to provide some variation between due date and required operation time from job to job.

Other conditions attached to the models are: 1) There is only one machine per machine group, 2) Each operation, once started, must be performed to completion, 3) Assembly operations are not allowed, 4) Lap-phasing is not permitted, 5) Operation times, including set-up times, are sequence-independent, and 6) Machines do not break down.

Documentation for each model includes: 1) A Table of Definitions, in which the model interpretation put on various GPSS entities is described, and 2) a block diagram version of the model, displaying the logic involved in implementing the scheduling rules and featuring explanatory comments adjacent to each block. A listing of the punch-cards corresponding to the second model is also shown. Typical output from the two models is exhibited, and results from all runs are compactly summarized. The various figures and tables have explanatory comments provided directly beneath them to aid in their interpretation whenever this is deemed necessary.

The second model has been used to process job files corresponding to all combinations of 6, 9, 12, and 15 jobs requiring 6, 9, 12, and 15 machines. The ability of the various rules to produce non-identical results increases as the complexity of the job shop increases. Figure 7 illustrates that in non-complex shops, many "ties" occur; the number of ties decreases sharply with increasing system complexity. As indicated in Figure 8, Rule 5 is superior in all cases. Its superiority increases as system complexity increases. Rule 6 is next best after Rule 5. Then come Rules 1 and 8 with about equal effectiveness. Finally, Rules 2, 3, 4, and 7 have about equal effectiveness and are least desirable. These results summarize a total of 80 simulations (each of the four different total number of jobs was simulated five times for each of the four different total number of machines in the shop). Assembly time for the model was 4.9 seconds. Total execution time for the 80 simulations described was 13 minutes, 7 seconds. The runs were made on the University of Michigan 360/67 under the Michigan Terminal System (MTS) monitor.

REFERENCES TO EARLIER WORK

Choice of the eight rules described above was directly motivated by the work of Gere.[3,4] He investigated all eight rules, processing a total of 25 static files involving 6 to 20 jobs, 1 to 16 operations per job, and 4 to 16 machines. He concluded that "the shortest imminent operation rule is less effective than a job-slack-based rule." This conclusion is not consistent with that of the present investigation. Conway, however, points out that "the performance of a shop with respect to meeting its due dates is a function not only of the sequencing rule employed but also of the method used to assign the due dates to the jobs."[5]

[3] Gere, William S., Jr., "Heuristics in Job Shop Scheduling", Management Science, Vol. 13, No.3, November, 1966, pp. 167-190.

[4] Gere, William S. Jr., "A Heuristic Approach to Job Shop Scheduling", Ph,D. Dissertation, Carnegie Institute of Technology, September, 1962.

[5] Moore, J. M., and Wilson, R. C., "A Review of Simulation in Job Shop Scheduling", Production and Inventory Management; January, 1967, pp. 1 - 10.

In particular, "for minimizing the number of jobs that are late, the shortest processing time rule was superior to the slack per operation when due dates were assigned" according to constant lead time or randomly.[5] Furthermore, "Conway also observed that under heavier loads, the shortest processing time rule appeared to be better than the slack per operation rule with respect to the number of jobs late."[5] Hence the results of the present investigation are not without support.

Execution time data for Gere's work, which was carried out in FORTRAN, is not available in References 3 or 4. Reference 4 contains some 20 pages of logical flowcharts in an Appendix. This suggests the complexity of the problem using a procedure-oriented approach. Gere did consider both the static and dynamic problems, however, and made provision for investigation of several heuristics in addition to the eight rules used for the present investigation.

The reader is referred to Reference 5 for a fairly recent summary of simulation research in job shop scheduling and for a bibliography.

GPSS Entity	Interpretation
Transaction	A job P1: Not used P2: Job number P3: Number of remaining operations P4: Imminent machine number P5: Imminent operation time P6: Job's "Job Slack per Operation" P7: Column Index pointing to imminent machine number in Job Information Matrix P8: Column Index pointing to imminent operation time in Job Information Matrix
Facility j	Machine j j = 1,2,3,...,MTOT, where MTOT represents the number of machines in the shop
Matrix 1 (Halfword)	Job Information Matrix Rows 1,2,3,...,JTOT carry information about jobs 1,2,3,...,JTOT, respectively, where JTOT represents the number of jobs Column — Information 1 — Number of operations 2 — Total operation time 3 — Due date 4, 5, :, 3+MTOT — Numbers of the machines, appearing in the sequence in which the job requires them 4+MTOT, 5+MTOT, :, 3+2*MTOT — Operation times on the machines in columns 4,5,...,3+MTOT, respectively
Matrix 2 (Halfword)	Starting Time and Completion Time Matrix MH2(i,j) is the time job i went onto machine j, for i = 1,2,3,...,JTOT, and j = 1,2,3,...,MTOT MH2(i,MTOT+1) is the time job i was completed MH2(i,MTOT+2) is the hours early or late that job i was completed
Savevalue JTOT (Halfword)	Total number of jobs
Savevalue MTOT (Halfword)	Total number of machines
Table DISTN	Job Completion Table
Variable HOURS	Earliness hours or lateness hours relative to job completion; a negative value means that the job was completed late
Variable JOBS	Total jobs in shop minus one; (used as A operand in SPLIT block)
Variable JSPO	Job Slack per Operation
Variable MACP1	Total machines in shop plus one
Variable MACP2	Total machines in shop plus two
Variable SHFT1	Total machines in shop plus three
Variable SLACK	Job slack
Variable TIME	Current value of clock minus one

Table 1. Definitions for "Job Slack per Operation" Model

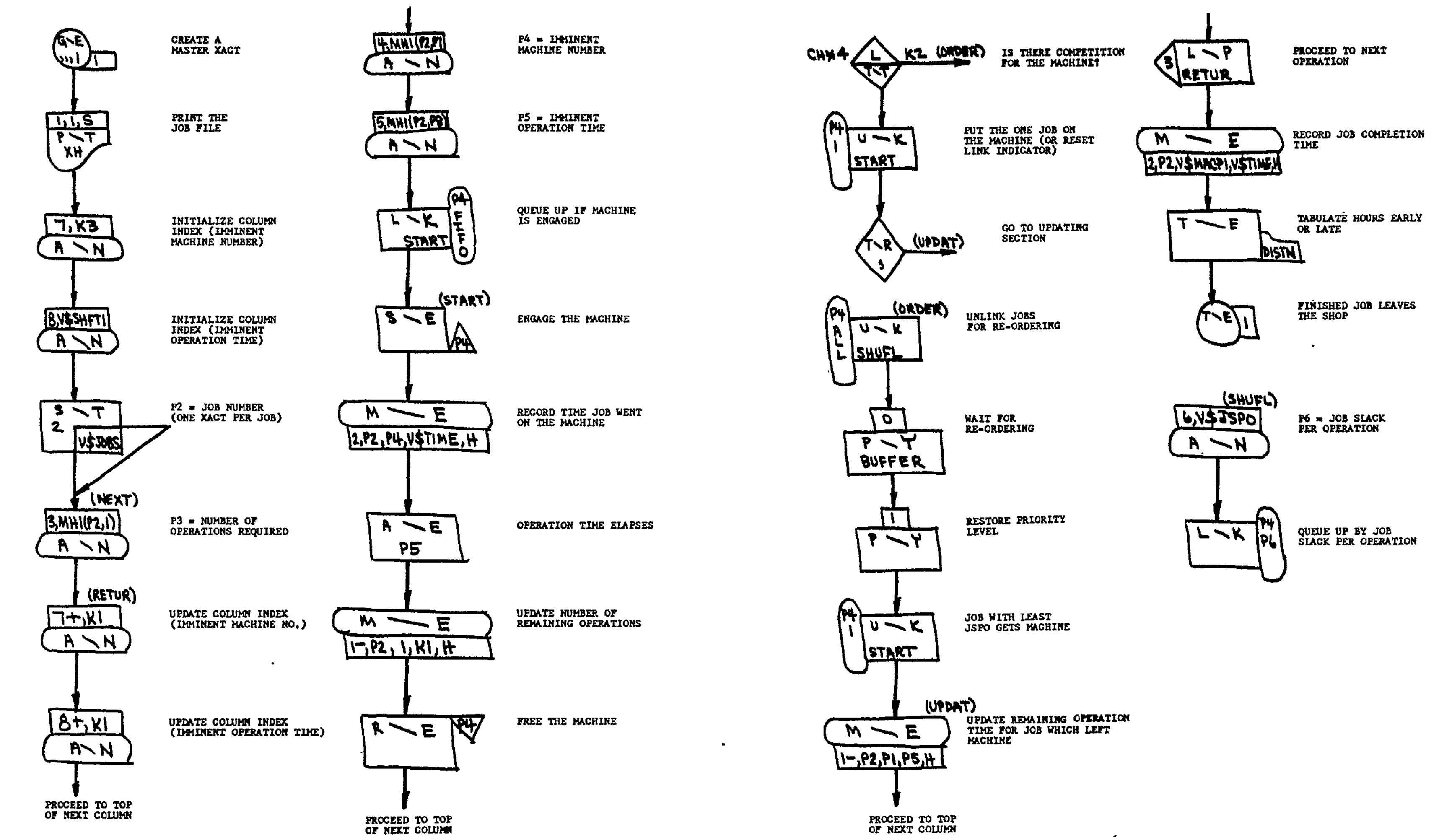

Figure 2. Block Diagram for "Job Slack per Operation" Model

MATRIX HALFWORD SAVEVALUE 1

	COL. 1	2	3	4	5	6	7	8	9	10	11
ROW 1	4	35	84	1	4	2	3	11	7	10	7
2	4	39	90	4	1	2	3	12	5	13	9
3	4	23	53	4	3	1	2	6	6	6	5
4	4	41	89	3	4	2	1	6	11	10	14
5	4	40	97	2	1	3	4	8	9	9	14
6	4	41	93	2	3	4	1	10	13	10	8
7	4	35	82	1	3	4	2	7	11	9	8
8	4	46	103	3	4	1	2	7	12	13	14
9	4	33	77	4	1	2	3	8	9	8	8
10	4	42	103	2	1	4	3	11	5	14	12

Figure 1. A Typical Job Information Matrix

Rows 1,2,3,...,10 carry information about jobs 1,2,3,...,10, respectively.

Column 1: Number of operations
Column 2: Total operation time
Column 3: Due date
Columns 4--7: Numbers of the machines, appearing in the sequence in which the job requires them
Columns 8--11: Operation times on the machines in Columns 4--7, respectively

MATRIX HALFWORD SAVEVALUE 2

	COL. 1	2	3	4	5	6
ROW 1	0	59	69	37	76	8
2	36	69	82	0	91	-1
3	41	54	31	12	59	-6
4	54	44	0	26	68	21
5	18	0	48	89	103	-6
6	81	8	18	65	89	4
7	11	82	37	56	90	-8
8	68	90	6	44	104	-1
9	27	36	57	18	65	12
10	47	18	91	75	103	0

Figure 3A. Starting Time and Completion Time Matrix

(Sample Job-Slack-per-Operation Model Output)

MH2(i,j) is the time job i went onto machine j, for i = 1,2,3,...,10, and j = 1,2,3,4.

MH2(i,5) is the time job i was completed, for i = 1,2,3,...,10.

MH2(i,6) is the hours early or late that job i was completed, for i = 1,2,3,...,10. Negative entries mean late completions.

FACILITY	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRAN
1	.828	10	8.699
2	.923	10	9.699
3	.838	10	8.799
4	.980	10	10.299

Figure 3B. Table of Machine Utilizations

(Sample Job-Slack-per-Operation Model Output)

Average Utilization of Facility j is equal to the utilization of Machine j, j = 1,2,3, and 4.

TABLE DISTN

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION
10	2.299	9.128

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER
-20	0	.00	.0	100.0
-15	0	.00	.0	100.0
-10	0	.00	.0	100.0
-5	3	29.99	29.9	70.0
0	3	29.99	59.9	40.0
5	1	9.99	69.9	30.0
10	1	9.99	79.9	20.0
15	1	9.99	89.9	10.0
20	0	.00	89.9	10.0
OVERFLOW	1	9.99	100.0	.0
AVERAGE VALUE OF OVERFLOW		21.00		

Figure 3C. Job Completion Table

(Sample Job-Slack-per-Operation Model Output)

The Job Completion Table records the distribution of job earliness or lateness times showing mean, standard deviation, and relative frequency.

GPSS Entity	Interpretation
Transaction	A job

P1: Number of the rule used to determine queue discipline

Value	Rule
1	Job Slack
2	Job Slack per Operation
3	Job Slack Ratio
4	Modified Job Slack Ratio
5	Shortest Imminent Operation
6	Shortest Imminent Operation (with Critical Job Slack Override)
7	First come, first served
8	Random

P2: Job number
P3: Number of remaining operations
P4: Imminent machine number
P5: Imminent operation time
P6: Numeric value of dispatching criterion
P7: Column Index pointing to imminent machine number in Job Information Matrix
P8: Column Index pointing to imminent operation time in Job Information Matrix
P9: Imminent machine number (used only in randomly generating the job file)
P10: Imminent operation time (used only in randomly generating the job file)

Facility j — Machine j

j = 4*(Rule Number - 1) + i, where

Rule Number = 1,2,...,8 (see P1 above),
i = 1,2,...,MTOT, and MTOT is the number of machines in the shop

Example: In a shop consisting of four machines, the Job Slack shop consists of machines 1,2,3, and 4; the Job Slack per Operation shop consists of machines 5,6,7, and 8, etc.

Function FAKTR — Used in sampling from the due date distribution to randomly determine a job's due date

Function MTIME — Used in sampling from the operation time distribution to randomly determine operation time for a particular job on a particular machine

Function QDSCP — Function which returns the numeric value of the dispatching criterion

Table 2. Definitions for "Parallel Simulation of Eight Rules" Model

Matrix 1 (Halfword) — Job Information Matrix

Rows 1,2,3,...,JTOT carry information about jobs 1,2,3,...,JTOT, respectively, where JTOT represents the number of jobs

Column	
1	Number of operations
2	Total operation time
3	Due date
4, 5, :, 3+MTOT	Numbers of the machines, appearing in sequence in which the job requires them
4+MTOT, 5+MTOT, :, 3+2*MTOT	Operation times on the machines in columns 4,5,...,3+MTOT, respectively

Matrix 2 (Halfword) — Remaining Operation Time Matrix

MH2(i,j) is the remaining operation for job i being processed according to Rule j, where

i = 1,2,3,...,JTOT and

j = 1,2,3,...,8

Matrix 3 (Halfword) — Machine Loading Matrix

MH3(i,1) is the remaining hours of operation required on machine i, for i = 1,2,...,MTOT. Machine loading information is required only for Rule 4 (Modified Job Slack Ratio).

Matrix 4 (Halfword) — Completion Time Matrix

MH 4(i,j) is the hours early or late that job i was completed in the shop dispatching according to Rule j, for i = 1,2,...,JTOT, and j = 1,2,3,...,8. Negative entries mean late completions.

MH4(JTOT+1,j) is the algebraic sum of earliness and lateness hours in the shop dispatching according to rule j, j = 1,2,3,...,8.

Savevalue JTOT (Halfword) — Total number of jobs

Savevalue MTOT (Halfword) — Total number of machines

Table j — Job Completion Table for Rule j

j = 1,2,3,...,8

The Job Completion Table records the distribution of job earliness or lateness hours.

Table 2. Definitions for "Parallel Simulation of Eight Rules" Model
(continued on next page)

Variable DUE	Used in randomly determining a job's due date as a function of job's required operation time and number of jobs in the shop
Variable GOBAK	Used in random dispatching (Rule 8) to determine how many jobs to move from front to back of the queue before coming to the job that takes the machine
Variable HOURS	Earliness hours or lateness hours; negative values imply late completions
Variable JNUM	Used to number the jobs in a shop
Variable JOBS	Total jobs in shop minus one
Variable JSPO	Job Slack per Operation (Job slack hours divided by the remaining number of operations)
Variable JSR	Job Slack Ratio (Job slack hours divided by hours remaining until the due date)
Variable JTP1	Total jobs in shop plus one
Variable MCOL	The column in MH1 in which to enter the number of the machine a job visits next (used only in randomly generating the job file)
Variable MMONE	Total machines in shop minus one
Variable MNUM	Number assigned to a machine
Variable PAUSE	Time units that elapse while a particular Transaction waits for the job file to be generated
Variable SHFT1	Total machines in shop plus three
Variable SHFT2	Used to map Rule 4 machine numbers (12,13,14,...) into the sequence 1,2,3,...
Variable SLACK	Job Slack (Due date minus current time minus remaining operation time)
Variable SLKLO	Used in Rule 6 to compute the threshhold at which a job's slack becomes critical
Variable SPAN	Hours remaining until due date
Variable TCOL	The column in MH1 in which to enter the operation time on the machine a job visits next (used only in randomly generating the job file)

Table 2. Definitions for "Parallel Simulation of Eight Rules" Model

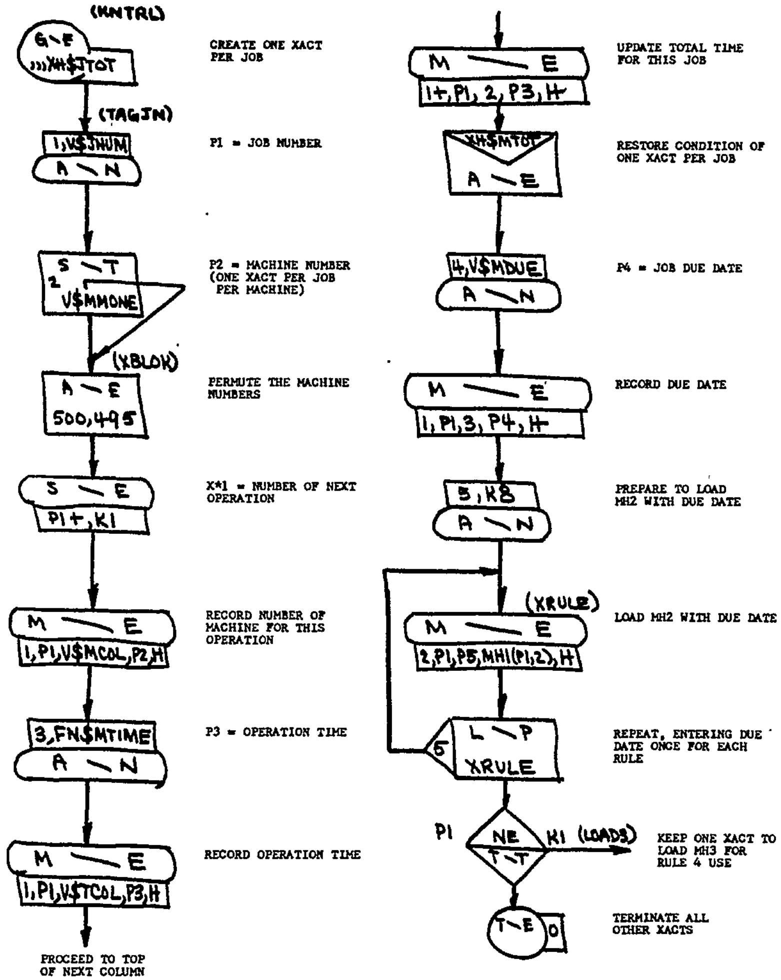

Figure 4. Block Diagram for "Parallel Simulation of Eight Rules" Model
(continued on next page)

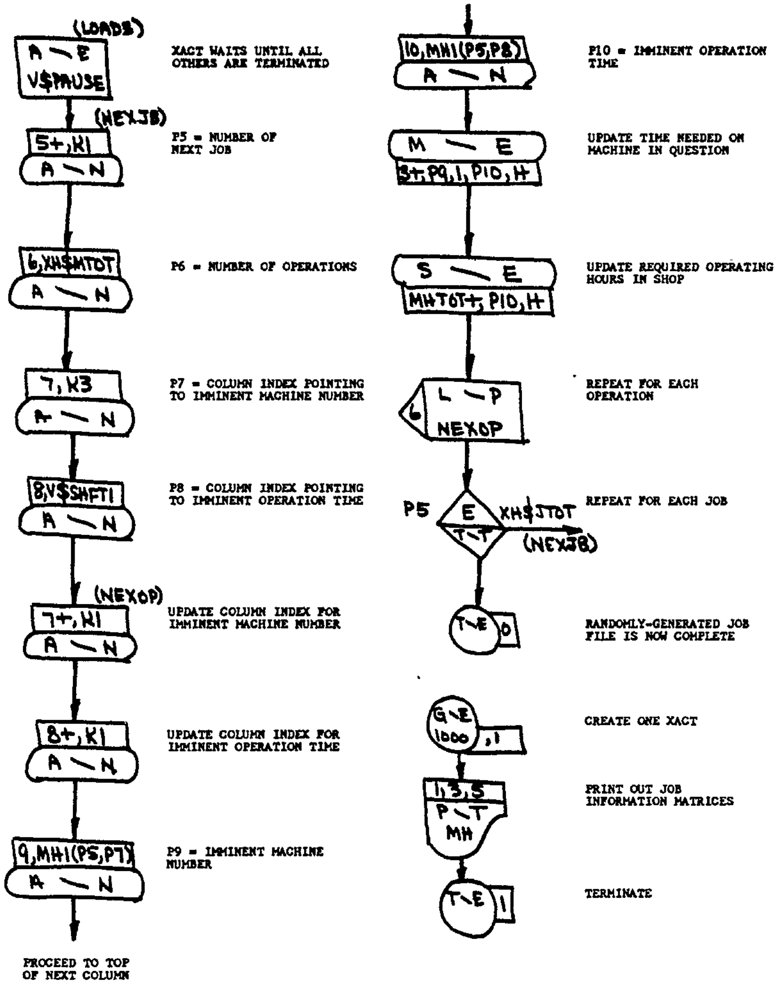

Figure 4. Block Diagram for "Parallel Simulation of Eight Rules" Model

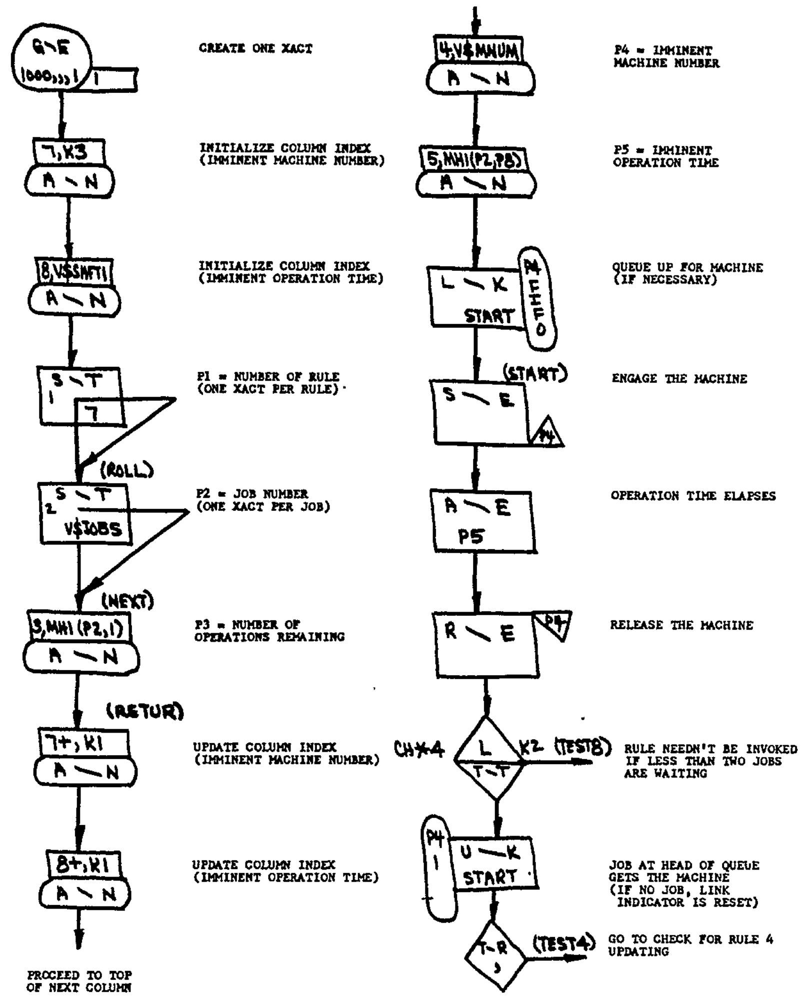

Figure 4. Block Diagram for "Parallel Simulation of Eight Rules" Model

(continued on next page)

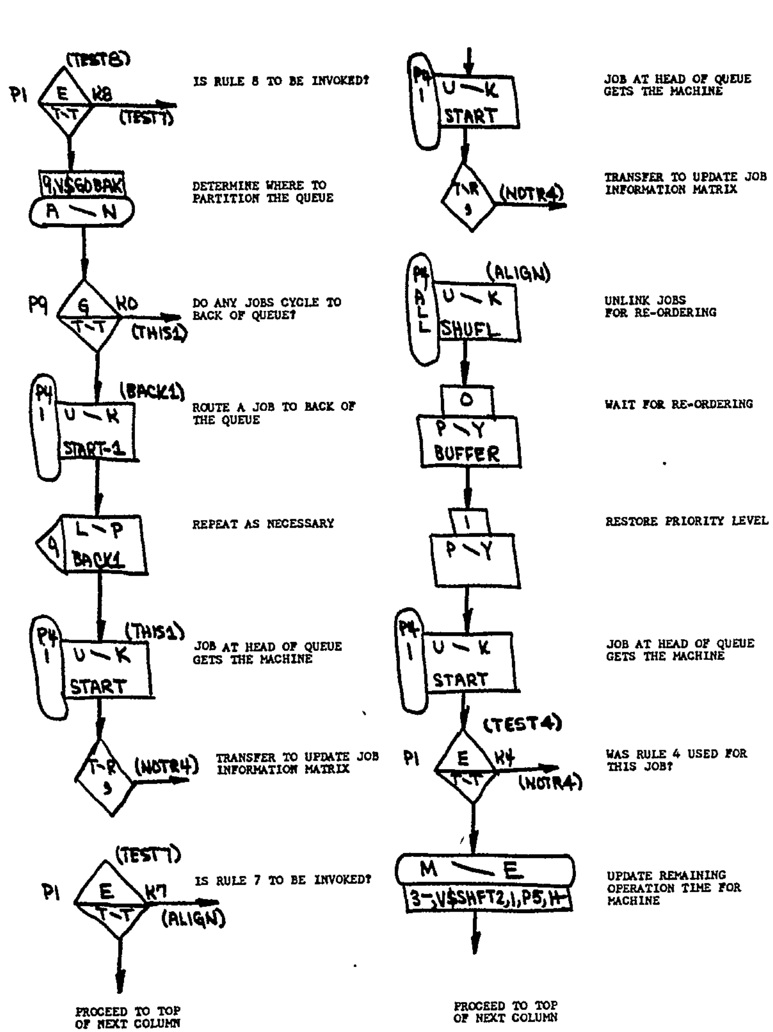

Figure 4. Block Diagram for Parallel Simulation of Eight Rules" Model

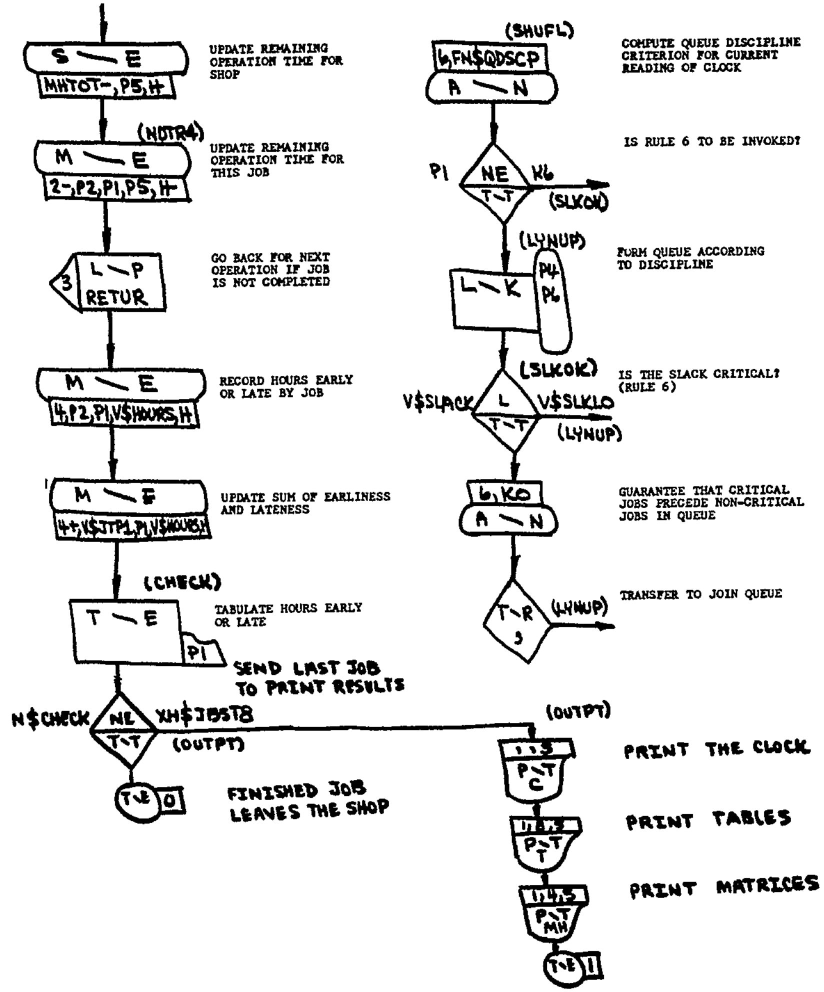

Figure 4. Block Diagram for "Parallel Simulation of Eight Rules" Model

```
         REALLOCATE HSV,6,CHA,230,GRP,1,BVR,1,FMS,1,HMS,4
         REALLOCATE BLO,90,STO,1,QUE,1,TAB,8
         REALLOCATE LOG,1,FUN,6,VAR,25,FSV,20
         REALLOCATE FAC,230,COM,18000,XAC,230
*         SIMULATE
FAKTR FUNCTION   RN2,C2
.100/1,115
MTIME FUNCTION   RN2,C2
.5/1,15
1        MATRIX     H,15,33
2        MATRIX     H,15,8
3        MATRIX     H,15,1
4        MATRIX     H,16,8
         INITIAL    XH$JTOT,15/XH$MTOT,15/XH$SCALE,146/XH$JBST8,120
         INITIAL    MH1(1-15,1),15
DUE      VARIABLE   MH1(P1,2)*XH$SCALE*FN$FAKTR/K10000
JNUM     VARIABLE   N$TAGJN+K1
MCOL     VARIABLE   X*1+K3
MMONE    VARIABLE   XH$MTOT-K1
PAUSE    VARIABLE   K999-C1
SHFT1    VARIABLE   XH$MTOT+K3
TCOL     VARIABLE   X*1+K3+XH$MTOT
KNTRL    GENERATE   ,,,XH$JTOT          CREATE ONE XACT PER JOB
TAGJN    ASSIGN     1,V$JNUM            P1 = JOB NUMBER
         SPLIT      V$MMONE,XBLOK,2     P2 = MACHINE NUMBER (ONE XACT PER MACH)
XBLOK    ADVANCE    500,495             PERMUTE THE MACHINE NUMBERS
         SAVEVALUE  P1+,K1              X*1 = NUMBER OF NEXT OPERATION
         MSAVEVALUE 1,P1,V$MCOL,P2,H    RECORD NO. OF MACHINE FOR THIS OPERATION
         ASSIGN     3,FN$MTIME          P3 = OPERATION TIME
         MSAVEVALUE 1,P1,V$TCOL,P3,H    RECORD OPERATION TIME
         MSAVEVALUE 1+,P1,2,P3,H        UPDATE TOTAL TIME FOR THIS JOB
         ASSEMBLE   XH$MTOT             RESTORE CONDITION OF ONE XACT PER JOB
         ASSIGN     4,V$DUE             P4 = JOB DUE DATE
         MSAVEVALUE 1,P1,3,P4,H         RECORD DUE DATE
         ASSIGN     5,K8                PREPARE TO LOAD MH2 WITH DUE DATES
XRULE    MSAVEVALUE 2,P1,P5,MH1(P1,2),H DUE DATE GOES INTO MH2
         LOOP       5,XRULE             ENTER DUE DATE ONCE FOR EACH RULE (1-6)
         TEST NE    P1,K1,LOAD3         ONE XACT LOADS MH3 FOR RULE 4 USE
         TERMINATE  0                   XACT WITH P1=K1 SETS UP REST OF FILE
LOAD3    ADVANCE    V$PAUSE             XACT WAITS UNTIL OTHERS TERMINATE
NEXJB    ASSIGN     5+,K1               P5 = NUMBER OF THE 'NEXT JOB'
         ASSIGN     6,XH$MTOT           P6 = NUMBER OF OPERATIONS
         ASSIGN     7,K3                P7 = COL INDEX POINTING TO MACH. NO.
         ASSIGN     8,V$SHFT1           P8 = COL INDEX POINTING TO MACH. TIME
NEXOP    ASSIGN     7+,K1               UPDATE COL INDEX FOR MACHINE NUMBER
         ASSIGN     8+,K1               UPDATE COL INDEX FOR OPERATION TIME
         ASSIGN     9,MH1(P5,P7)        P9 = 'NEXT MACHINE NUMBER'
         ASSIGN     10,MH1(P5,P8)       P10 = 'NEXT OPERATION TIME'
         MSAVEVALUE 3+,P9,1,P10,H       UPDATE TIME NEEDED ON THAT MACHINE
         SAVEVALUE  MHTOT+,P10,H        UPDATE MACHINE HOURS NEEDED IN SHOP
         LOOP       6,NEXOP             REPEAT FOR EACH OPERATION
         TEST E     P5,XH$JTOT,NEXJB    REPEAT FOR EACH JOB
         TERMINATE  0                   JOB FILE IS NOW COMPLETE
         GENERATE   1000,,,1            CREATE AN XACT WHEN JOB FILE IS DONE
         PRINT      1,3,MH,S            PRINT OUT THE JOB FILE
         TERMINATE  1                   PERMIT JOB FILE TO BE PROCESSED
SLACK    VARIABLE   MH1(P2,3)-MH2(P2,P1)-C1
JSPO     VARIABLE   V$SLACK/P3
JSR      VARIABLE   V$SLACK/(MH1(P2,3)-C1)
MJSR     FVARIABLE  K1-MH2(P2,4)*(K1+XH$MHTOT/MH3(V$SHFT2,1))/V$SPAN
GOBAK    VARIABLE   RN1aCH*4
HOURS    VARIABLE   MH1(P2,3)-C1
JOBS     VARIABLE   XH$JTOT-K1
JTP1     VARIABLE   XH$JTOT+K1
MNUM     VARIABLE   MH1(P2,P7)+XH$MTOT*(P1-K1)
SHFT2    VARIABLE   P4-XH$MTOT*K3
SLKLO    VARIABLE   (MH1(P2,3)-MH2(P2,6))*K3/K10
SPAN     VARIABLE   MH1(P2,3)-C1
QDSCP    FUNCTION   P1,M6
1,V8/2,V9/3,V10/4,V11/5,P5/6,P5
```

Figure 5. Program Listing for "Parallel Simulation of Eight Rules" Model

```
      1  TABLE      V$HOURS,-20,5,10
      2  TABLE      V$HOURS,-20,5,10
      3  TABLE      V$HOURS,-20,5,10
      4  TABLE      V$HOURS,-20,5,10
      5  TABLE      V$HOURS,-20,5,10
      6  TABLE      V$HOURS,-20,5,10
      7  TABLE      V$HOURS,-20,5,10
      8  TABLE      V$HOURS,-20,5,10
         GENERATE   1000,0,,1,1         CREATE ONE XACT
         ASSIGN     7,K3                INITIALIZE COL INDEX, IMMINENT MACH NO
         ASSIGN     8,V$SHFT1           INITIALIZE COL INDEX, IMMINENT OP TIME
         SPLIT      7,ROLL,1            P1 = RULE NUMBER (ONE XACT PER RULE)
ROLL     SPLIT      V$JOBS,NEXT,2       P2 = JOB NUMBER (ONE XACT PER JOB)
NEXT     ASSIGN     3,MH1(P2,1)         P3 = NUMBER OF OPERATIONS REMAINING
RETUR    ASSIGN     7+,K1               UPDATE COL INDEX, IMMINENT MACH NO
         ASSIGN     8+,K1               UPDATE COL INDEX, IMMINENT OP TIME
         ASSIGN     4,V$MNUM            P4 = IMMINENT MACHINE NUMBER
         ASSIGN     5,MH1(P2,P8)        P5 = IMMINENT OPERATION TIME
         LINK       P4,FIFO,START       QUEUE UP FOR MACHINE (IF NECESSARY)
START    SEIZE      P4                  ENGAGE THE MACHINE
         ADVANCE    P5,0                OPERATION TIME ELAPSES
         RELEASE    P4                  RELEASE THE MACHINE
         TEST L     CH*4,K2,TEST8       CH*4 < 2 => DISREGARD THE RULE
         UNLINK     P4,START,1          PUT JOB AT HEAD OF QUEUE ON THE MACHINE
         TRANSFER   ,TEST4              GO TO TEST FOR RULE 4
TEST8    TEST E     P1,K8,TEST7         IS RULE 8 TO BE INVOKED?
         ASSIGN     9,V$GOBAK           DETERMINE WHERE TO PARTITION THE QUEUE
         TEST G     P9,K0,THIS1         DO ANY JOBS GO BACK?
BACK1    UNLINK     P4,START-1,1        SEND A JOB TO THE BACK OF THE QUEUE
         LOOP       9,BACK1             REPEAT AS NECESSARY
THIS1    UNLINK     P4,START,1          PUT JOB AT HEAD OF QUEUE ON THE MACHINE
         TRANSFER   ,NOTR4              GO TO UPDATE JOB INFORMATION MATRIX
TEST7    TEST E     P1,K7,ALIGN         IS RULE 7 TO BE INVOKED?
         UNLINK     P4,START,1          PUT JOB AT HEAD OF QUEUE ON THE MACHINE
         TRANSFER   ,NOTR4              GO TO UPDATE JOB INFORMATION MATRIX
ALIGN    UNLINK     P4,SHUFL,ALL        REORDER THE QUEUE ACCORDING TO THE RULE
         PRIORITY   0,BUFFER            WAIT UNTIL THE REORDERING IS FINISHED
         PRIORITY   1                   RE-ESTABLISH NORMAL PRIORITY LEVEL
         UNLINK     P4,START,1          PUT JOB AT HEAD OF QUEUE ON THE MACHINE
TEST4    TEST E     P1,K4,NOTR4         WAS RULE 4 USED FOR THIS JOB?
         MSAVEVALUE 3-,V$SHFT2,1,P5,H   UPDATE REMAINING OP TIME FOR MACHINE (4)
         SAVEVALUE  MHTOT-,P5,H         UPDATE REMAINING OP TIME FOR SHOP (4)
NOTR4    MSAVEVALUE 2-,P2,P1,P5,H       UPDATE REMAINING OP TIME FOR THIS JOB
         LOOP       3,RETUR             GO BACK IF THERE ARE REMAINING OPS
         MSAVEVALUE 4,P2,P1,V$HOURS,H   RECORD HOURS EARLY OR LATE BY JOB
         MSAVEVALUE 4+,V$JTP1,P1,V$HOURS,H UPDATE SUM OF EARLINESS AND LATENESS
CHECK    TABULATE   P1                  TABULATE HOURS EARLY OR LATE
         TEST NE    N$CHECK,XH$JBST8,OUTPT
         TERMINATE                      FINISHED JOB LEAVES THE SHOP
OUTPT    PRINT      ,,C,S               PRINT CLOCK VALUES
         PRINT      1,8,T,S             PRINT JOB COMPLETION TABLES
         PRINT      1,4,MH,S            PRINT RUN AUDIT INFORMATION
         TERMINATE  1                   SHUT OFF THE RUN
SHUFL    ASSIGN     6,FN$QDSCP          UPDATE QUEUE DISCIPLINE CRITERION
         TEST NE    P1,K6,SLKOK         IS RULE 6 TO BE INVOKED?
LYNUP    LINK       P4,P6               FORM QUEUE ACCORDING TO DISCIPLINE
SLKOK    TEST L     V$SLACK,V$SLKLO,LYNUP IS THE SLACK CRITICAL? (RULE 6)
         ASSIGN     6,K0                P6 = 0 => JOB PRECEDES NON-CRITICAL JOBS
         TRANSFER   ,LYNUP              FORM QUEUE ACCORDING TO DISCIPLINE
         START      1,NP
         RESET
         START      1,NP
         CLEAR      XH$JTOT,XH$MTOT,XH$SCALE,XH$JBST8
         INITIAL    MH1(1-15,1),15
         START      1,NP
         RESET
         START      1,NP
         END
```

Figure 5. Program Listing for "Parallel Simulation of Eight Rules" Model

MATRIX HALFWORD SAVEVALUE 1

COL.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ROW 1	6	62	135	1	3	6	2	4	5	8	10	5	13	14	12
2	6	49	100	6	3	5	2	1	4	9	13	7	9	5	6
3	6	56	112	6	5	4	1	2	3	6	7	14	14	6	9
4	6	48	109	5	2	4	6	1	3	13	6	6	7	11	5
5	6	53	117	2	5	1	3	4	6	11	13	10	7	5	7
6	6	54	117	1	2	3	5	6	4	8	13	12	10	5	6
7	6	54	121	1	2	5	6	4	3	12	8	6	7	7	14
8	6	56	113	1	2	3	4	6	5	10	7	14	8	12	5
9	6	61	136	5	6	1	2	4	3	12	11	8	12	12	6
10	6	54	121	1	2	6	4	3	5	11	7	7	9	9	11
11	6	66	141	5	2	4	6	1	3	12	8	14	10	13	9
12	6	51	116	5	1	6	2	3	4	11	7	9	12	7	5

Figure 6A. Job Information Matrix

Rows 1,2,3,...,12 carry information about jobs 1,2,3,...,12, respectively.

Column 1:	Number of operations
Column 2:	Total operation time
Column 3:	Due date
Columns 4--9:	Numbers of the machines, appearing in the sequence in which the job requires them
Columns 10--15:	Operation times on the machines in columns 4- 9, respectively

MATRIX HALFWORD SAVEVALUE

COL.	1	2	3	4	5	6	7	8
ROW 1	16	27	32	32	11	15	32	43
2	-6	-11	-24	-42	28	17	-47	0
3	-6	-4	-28	-28	-8	1	-18	-11
4	23	-12	44	44	46	46	36	31
5	-8	-6	-30	-20	-13	-36	-15	-23
6	-7	-10	-13	-19	39	23	-12	37
7	-11	-14	-40	-33	-37	-34	-32	-46
8	17	17	-10	-17	23	23	-17	16
9	-11	-14	21	21	10	-4	15	4
10	14	2	-21	-21	9	20	-29	-66
11	0	-3	34	34	-28	-23	34	-5
12	-13	-5	-19	-31	33	28	-25	-42
13	8	-33	-54	-80	113	76	-78	-62

Figure 6B. Completion Time Matrix

(Sample Model Two Output for Figure 6A Job File)

MH4(i,j) is the hours early or late that job i was completed in the shop dispatching according to Rule j, for i = 1,2,3,...,12, and j = 1,2,3,...,8. Negative entries mean late completions.

MH4(13,j) is the algebraic sum of earliness and lateness hours in the shop dispatching according to Rule j, j = 1,2,3,. .,8.

TABLE 5

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION
12	9.416	26.375

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER
-20	2	16.66	16.6	83.3
-15	0	.00	16.6	83.3
-10	1	8.33	24.9	75.0
-5	1	8.33	33.3	66.6
0	0	.00	33.3	66.6
5	0	.00	33.3	66.6
10	2	16.66	49.9	50.0
15	1	8.33	58.3	41.6
20	0	.00	58.3	41.6
OVERFLOW	5	41.66	100.0	.0
AVERAGE VALUE OF OVERFLOW		33.79		

TABLE 6

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION
12	6.333	25.875

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER
-20	3	25.00	25.0	75.0
-15	0	.00	25.0	75.0
-10	0	.00	25.0	75.0
-5	0	.00	25.0	75.0
0	1	8.33	33.3	66.6
5	1	8.33	41.6	58.3
10	0	.00	41.6	58.3
15	1	8.33	49.9	50.0
20	2	16.66	66.6	33.3
OVERFLOW	4	33.33	100.0	.0
AVERAGE VALUE OF OVERFLOW		30.00		

Figure 6C. Job Completion Tables for Rules 5 and 6

(Sample Model Two Output for Figure 6A Job File)

The Job Completion Table records the distribution of job earliness or lateness times showing mean, standard deviation, and relative frequency.

		Machines			
		6	9	12	15
Jobs	6	11	5	5	9
	9	2	6	8	8
	12	0	3	1	3
	15	0	0	2	3

Figure 7. Table of Ties Encountered

		Rule							
		1	2	3	4	5	6	7	8
Jobs	6	4.1	5.025	4.625	4.725	3.525	4.625	4.725	4.65
	9	4.6	3.825	4.575	5.075	3.275	4.0	5.25	5.4
	12	4.875	5.825	4.775	5.25	1.8	4.05	5.375	4.05
	15	4.65	5.45	5.625	5.35	1.2	3.25	5.975	4.5

Figure 8. Table of Average Rank Attained by the Various Rules

For each different number of jobs, a total of 20 simulations was conducted (5 simulations each for the cases of 6, 9, 12, and 15 machines). Each table entry is the average rank attained by the various rules during the 20 simulations.