Apr 3rd, 12:00 AM

Simulation of a Large Scale Data Reduction System

Kingsley E. Forry
Operations Research Analyst, Chief Scientist's Office, RCA MTP, Patrick AFB, Florida

Scholarly Commons Citation
Forry, Kingsley E., "Simulation of a Large Scale Data Reduction System" (1967). *The Space Congress® Proceedings*. 7.
https://commons.erau.edu/space-congress-proceedings/proceedings-1967-4th/session-12/7

SIMULATION OF A LARGE SCALE DATA REDUCTION SYSTEM

Kingsley E. Forry
Operations Research Analyst
Chief Scientist's Office
RCA MTP
Patrick AFB, Florida

Introduction

Over the next few years, one of the great challenges mankind will be facing is the design and management of increasingly complex systems. Simulation models of these systems will play a major role in meeting this challenge expeditiously and economically. As an example of the use of such techniques this paper describes briefly a Monte Carlo simulation of the Air Force Eastern Test Range data reduction computer system. This system included an IBM 7094/7044/1301 configuration called POD for "Process on Demand." It was installed in mid-1964 replacing a 7094/1401 computer system. Additional components of the system included two remaining 1401 systems and a keypunch operation. The purpose was to analyze and predict the behavior of this system under a variety of conditions without interfering with its operation.

Indeed, experiments and analyses of the real system were prohibitive at that time, yet, data on the probable response of the system to increasing demands were vitally needed. Thus, some of the conditions explored with the model included:

(1) increase or decrease in workload

(2) changes in timing schedule of workload input

(3) changes in balance of kinds of workload

(4) addition or deletion or substitution of hardware

(5) software changes which will produce known changes in system parameters

(6) internal priority system changes

A few of the experiments conducted were to:

(1) determine effect of more programming code check upon the system

(2) determine optimum manner of submitting programming code check so as to minimize total average system turnaround

(3) determine effect of unevenness of input rate of production work into the system

(4) determine effect on system turnaround time of moving Management Information systems from POD to 1401

The new system inherently had the capability to significantly reduce the data reduction turnaround time. However, it appeared that the job turnaround time had not improved by the anticipated amount. Some portion of the additional turnaround was due to "bugs" in the system which would finally be worked out. It appeared, however, that there might also be some more fundamental problems.

Motivated by this concern, it was decided to make a detailed analysis of the workload and its flow through the system to identify bottlenecks and make recommendations for their removal.

Several things were immediately apparent:

(1) This problem was a waiting line type wherein jobs were either waiting for service or the computer was waiting for jobs.

(2) Because it was a waiting line type, considerable details of job arrival patterns, processing rules, procedures and times had to be gathered, condensed and analyzed.

(3) Because of its complexity of operation and mathematical representation of inputs and processing distribution functions, effort to achieve an analytic solution would be impractical. Consequently, a simulation approach was used to explore the system behavior and draw conclusions.

Since the simulation of the POD system is presented as an example, data, results and conclusions should be regarded as hypothetical.

The POD Waiting Line

Waiting lines result from one of two, or both types of conditions:[3]

(1) Computer jobs requiring processing must wait because there is a shortage of facilities. The shortage may be due to lack of computer input capability, memory, output, excess processing time or inefficient scheduling.

(2) The computer remains idle, waiting for jobs. This costly idle time may be caused not only by lack of jobs, but also by the nature of the time-spacing between job arrivals and the distribution of processing time.

The problem is visualized in Fig. 1. In either of these situations, a waiting line develops. In situation (1) the jobs are the waiting line are waiting for service. In (2) the units in the waiting line are computer facilities waiting for jobs. Complicating the problem is the cascading of stages through which the jobs must flow. Here the mathematical representation becomes most difficult and solution practically impossible in most cases.

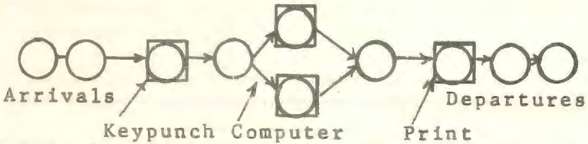

FIG. 1. EXAMPLE OF COMPUTER WAITING LINES

Solution To Waiting Line Problem[2]

The solution to waiting line problems involves the manipulation or control of job arrival rates, priorities, number and capability of computers and components. The purpose of the manipulation is to balance the cost of waiting against cost of idle computer units. Obviously, if one had the right number of fast computers, there would scarcely ever be a waiting line. But this could be very expensive, especially if the machines were idle most of the time. This would be justified however, if the cost of customer waiting were sufficiently large.

There are essentially two methods of approach to investigating and solving the waiting line problem:

(1) The mathematical approach, in which assumptions are made regarding the probability distribution functions of job inter-arrival time and job processing time. These are formulated into equations involving the total cost of the system as a function of these variables. Analytically, the minimum of this function is determined if possible to arrive at the appropriate decision. Suppose, for example, that too few printers are available to handle the output of the computer. Then the jobs pile up in memory and may eventually jam up the whole system. Here, the cost of waiting varies approximately inversely with the number of printers, whereas the cost due to idle printers varies directly with the number of printers. The sum of these costs will be large when there are too few and too many printers. The problem is to arrange the process, if possible, so as to minimize this total cost (see Figure 2).

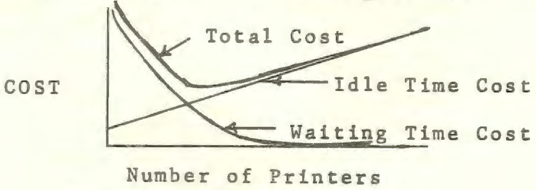

FIG. 2 WAITING LINE COSTS

(2) The simulated sampling approach in which statistics on arrivals, service and other system parameters are duplicated in a mechanical way. This method, known as the Monte-Carlo method, solves the problem 'on paper' by playing a sort of game with either the actual or assumed probability distributions of inter-arrival and service times involved. By varying the statistics and duplicating thousands of jobs for example, the effect of changing the number of printers, computer speed, memory, etc. can be studied without interfering with the real system or without any system at all. The procedure as illustrated in Figure 4 is quite simple and depending upon the number of trials, quite effective.

The bulk of the effort obviously is the determination of valid probability functions from empirical data if available, the processing logic, and the cost functions. But once this basic research is complete and model flow determined, simulation proceeds in a mechanical manner. The process is very time and paper consuming, however, for systems of even slight complexity if manually performed.

The Computer Simulator

Several years ago, IBM developed a simulation tool for use on the 7090/7094 system which essentially solved this problem. This tool is called the GPSS (General Purpose System Simulator)[1] and is essentially a programming language much like FORTRAN which enables the analyst to structure practically an infinite variety of processing systems. This, in turn, allows the analyst to spend his efforts on model design and analysis rather than on programming a special purpose model. Because of these economies and conveniences, the GPSS was used to construct the model of the **POD** system.

The POD Model[4]

The flow of jobs through the **POD** system in addition to the flow through other processing units affecting **POD** operations is illustrated in Figure 3. Jobs arrive for processing at the ENTER block. Of these, 11.9% go directly to the 1401 computer for processing, 64.5% must be keypunched, 22.4% can proceed directly to the P.O.D. system through the 7044, and 1.2% must be converted from paper tape to magnetic tape format. Similarly, 3.2% of the jobs keypunched proceed to the 1401 computer, 4% of them are complete, 88.3% proceed to the P.O.D. system and 4.5% must be converted from paper tape to magnetic tape format. Continuing in this way, the sequence of processing for each type of job entering the system is modeled.

The distributions of interarrival time of various job types used were actual, theoretical and combinations of these. Probability distributions of actual processing times were used at each processing stage.

Whereas complete details of the actual model are not particularly relevant to this presentation, a small segment of the model is shown in Figure 4. The following discussion serves to familiarize one with the general block diagramming technique required by GPSS which is incidentally a useful discipline in systems analysis. The blocks are characteristic to GPSS Model II and further details on their use and behavior can be found in IBM GPSS II User's Manual.

Discussion of Model[5] (See Fig. 4)

Block. No.	Function
130	Delay of from 1 to 11 clock times to find related tapes, etc.
131	Assign to parameter 2 routings so that 95% of all jobs print on POD and 5% do not.
132	POD queue
133	Passes jobs only when switch 15 is open (Normal Condition) Closes only when management information is in system or when system is down for maintenance.
150	Routes production jobs to 151, 1401 job to 134 and management information to 152.
151	Zeros parameter 4 which was built up in keypunch.
152	Sets switch 15 thus not allowing other jobs to enter POD until management information job enters 7094.
134	Enter 7044 each job coming into this block takes up one 7044 buffer unit.
135	Read into POD. This single facility simulates all read in devices (card or tape). A more refined mode could simulate the separate card reader and tape drives but this version does not.
136	Leave 7044. The buffer unit occupied in block 134 is now released.
137	Enter disk storage. Storage capacity is removed from total available. Amount required for each job is the value carried in parameter 3 assigned at block 9.
138	This split is a function of the simulator. It allows code check packages to punch cards at same time 7094 is working on other parts of code check package.
141	This block allows no job to pass until the other part of the job is ready to enter 7094, that is, until there is a match at block 149. Also the output from this block uses parameter 7 as an identifier for code check and routes all other types of jobs to terminate allowing only code check to continue on to the card punch. That is possible because production work carries a 1 in parameter 7 assigned in block 7. Code check carries a value between 75 and 140. All other type jobs carry higher numbers.
140	7094 queue data is collected at this block.
145	This gate allows no job to pass unless the 094 (block 153) will accept this job in the current clock time.
149	This match allows both halves of the job created by block 138 to proceed. Since block 141 is fed directly from the split block 138, this match can never delay a job in block 149.
143	Enter 7044, buffers as required by parameter 6 assigned in block 11 for production jobs, block 305 for code check, block 243 for paper tape jobs and block 259 for management information programs. Management information programs carry a 26 in this parameter. All other job types carry a random number between 4 and 13. Also management information jobs are routed to block 142 and all others to block 153. Parameter 4 is used as the identifier for this sort.
142	Reopens gate set at block 152.
157	7094 Facility. Time in facility is controlled by Parameter 5 assigned in block 258. Only management information is routed thru this block.
153	7094 Facility. All POD jobs except management information systems are routed thru this block. Time in the

Block No.	Function
	facility is determined by parameter 5 and a modifying function. Parameter 5 for production jobs is assigned a + block 10, for code check at block 304. The modifying function is used to convert from time per unit control number to time/computer work request.
154	7044 Buffers occupied in block 143 are released. On output side this block separates code check from all other jobs, terminating code check (the other half from the split at block 138 is already being allowed to go to printing and card punch.) and allowing other jobs to pass.
155	This block uses parameter 4 to separate management information jobs from all other jobs. MI Jobs go to block 359, all other go to 156.
156	This split is necessary to allow output operations of card punch, print and tape write to proceed simultaneously. One side of this split goes to card punch and the other will be resplit to go to tape write and print.

Sample of Experiments

Some of the more interesting and valuable experiments performed were to:

(1) determine the affect of more programming code check upon the system

(2) determine a schedule of submitting code check so as to reduce and if possible minimize total average turnaround time

(3) determine effect of unevenness of arrival rate of production jobs

(4) determine the effect on turnaround time of moving the processing of Management Information Systems from the POD system to the 1401 system

Data Summary

A summary of the statistics used in the simulation is shown in Table 1. In this table the notation 6:3 min. is used to indicate a mean value of 6 with a spread of 3. In this particular case any value between 9 and 3 min. is equally likely.

The following facility use times could not be adequately described by a mean and spread but required defining of a specific function. The operation of these functions in the simulator may be understood if we imagine a horizontal axis uniformly scaled from 0.000 to 1.000. This is the random number axis. The vertical axis is the function axis (keypunch time, etc.). The given points are plotted and joined by straight line segments. Thus a curve can be approximated to any desired degree by including as many points as needed to give required approximation. In use, a random number is generated to enter the curve and the corresponding value of the function determined by linear interpolation between the nearest two given values. In the keypunch data shown in Table 2 for example, if the random number generated was .500 this is 1/2 of the way from 4 to 6, therefore the value of the function used would be 5 minutes.

Weekend workload - .3 normal workload. Average workload was noted to be day dependent during the week as well as on weekends, but the variation was small enough that it was not felt to be worth the considerable extra effort to have the simulator generate different workloads for each day of the week. Weekend drops were significant enough to be built into the model.

Sample of Results

To accomplish 1 and 2, simulations were run with from one to two hours of code check a day and with four to six hours of code check per day. Two simulations were run for each level of code check input, one with all code check submitted at the same time and the other with code check input divided into 10 batches and evenly distributed throughout the day.

The results of this experiment suggested that other unevenness of input might well be affecting turnaround time which led directly to 3. Data was collected as to rate of input for each hour of the day and a function was constructed to simulate this. Two runs were made each simulating five weeks of operation. One run used the functional input and the other was input by a mean and a spread which kept input rate constant within fairly narrow limits. In order to achieve some randomness in using the functional input a slight random displacement about the mean on the time axis was introduced into the simulator.

To accomplish experiment 4, a run was made exactly like the one using actual loading function except that all management information systems were diverted from POD to the 1401 computers. Running times for these jobs were proportionately increased.

Results of the experiments are:

(1) Increasing code check increased turnaround time. In general, the increase in system turnaround was greater than the increase in code check itme if the code check was submitted as a single package.

(2) To minimize the effect of increased code check on system turnaround time code check work should be input as nearly uniform around the clock as possible.

(3) Unevenness of production input caused a considerable increase of turnaround time. The Tables 3-A and 3-B are a summary of results.

(4) Relieving POD of the management information type runs significantly improved POD turnaround time without seriously damaging 1401 turnaround time. Results are summarized in Table 3-B.

The graphs in Figure 5 were plotted from data collected as follows: Two hour groupings were used. All jobs thru POD which were ready for release within two (2) hours of their time-in, were put in group one; all jobs with thru times of 2 to 4 hours in group two, etc., thru the group 36 to 38 hours. All jobs requiring more than 38 hours were clumped into a final group. Times plotted are mid times of the group interval. Percentage is

$$\frac{\text{number of jobs in 2 hour group X 100}}{\text{total number of jobs}}$$

The shapes of these curves are significant. Notice that the two curves with the functional input show significant tails beyond 24 hours while the uniform input does not. These long turnaround jobs are jobs which have been caught in queues which develop as a result of bunching of input and were therefore abnormally delayed. Also note that the curve from the simulation with management information diverted from POD drops very sharply up to 13 hours and then starts an erratic tail. A prediction can be made that if a simulation were run with uniform input and with management information system diverted the resulting curve would follow this one up to this break at 12 hours and then continue on a well-behaved negative exponential curve might be expected to.

Conclusion

The use of Monte Carlo simulation is a feasible and economic approach to analyzing and predicting the behavior of complex systems such as described in this paper. Indeed, it is the only approach available to explore the operation of systems for which analytic description and solution may be impossible. The availability of a computerized general purpose simulator, such as the GPSS, significantly reduces the model construction analysis and simulation time thus allowing for more efficient use of analyst time. The generalized nature of the model makes it versatile so that many different situations and alternative courses of action can be evaluated quickly and economically.

References

1. I.B.M. Corporation, General Purpose Systems Simulator II User's Manual, I.B.M. Data Processing Division, White Plains, N. Y.

2. Sasieni, Yaspan, and Friedman, Operations Research Methods and Problems, Chapter 6, John Wiley & Sons, Ind., 1964.

3. Van Voorhis, W. R., "Waiting-Line Theory as a Management Tool," Operations Research Vol. 4, No. 2, April 1956.

4. Forry, Smith, Tidwell, Data Processing Study, Memorandum, 5 December 1964.

5. Smith, H. L., Details of Computer System Operations Model, Memo for the Record, January 1965.

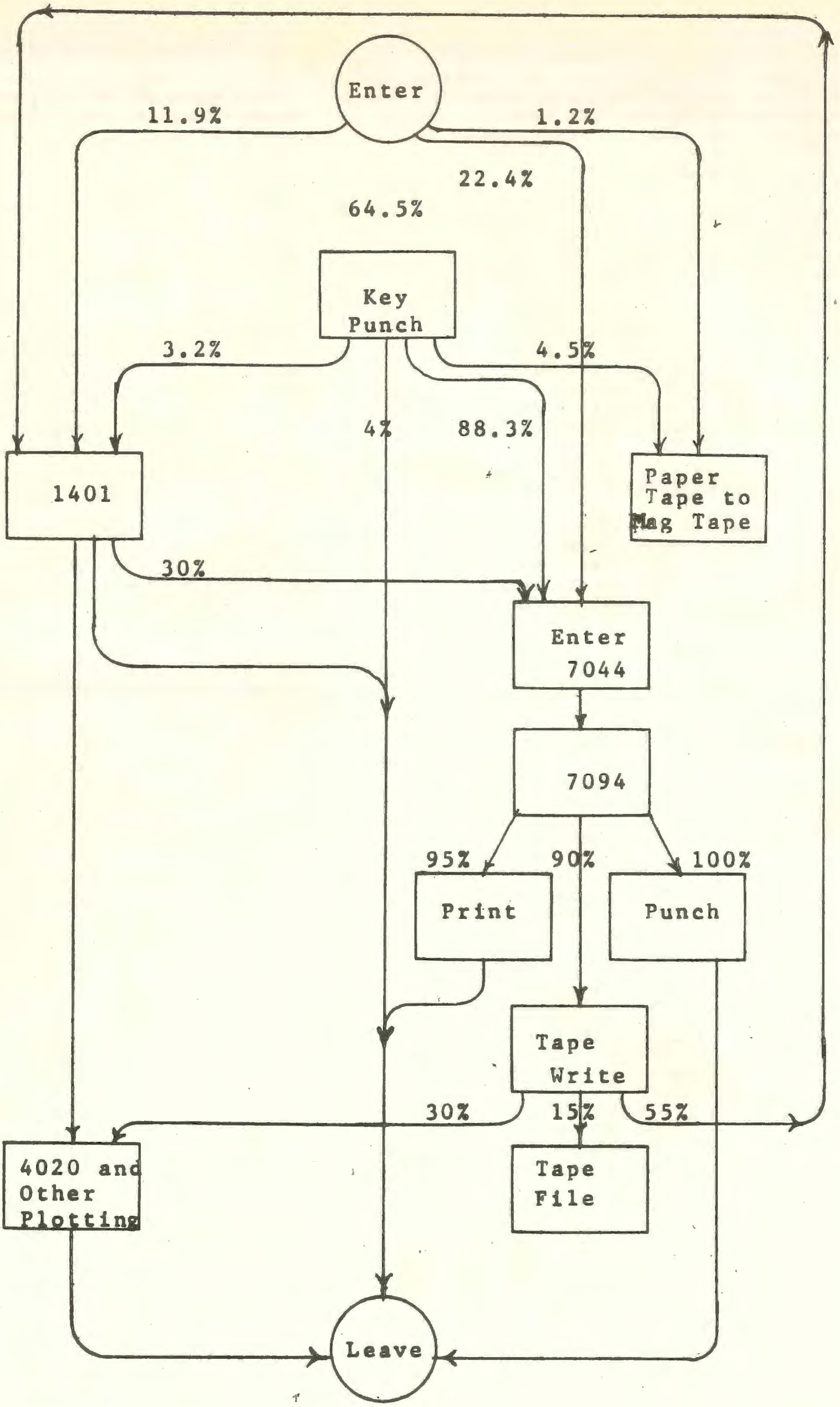

FIGURE 3. DIAGRAM OF P.O.D. MODEL

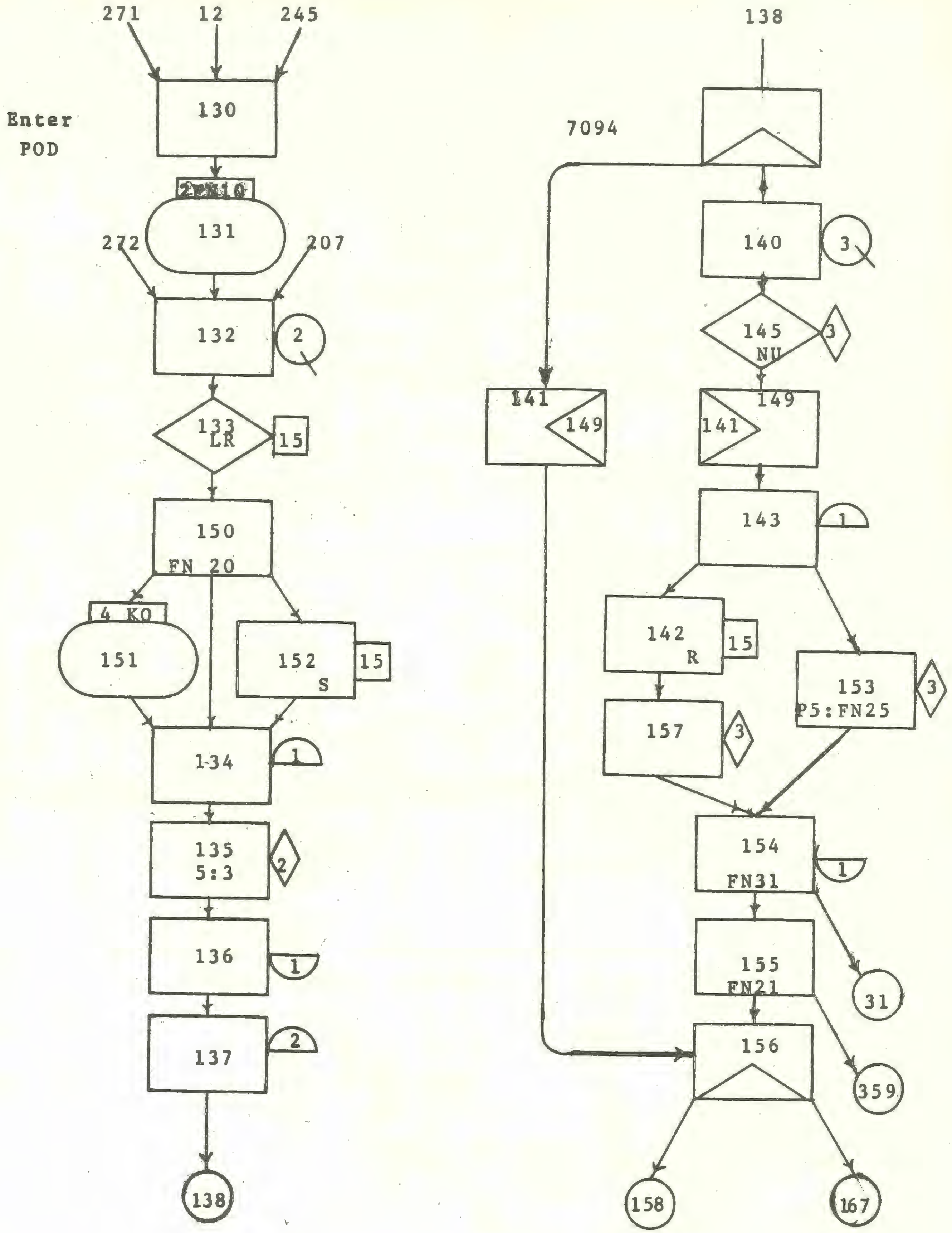

FIGURE 4. ILLUSTRATIVE SEGMENT OF THE P.O.D. SIMULATOR BLOCK DIAGRAM.

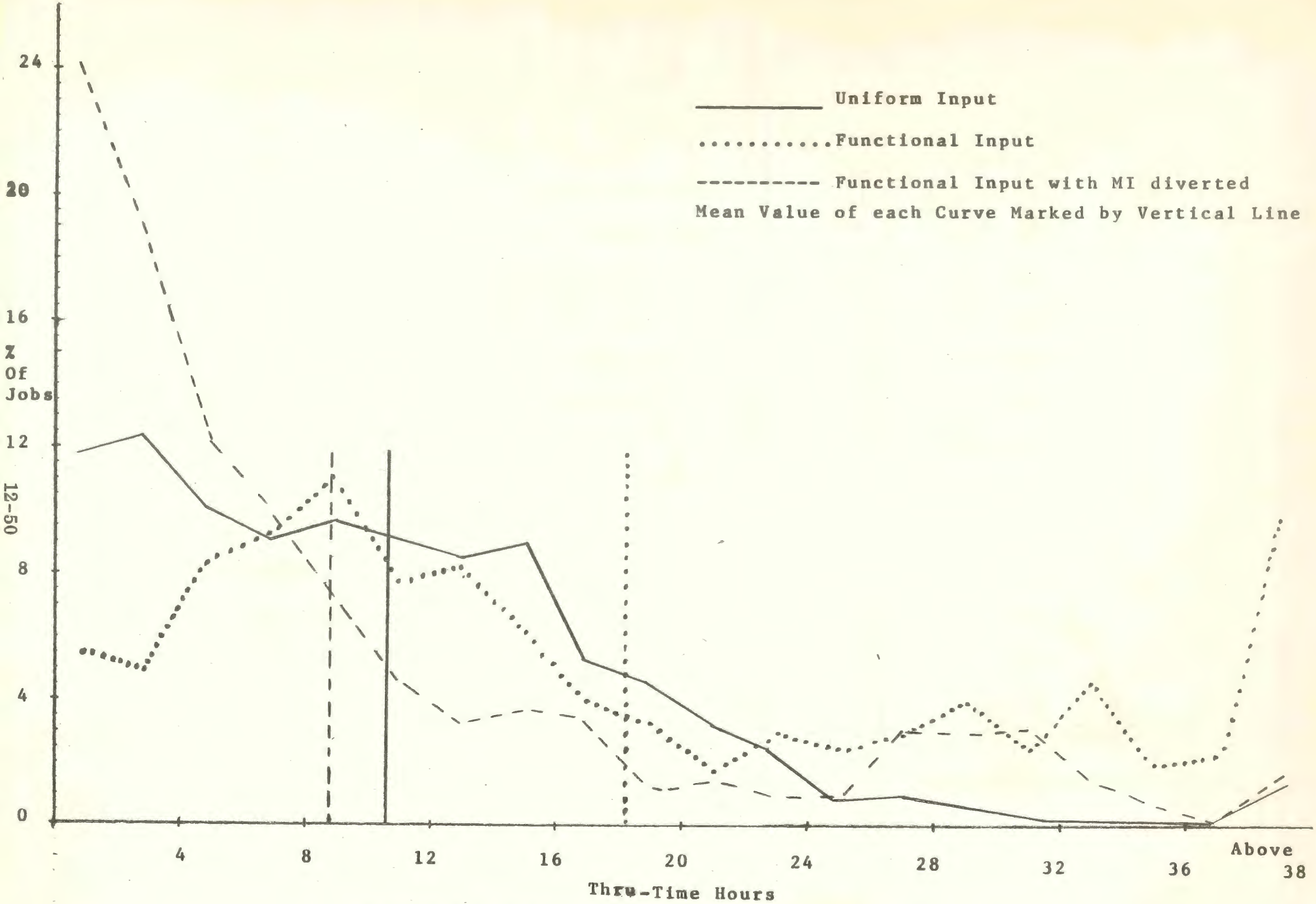

FIGURE 5. DISTRIBUTION OF POD TURNAROUND TABLES

FACILITY	ACTIVITY	TIME Mean: Spread
Keypunch	Blue Sheet	6:3 min.
EAM	" "	4.2 min.
1401	" "	4.2 min.
POD Read-in		5:3 min.
POD Print		
POD Print (Paper Change)		2:1 min.
Decolate	All	3:2 min.
POD Tape Write	All	3:2 min.
1401	Production	45:45 min.
1401	Management Info.	45:45 min.
Plotting	Production	12:6 min.
1011	Paper Tape to Mag Tape	30:18 min.
POD 7094	Management Info.	1-1/2 hr.
Keypunch	Management Info.	40 hrs/day
POD	Code Check	5:1 hr/day

TABLE 1. STATISTICS USED IN SIMULATION

Random Number	0	.2	.8	.98	1.0					
Keypunch Time	3 Min.	4 Min.	6 Min.	1 Hr.	18 Hrs.					
Random Number	0	.64	.83	.91	.95	.97	.98	.99	.995	1.0
7094 Time	0	2 Min.	5 Min.	12 Min.	24 Min.	33 Min.	42 Min.	1 Hr.	1.5 Hr.	4 Hr.

TABLE 2. PRODUCTION KEYPUNCH AND 7094

		Systems Turnaround Time	7094 Queue		
			Avg. Content	Max. Content	Avg. Time in Queue
A	Uniform Input	10.8 Hrs.	17 Jobs	66 Jobs	4.6 Hrs.
	Functional Input	18.2 Hrs.	30 Jobs	134 Jobs	8.0 Hrs.
		POD System Turnaround		1401 Turnaround	
		Time	Std. Deviation-σ	Time	Standard Deviation-σ
B	M.I. on POD	18.2 Hrs.	15.9 Hrs.	2.8 Hrs.	5.2 Hrs.
	M.I. on 1401	9.0 Hrs.	9.18 Hrs.	3.4 Hrs.	6.0 Hrs.

TABLE 3. RESULTS SUMMARY

FACILITY	ACTIVITY	TIME Mean: Spread
Keypunch	Blue Sheet	6:3 min.
EAM	" "	4.2 min.
1401	" "	4.2 min.
POD Read-in		5:3 min.
POD Print		
POD Print (Paper Change)		2:1 min.
Decolate	All	3:2 min.
POD Tape Write	All	3:2 min.
1401	Production	45:45 min.
1401	Management Info.	45:45 min.
Plotting	Production	12:6 min.
1011	Paper Tape to Mag Tape	30:18 min.
POD 7094	Management Info.	1-1/2 hr.
Keypunch	Management Info.	40 hrs/day
POD	Code Check	5:1 hr/day

TABLE 1. STATISTICS USED IN SIMULATION