WHOLE MILL SIMULATION OF SMALL LOG SAWMILLS WITH HEAD SAWYERS

K. H. Kempthorne

ABSTRACT

Two small log sawmills of different but typical configurations were modelled using GPSSV. Both models covered the whole mill from infeed deck to green chains and stackers. Such models have large memory requirements and required reallocation of common memory. User chains were also used to reduce processor scanning time.

Very simple downtime generators were used to randomly distribute downtime episodes of random length to several different sawing centers. These downtime segments make use of the 'unavailable' feature.

The head sawyer is also included in each model. The understanding of the decisions this person has to make was a major benefit of the model. Some experimentation has taken place on the possibility of using a micro-computer to relieve him of some of his decisions.

The creation and use of the sawmill models was valuable for mill operating people as they now better understand the systems they are operating.

One of these mills is known as Mill Five and is now under construction in Washington. The other, more complex, mill is located at Blue Ridge, Alberta. It was finished just over a year ago.

SAWMILLS

Small log sawmills convert logs into boards just as large log mills do but they operate on a high volume basis using logs once considered too small for sawing into lumber. Logs from the Olympic Peninsula range from as small as four inches in diameter to more than six feet. The small log mill usually has a maximum log size of twenty inches at the big end. The mill must operate at high speed to be profitable yet it must deal with a range of lengths of logs as well as a range of diameters.

Mill Five is being constructed to process logs from forests that have grown on lands first harvested late in the last century. This mill will handle only the smaller logs from these forests, the larger ones going to another mill. Small logs will be brought directly to the sawmill from the forest without any pre-sorting. They will be trucked to the mill in full tree lengths or less. Processing consists of debarking, buck sawing into short pieces called bolts and sawing the bolts into boards. Bolts will all be eight feet long starting from the large (butt) end of the log. If the last bolt will not come out to be eight feet long, it will be left as a part of the previous bolt and cut off at either three or four meters. So a tree length log 44 feet long will produce 4 eight foot bolts and one three meter bolt.

Rather than present these bolts to the primary breakdown saw as they are bucked, they are distributed into bins. At Mill Five there will be four bins. Three will contain eight foot bolts in three diameter classes and the fourth bin will contain the 3 and 4 meter bolts. The primary breakdown saw is called the headrig. At Mill Five it will consist of two blades so that logs can be sawn into three pieces. Figure 1 shows how the log is cut. The headrig makes only the vertical cuts. The smallest logs produce only the center piece of material for further processing. This piece of wood is called a cant. Larger logs yield one or two side boards in addition to the center cant. Chipper heads in the headrig remove the shaded areas in the form of wood chips. Chips are used to make paper and have a value to the mill of about 20% of the value of a similar mass of lumber. The saws in the headrig can be adjusted to suit the bolt to

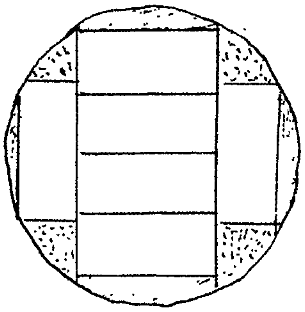

FIGURE 1

be cut. The smallest logs require saws set close together and vice versa. Setting takes time, time during which the saw is not cutting and is, therefore, lost production. Ideally the bolts would all be the same size so that no time would be lost to resetting the saws. In Mill Five, however, resetting time is reduced by pre-sorting bolts into bins. One bin is used as long as possible and then the saw is reset for the next bin. However, bin changes must be made even if there are still bolts in the bin in order to prevent downstream flooding or starving. Figure 2 is a simplified flow diagram of Mill Five. After the headrig there are three saws. The center cant is further ripped into boards by the vertical arbor edger. The sideboards, from bolts from the larger bolts bins, are edged in either the gang edger or the line bar resaw. Edgers remove edges (wane) by chipping away the speckled areas. Ideally they remove only some of the wane (curved edge) as grade rules allow some to remain on the board. The gang edger is also capable of ripping the sideboard into two individual, narrower boards.

When bolts are fed from the large bolt bin, the side edgers accumulate queues of sideboards. If the sideboard transfer conveyors fill, the headrig must either stop or change to the smallest bolts. The smallest bolts produce no sideboards so that while they are being processed, the side edger queues decrease. A happy balance must be achieved and that is where simulation comes in.

GPSS SIMULATION

GPSS was used to simulate the sawmill. This IBM program product is both a program and a language. For simulation of industrial facilities where it is desired to simulate the flow of objects (transactions) through some system, it appears to be superior to all other languages. It is very definitely superior to general languages such as Fortran for this type of task. This type of simulation is called "discrete event simulation." Each event has, or can be modelled as having a finite duration. Each event has a starting clock time, represented as an integer, and a stopping clock time. For each event to be simulated, a clock time for starting is computed. After each move of a transaction from its present block to the next block, the computer program searches for the next occurring event time. Suppose the last move was at clock time 50 and the transaction has come to rest. The search then begins for the event with the next closest start time. Suppose some other transaction is to leave the block it is in at time 90. The processor's clock is then advanced to 90 and the event is initiated. It can take more CPU time to simulate one minute in one model than it does to simulate a million years in another. CPU time is related to the number of transactions processed and the number and nature of the blocks they must be processed through. If nothing happens in the model between year one and year one thousand, the clock is advanced to 1,000 and the next event then takes place.

Creating models in GPSS is usually done in modules. Probably the best procedure is to build a model of the headrig and edgers first. Once this segment works, one can add the bins, and barker and cut off saws. The outfeed is easy to do if volume and grade simulation is left off. Most likely, the model should ignore many of the details until the bare minimum part works. There are five phases to go through with GPSS models where errors can occur. They are:

1. Assembly - converts GPSS to assembler.
2. Execution - moves transactions thru the model.
3. Logic - does it all make sense?
4. Structural Validation - is this our mill?
5. Data Validation - are the values we put in correct?

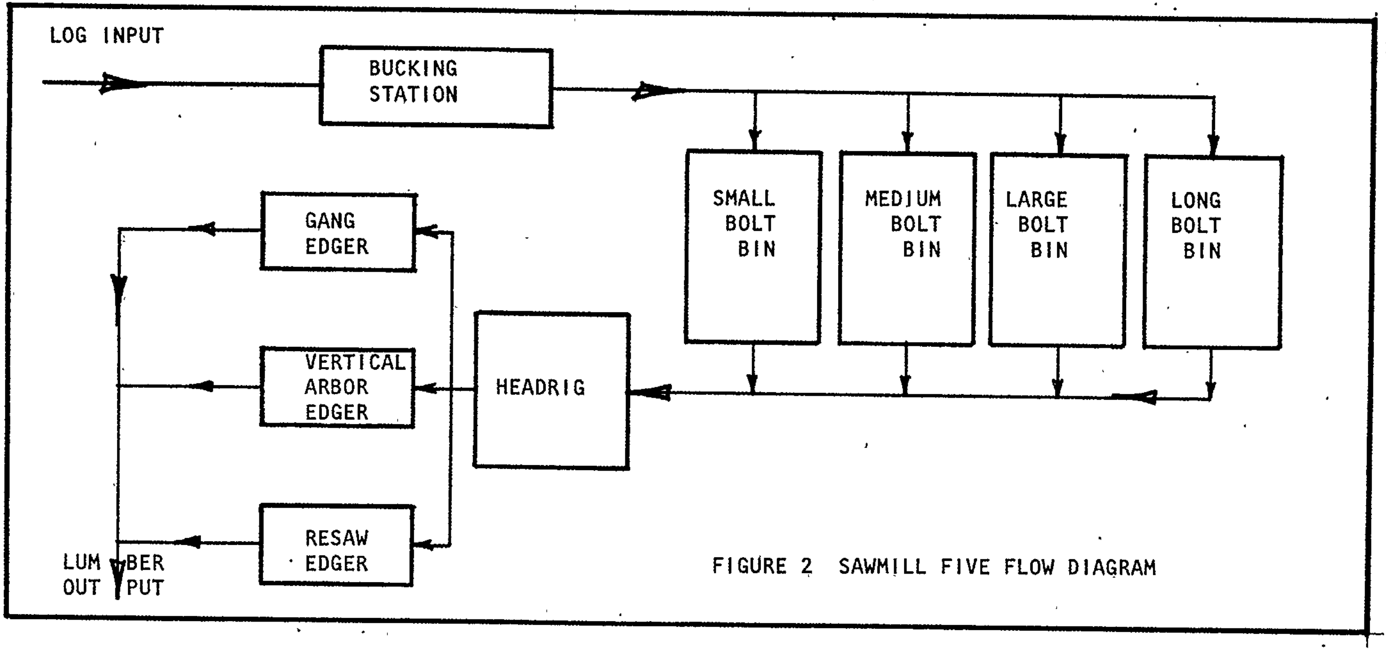

FIGURE 2 SAWMILL FIVE FLOW DIAGRAM

First one gets the model written in good GPSS. Then an assembled model can be generated and some useful aids can be found. The cross reference listings make location transfers and jumps easy to track down.

When execution errors occur, the GPSS system automatically prints detailed information on each transaction in the model. In these models, where a thousand transactions may be active at once, the output becomes voluminous. This detail can be very helpful in debugging. It is also helpful to give transactions some identifiers as they pass through the model. This can be done by use of parameter assignments, priority changes and the like.

As soon as the assembler is satisfied with the model, execution will begin (assuming you asked for it). Execution errors are more troublesome than assembly errors but there is a lot of help in the IBM user's guide (1), especially Schriber (5) is also plenty of help.

Logic errors, structural problems and data validation are common struggles in all programming. Most of these problems are jointly owned by the model builder and by the people designing or operating the actual mill. The modeler can hardly expect these people to locate problems and make corrections, those tasks belong to the analyst. The mill people do need to help identify problems and they need to help figure out what to do about them. Silver (6) discusses the need for involvement of design engineers and operating managers. Maisel (3) discusses several matters concerning the involvement of managers. Managers and operations people (mill people) must be involved with the development, validation and use of the simulation model if the project is to be of value to the organization.

THE PROBLEM

When management asks for a simulation model, the first task is to determine what use will be made of the model. For the construction of Mill Five the question was not hard to answer. Management wanted to be sure the mill could meet planned production levels. They wanted to be sure that the headrig would be utilized 100% of the time that it was available. The headrig must never have to wait for bolts because, for example ,the bins are empty. It should also never have to wait for downstream processing. In short, the headrig should be the bottleneck.

In a mill like the one already running at Blue Ridge the question to be answered was less clear. Specific questions were asked about the adequacy of the bucking station saws for example. The central question concerned increasing production. Some of the things that can be done to increase production are obvious but not economically viable. Since money is a constrained resource, the question has to be, "What can be done to increase the return on investment the most." It may appear to be obvious that the reduction of downtime at the headrig would increase uptime and hence, total production. This is true, however, only if the headrig is unconstrained by prior and subsequent system elements. Furthermore, it must remain so unconstrained at least up to the new level of uptime.

A good example of this situation is the excess downtime caused by bolts that are inproperly bucked. Bolts with excessive sweep (curved bolts) jam the headrig and account for, let's assume, 5% downtime at the headrig. Suppose we retrain the cutoff saw operators to buck these bolts shorter. Now the headrig will have less downtime. There will be a direct relationship between the reduced downtime and increased production if the headrig remains unconstrained. If, however, the increased workload at the cutoff saws causes them to produce fewer bolts so that the headrig is now up and ready but has no logs to process, then gains are less than direct. If the cutoff saws were already the bottleneck, the change will be for the worse. A slightly less obvious effect would be that bins would go empty more frequently resulting in lost headrig operational time due to more frequent bin changes and hence, resets.

EXPERIMENTATION

In general our pattern at the Blue Ridge Mill has been to devise experiments to perform on the sawmill to increase its output. The experiments are then run on the simulation model first. While this hardly guarantees which changes have the most significant effects it is an improvement over using averages.

Management must understand that the simulation model will not yield the one best solution. This confusion can arise pretty easily in the mind of a person who has dealings with linear programming, especially if he has heard that simulation can be used to answer questions of a similar nature. It must be understood that the simulation model is used because LP can't do the job. In fact, it won't hurt for people to realize that simulation is a technique of last resort. The desire to experiment first with a model rather than the actual production system is explained very well by Rao and Smith (4).

Gordon (2) the originator of GPSS, speaks of the experimental nature of simulation this way:

> "The simulation technique makes no specific attempt to isolate the relationships between any particular variables; instead, it observes the way in which all variables of the model change with time. Relationships between the variables must be derived from these observations." (2)

He refers to simlation as an "experimental problem-solving technique." When operating people know that it is a tool to give them better information upon which to exercise their own judgment, they feel more comfortable with it. The mistaken idea that computers can make decisions is an error too often encouraged by people who ought to know better.

The flow of material is not constant through the sawmill. The bucking saws generate pulses of bolts. These pulses are not regularly spaced due to the varying length of the stems coming to the bucking station, the variability of movement along the conveyor due to logs snagging or slipping and due to time variation in positioning the long log before cutting.

The number of bolts destined for each bin varies from log to log. There may be one bolt sent to the largest bolt bin from one stem, none from the next several stems, then 2 bolts from another stem and so on. If the bins are quite full this surge is easily smoothed. If not, it can be a serious problem that reduces output. The headrig processes logs from the smallest bolt bin at 220 feet/minute. Those from the long (3 & 4 meter) bolt bin are also at this speed because they are small diameter bolts. The larger bolts are processed at 180 fpm. There are no sideboards in the bolts from the two small diameter bolt bins while there are either one or two in the larger ones. Time is lost during headrig resetting which is the reason for several bins. Ideally one would have bolts with just the right number of sideboards to permit the headrig to run flat out without either flooding or starving the downstream machines. This is not realistic so one must saw large bolts to generate material for the other machine centers and when their queues fill up, switch to small bolts. So even if the headrig receives a smooth flow its output will be pulses. Again, in the ideal world, the side edgers receive a smooth flow. In reality, they don't. Ordinary queueing equations could deal with this. When we throw in downtime at the various machines, the flow becomes even more irregular and the problem becomes mathematically intractable. Simulation, the technique of the last resort, is then essential.

Experimentation with the actual mill can be done, of course. It often is, but in many cases and especially when the mill is not yet built, this procedure doesn't work. One reason is cost.

To experiment with mill means lost production which, of course, means lost revenue. In a business that has a very small margin, this is intolerable. Experiments that call for different machinery are obviously quite difficult in any plant. Less obvious are the people problems that experimentation can either lead to or suffer from. If an experiment calls for a different set of operator procedures for example, the operator may need to be retrained at a high cost. With a model one only need type in a few numbers or at worst change a few lines.

The model is not the last word, of course. If a proposed change in procedures is modeled and proves to work well, then one may feel justified in making an expenditure to test it in the real mill. In many ways, this experiment is more difficult due primarily to the lack of good control and secondarily to the inability to accumulate statistics. The model described here, for example tallied the number of bolts in each of four bins once each minute and at each headrig setting. This is all but impossible in reality.

THE INPUT LOG MIX

Just how much detail should be present in the log mix is dependent upon the kinds of questions to be asked of the model. For the models discussed here, the following procedures were used.

For Blue Ridge, a stem enters the model and is split into bolts for the large log barker deck. The number of bolts of large size in each stem is found and by evaluating the function LARGE. The function is a three point discrete function expressed in GPSS as ".72,1/.94, 2/1.0,3". If the random number generator returns .72 or less ($Rn \leq .72$) then one bolt is split off for the large log deck. If $.72 < Rn \leq .94$ then two bolts are split off to the deck and so on. The master tansaction, the stem, then moves into the next block whereupon the function SMALL yields the number of bolts to go to the second deck. After bolts pass through the barkers, they are sent to bins. Deck one bolts all go to Bin 1 while Deck 2 bolts are distributed by evalutating the function BINS. Once the bolts reach the bins, they are assigned a value. Based on the evaluation of the function for Bin 1 it is BORDA. For Bin 2 it is simply a constant 231. For bolts destined for the smaller V-6 headrig, the slasher deck precedes the single bin, bin 4. At the slasher deck, the function EIGHT is evaluated. An easy way to use the function to find the average number of studs in each piece sent to the slasher deck is to calculate the total number generated by a hundred entries to the function. If a hundred pieces evaluate the function EIGHT you find (51*1) + (92-51*2) + (99.5-92*3) + (1.0-99.5*4) = 157.5 bolts generated. Sawmill people express a machine center's I/O as N in, XN out or 1 in 1.575 out. When each of these bolts meets the headrig, they are split into FN$STUDS, an average 1.87 studs per bolt.

Boards to come out of each bolt from Bin 1 is obtained when bolts have their fifth halfword parameter examined and evaluated. This takes place first at the headrig, MKII. The block reads "SPLIT V$HDOUT, ANEXT". How many pieces are to go to ANEXT is found by evaluating the variable HDOUT. The definition of HDOUT is "PH5/100-1". PH5 was assigned the value of the function BORDA (FN$BORDA) so the GPSS processor takes this number, assume it was 554 ($.78 < Rn \leq .87$), and divides by 100 giving 5.54 and subtracting 1 giving 4.54 which is then truncated to 4 and 4 sideboards are then sent to the edgers. Additionally, every bolt has a center cant. This cant goes to the Shurman cant edger. One cant goes in and V$SHMO come out. This variable is defined as (PH5/10)@ 10 which uses the packed function again. We assumed, above, that PH5 is 554. So 554/10 is 55.4 which divided modulo 10 yields 5 boards out of the Shurman. One sideboard goes in and V$CNEOU boards come out. This variable is defined as "PH5@10" where PH5 is the packed function we got from FN BORDA above, 554 which divided modulo 10 yields 4 boards out of the cant.

In the Mill Five model, the stem breakdown is done with the packed function, 'sizes', listed in figure 3. Here, the kinds of stems that can exist are listed like a table and each random number drawn results in a table look up. A table can be built based on the standard log scanner reports in an operating mill. Good practice would call for an average of several such reports. One might prefer to use ten such reports to create ten separate functions. The model could then be run for several shifts, drawing from these ten functions through still another function.

DOWNTIME SEGMENT

Several problems have been experienced in the simulation of downtime in the sawmill. The first problem was encountered in the construction of the model. How can a machine be put in "down" status in GPSS? In the versions of GPSS prior to the current GPSSV, a transaction must be created which will either seize or preempt a facility, enter a storage, or set a logic switch. The output statistics are not affected in a convenient way by this approach. In GPSSV one has the 'FUNAVAIL' and 'SUNAVAIL' blocks. These blocks, entered by a separate transaction, make either a Facility or a Storage unavailable for use. These blocks have a nice effect on the statistics as they alter the utilization values. Facility statistics then look like those in figure 4.

With downtime simulated through the unavailable feature, resetting of the headrig ran into conflict. The setting time for this saw, something on the order of 1 or 2% of the total time, was also simulated by the use of the 'unavailable' feature. Problems arose when the saw operator attempted to reset the saw while it was down. An entry to a FUNAVAIL when the facility is already in unavailable status does nothing, i.e. acts like a no op block. However, the setting transaction taking an advance time of only a few seconds soon hit the FAVAIL block putting the machine back into operation and distorting the downtime statistics. A really complete simulated operator would reset the saw at every onset of downtime! Out of the need to hurry along, the easy way out was sought and appeared to be to use a PREEMPT block to simulate resetting. This had the disadvantage that utilization statistics now considered setting time as utilization. The result was that readers of the output statistics had to manually reduce the utilization values.

When time permitted, more thought was given to the problem. The possibility of using the HELP block subroutine feature was considered. This feature allows use of FORTRAN, PL1, or ASSEMBLER to perform special functions. However, GPSS is itself flexible enough to handle this problem. The insertion of the block 'GATE FV' before a reset FUNAVAIL and before the downtime FUNAVAIL prevented resetting if the machine was down and prevented going down during resetting, solving all the problem. The benefit is that setting time, like downtime, is considered to be unavailable time.

Problems occur with the language used in the mills. Downtime was understood by sawmill people to mean time the saw is not 'in the wood'. If there were no bolts available to the headrig, for example, it would be described as down. Additionally, if the downstream portion of the system was stopped or completely full of boards so that the headrig could not output, it was considered down. After some struggle with different ideas about downtime (some people believe words are our masters) it was decided that different terms

Fig 3

```
*
 SIZES FUNCTION   RN1,D7   AVERAGE DIAMETER = 5.8 INCHES.
.02,1000/.12,2000/.22,3000/.42,4000/.67,5000/.87,3210/1.0,3220
```

Fig 4

```
                                          -AVERAGE  UTILIZATION  DURING-
FACILITY    NUMBER        AVERAGE          TOTAL     AVAIL.    UNAVAIL.  CURRENT    PERCENT
            ENTRIES       TIME/TRAN        TIME       TIME       TIME    STATUS AVAILABILITY
  INFED      1157           46.511         .192                                    100.0
  BARKR      1086           99.854         .388       .421       .000       A        92.1
  COSAW      1086          120.000         .467       .488       .000       A        95.5
  HDRIG      4906           43.233         .760      1.000       .000       A        76.0
  VAREG      1963           33.579         .236       .241       .000      NA        97.8
  GNGER      4625           45.404         .752       .883       .000       A        85.1
  RESAW      5392           39.162         .756       .775       .000       A        97.6
  STAKR      6576            7.000         .164       .177       .000       A        92.9
```

should be employed. By teaching people the terms 'available/unavailable' and 'utilization' some of the confusion was eliminated. The saw is available to saw or it is not. If it is available and sawing it is utilized. Utilization during available time is considered, by sawmillers, to be the primary criterion of mill performance.

ACCUMULATING STATISTICS

When the system does not perform as expected (or performs better) people want to know what is happening. If the log bins run low and cause the headrig to have no logs to saw, the standard facility and storage output may not give you a clue. They will show bins to be less than full, of course. They will show the headrig to be less than fully utilized too. But they will not show relationships. Standard statistics are good for averages but less desireable for actual figures. Suppose Bin 3 (Storage 3) has a graph of the number of logs in it versus time like figure 5.

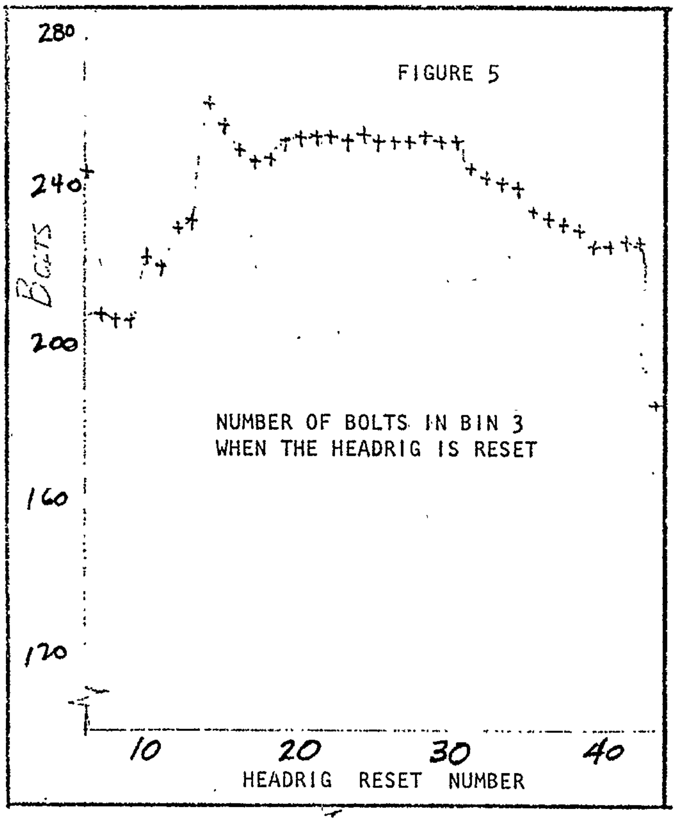
FIGURE 5

The standard statistics show an average of 50 logs in the bin. The maximum content is 100, but for how long? Is the bin ever empty? If a table is created, considerably more information is saved. With the table one gets a standard deviation. In addition, of course, one gets the frequency distribution, see figure 6. Still there is information that is needed and not present and that is a picture of the value of S1 over time. The easy way to get this information is with the Matrix Savevalue. From it, graphs can be constructed. As long as a matrix is being constructed, one might as well capture other data as needed. The matrix SETNO generates one row every time the headrig is reset. The listing figure 7 shows how the operator falls through each of the MSAVAEVALUE blocks and gathers dynamic information on the functioning of critical parts of the mill.

OUTPUT

GPSS creates output in a standard format. Terms used in this output should be made familiar to the operations people. They need to understand the differences between facilities and storages. They need to understand why the queues between saws for example, are represented as storages rather than as queues. They will have extra trouble with the concepts of 'utilization' and 'availability'. They will probably have trouble with the separation of the two sorts of downtime at, for example, the headrig where they will use observed or estimated values for unavailable time that include only endogenous downtime. System dependent downtime, such as the headrig being unutilized due to having no input bolts available to it, is to be found by simulation. All these familiar concepts have to be patiently explained and reexplained. The most competent operating people are, however, very adept at getting staff technical people to explain what they mean without getting frustrated. The worst case is a primadonna staff expert pitted against an authoritarian and insecure manager. Both parties have responsibility for the success of the project.

Figure 5 is a particularly pleasing way to display the graphic information. GPSS can't do this graph but a subroutine could be attached or used separately. It can be done by hand, of course, just as this was.

Fig 6

```
TABLE  BINT1
ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION
          50                    33.519                        21.500

          UPPER       OBSERVED       PER CENT       CUMULATIVE        CUMULATIVE
          LIMIT      FREQUENCY       OF TOTAL       PERCENTAGE         REMAINDER
              0              0            .00               .0             100.0
             10              7          13.99             13.9              86.0
             20             10          19.99             33.9              66.0
             30              9          17.99             51.9              48.0
             40              7          13.99             65.9              34.0
             50              3           5.99             71.9              28.0
             60              6          11.99             83.9              16.0
             70              8          15.99            100.0                .0
REMAINING FREQUENCIES ARE ALL ZERO
```

Fig 7

```
        *
        *      MODEL SEGMENT 10 - AN OPERATOR WHO SELECTS THE PROPER BIN
        *
235            GENERATE    1,,600,1,99,5PB,4PF
236            TRANSFER    ,NOHIS
237      ANYHI TEST E      BV$FULL1,1,GOXON     NONE FULL?  THEN GO ON
238            TEST E      BV$FULL2,1,D003 NEITHER 1 NOR 2 IS FULL - 3 IS
239            TEST L      S21,60,D001      IF 1 I S FULL DO IT - ELSE DO 2
240            TRANSFER    ,D002
241      GOXON TEST LE     S*PB5,10,DUPB5   STAY ON PRESENT BIN
242      NOHIS TEST L      S21,10,D001
243            TEST L      S22,10,D002
244            TEST L      S23,20,D003
245            SELECTMAX   5PB,21,23,,S
246            TRANSFER    ,SCAN
247      D001  ASSIGN      5,21,PB
248            TRANSFER    ,SCAN
249      D002  ASSIGN      5,22,PB
250            TRANSFER    ,SCAN
251      D003  ASSIGN      5,23,PB
252            TRANSFER    ,SCAN
253      SCAN  TEST NE     PB4,PB5,DUPB5
254            GATE FV     MARK2     NO RESETS WHILE DOWN
255            FUNAVAIL    MARK2
256            INDEX       1PF,1
257            MSAVEVALUE SETNO,PF1,1,C1,MX  CLOCK TIME
258            MSAVEVALUE SETNO,PF1,2,PB5,MX BIN TO FEED
259            MSAVEVALUE SETNO,PF1,3,S24,MX QUANTITY IN BIN 4
260            MSAVEVALUE SETNO,PF1,4,S10,MX QTY ON SIDE CVYRS
261            MSAVEVALUE SETNO,PF1,5,S21,MX QTY IN BIN 1
262            MSAVEVALUE SETNO,PF1,6,S22,MX QTY IN BIN 2
263            MSAVEVALUE SETNO,PF1,7,S23,MX QTY IN BIN 3
264            ADVANCE     20
265            FAVAIL      MARK2
266      DUPB5 UNLINK      V$UPBBN,MAIN1,10
267            ADVANCE     10
268            ASSIGN      4,PB5,PB
269            GATE NU     MARK2
270            GATE FV     MARK2
271            GATE SNF    10,D003
272            TRANSFER    ,ANYHI
```

BIBLIOGRAPHY

IBM, General Purpose Simulation System V - User's Manual, IBM Corp., Pub. Number SH20-0851-1, 1971.

Gordon, G., System Simulation, Prentice-Hall, 1969.

Maisel, H., "Making Management Simulations Useful", Simulation Today, No. 41:61.

Rao, A. & Smith, H. Jr., "Simulation and its Role in Manufacturing Experimentation", Journal of Quality Technology, January 1974, pg. 37.

Schriber, T. J., Simulation Using GPSS, Wiley, 1974.

Silver, A. M., "Predicting the Performance of Complex Systems", in Gudnason, C. H. & Corlett, E. N., eds, Development of Production Systems, Halstead Press (Wiley) 1974.