# DESIGN OF FEED MILL FACILITIES BY COMPUTER SIMULATION

by 557

OMAR J. HERNANDEZ

Ingeniero Industrial, Universidad de Carabobo
Venezuela, 1966

---

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Grain Science and Industry

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971

Approved by:

Major Professor

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERENTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

LD
2668
T4
1971
H4
c.2



# TABLE OF CONTENTS

LIST OF TABLES . . . . . iii

LIST OF FIGURES . . . . . iv

INTRODUCTION . . . . . 1

REVIEW OF LITERATURE . . . . . 4

SOME BASIC PRINCIPLES OF SIMULATION . . . . . 6

SIMULATING A FEED MILL . . . . . 8

Description of Events and Routines . . . . . 10

Uncontrollable Variables . . . . . 25

Controllable Variables . . . . . 28

Measures of Performance and Statistics Collection . . . . . 30

SIMULATION PROCEDURE AND EXPERIMENTAL RESULTS . . . . . 33

Analysis of Results . . . . . 37

SUMMARY . . . . . 51

ACKNOWLEDGMENTS . . . . . 53

BIBLIOGRAPHY . . . . . 54

APPENDIX A: Q-Systems of Inventory . . . . . 57

The Approximate Method . . . . . 57

The Exact Method . . . . . 61

Perishable Products . . . . . 62

APPENDIX B: Computer Programs and Data Arrangement . . . . . 66

APPENDIX C: Demand Distribution . . . . . 94

APPENDIX D: Frequency Histograms . . . . . 98

THIS BOOK CONTAINS NUMEROUS PAGES WITH MULTIPLE PENCIL AND/OR PEN MARKS THROUGHOUT THE TEXT.

THIS IS THE BEST IMAGE AVAILABLE.

## LIST OF TABLES


| Table | | Page |
|---|---|---|
| 1. | Fix Point Attributes (JTRIB'S) . . . . | 26 |
| 2. | Floating Point Attributes (ATRIB'S) . . | 27 |
| 3. | COLCT Statistics . . . . . | 31 |
| 4. | HISTO and TMST Statistics . . . . . | 32 |
| 5. | Inventory Parameters . . . . . | 36 |
| 6. | Simulation Results . . . . . | 38 |
| 7. | Simulation Statistics. Model 1 . . . . | 39 |
| 8. | Simulation Statistics. Model 2 . . . . | 40 |
| 9. | Simulation Statistics. Model 3 . . . . | 41 |
| 10. | Pelleting Report . . . . . | 42 |

# INTRODUCTION

Simulation is a natural development in the discipline of Operations Research. The group of techniques that today is known as Operations Research evolved out of efforts to provide formal, efficient, decision-making techniques for the design of air defense operations in Britain during World War II. Since that time the methods of Operations Research have been refined and its areas of application expanded until today when it is being successfully applied to a wide range of problems.

As techniques for the solution of management problems were developed and utilized (linear programming, queueing theory, inventory control--among the more popular ones) the operation researcher turned his attention to more complicated management decision processes discovering that neat mathematical procedures could no longer be found to perform the necessary analysis. Difficulties also occur in applying mathematical techniques to everyday management problems like scheduling jobs through a shop. Although scheduling appears to be straightforward, operations researchers have yet to provide a complete solution. Decision makers who face these problems and somehow make decisions are the first to admit that the basis of their decision is qualitative and could be improved if appropriate quantitative aids were available. This need for quantitative techniques for complex situations, combined with the ready availability of large and fast computers, led to the development of simulation. Its widespread use today is apparent from the number of recent books published on this topic.

In the determination of rational policies through the use of simulation, it must be recognized that simulation is not an end in itself. The simulation

model is, rather, a vehicle from which data for further analysis are collected. Thus, the simulation model becomes an intermediate phase of the experimental process to provide information which may be analyzed in order to draw conclusions about the system under study. In many cases, the actual simulation model is used in an interactive manner with classical optimization techniques which dictate various policies to simulate.

This thesis has as a main objective the construction of a model that will simulate the internal operation of a plant for the production of feed-stuffs, which could be used either in a design stage to try to achieve an adequate combination of production facilities or in an operational installation to gain knowledge on the effects of possible policy changes that by their specific nature may be too unsafe, costly or uncertain to be attempted in the real world system.

An important feature that simulation models present is their capability of integrating other techniques that apply to specific areas within a large system. In this work, special emphasis has been made in the utilization of dynamic inventory control techniques to dictate the policies regarding stock handling and levels of operation of the feed mill within the boundaries defined for the system under consideration..

When analyzing an integrated production-inventory system for a feed mill, current inventory techniques must be understood within the context of the overall system. That is, in trying to optimize the general operations it is not sufficient to determine the optimal inventory variables, namely how much to order and how often to order. It is even more important that most of the efforts be directed toward accomplishing the best combination plant-warehouse-inventory system.

Analytical models of inventory exist in which parameters considering physical warehouse dimensions and capacities can be inserted and their values

obtained automatically as a consequence of the inventory cost optimization process (3). But this, by no means, guarantees that the overall total operational cost will be optimal. Then, it is logical to initate a search that would possibly deviate from the total optimal inventory cost but ultimately yield a better overall total cost.

The intention of this work was to develop a basic simulation model and initiate a search for a better combination of the factors influencing the performance of a subsystem of the general feed production operation. It was not expected that under the constraints of computer time available for this research the optimal solution to the problem would be obtained. An illustration of the basic steps for building a computer simulation model may contribute, however, to more rational decision-making procedures in the feed industry.

# REVIEW OF LITERATURE

No extensive application of computer simulation techniques to problems of feed production operation is available in the literature, except for work by Stafford and Snyder (15) on a raw material procurement model that uses simulation to evaluate different safety stock policies.

Starr and Miller (16), treat the simulation of a Q-model of inventory control for a single product (fixed lot size, variable frequency of ordering), assuming that no sequential demand dependency exists. In other words, each month's demand is independent of the demand that occurred in the preceding month. They use a demand generator to produce random series of monthly demand which can be used as input to various decision rule boxes and satisfactorily compare the results obtained using their simulator with the results of the analytical model as far as the measures of performance were concerned. They suggest that assuming randomness of the demand process is the best policy to prevent bias in the results when less than one year's data is available. The use of queueing model simulators is recommended for low demand-high stockout cost items.

Naylor et al. (10), outline a production-inventory simulation model pointing out some of the elements in such a system. This model includes basic elements like generation of stochastic variates, inventory recordkeeping, scheduling according to predefined rules, and statistics collection.

Schmidt and Taylor (14), construct a next-event general Fortran simulator which can be employed in the analysis of either reorder point or periodic review systems.

Pritsher and Kiviat (13), illustrate the use of their simulation set of Fortran subroutines GASP II by showing two inventory applications. One of them deals with the Q-model of inventory control with Poisson-assumed demand, lost sales allowed, and constant lead time. The second application tests the effects of distribution types on inventory policy decision for another Q-model with backorders.

Wagner (18), indicates an algorithm to evaluate the operating characteristics of a proposed inventory replenishment rule by simulation.

Bowman and Fetter (2), describe a single-product production-inventory simulator written in Fortran that allows experimentation with a variety of decision rules for setting production rates according to an exponential smoothing forecast of the demand.

Nanda (9), developed a production scheduling simulator intended to be used as a decision-making exercise in job-shop problems.

Batra (1), investigated the behavior of the demand during lead time distribution when the variance of the demand is altered by using a probabilistic inventory control simulator.

As can be seen, most of the simulation work in the general literature that could be applied to feed mill problems are concentrated in the inventory control area, with some work done in the area of scheduling, and very little done in integrated production-inventory systems.

## SOME BASIC PRINCIPLES OF SIMULATION

Simulation has been defined as "a numerical technique for conducting experiments on a digital computer, which involves certain types of mathematical and logical models that describe the behavior of a business or economic system over extended periods of time", (10).

Computer simulation models often include descriptions of interactions which exist in the real world system but are difficult to express adequately in purely mathematical models.

To simulate a system one must be able to describe it in numerical terms. The key concept for doing this is that of the system state description.

If a system can be characterized by a set of variables, with each combination of variables values representing a unique state or condition of the system, then manipulation of the variables simulates movement of the system from state to state. This is precisely what simulation is: representation of the behavior of a system by moving it from state to state in accordance with well-defined operating rules.

The concept of an event which takes place at a point in time and either starts or ends an activity, thereby changing a system state, is a crucial one. In fact, the behavior of a system is reproduced by examining the system at event times.

The key to discrete-event simulations is to organize system events so that the order in which they are executed within the computer corresponds to the order in which they would occur in the real world system. This preserves the time relationships between simulated and real events.

Another basic concept is that of events scheduling. It is necessary to keep track of the times at which the different events are due to occur. This

is done by using information storage and retrieval tables (13), a very useful concept in computer-related techniques.

Basically, only one initial event would be required to keep the simulation moving in time based on the fact that this event can trigger the occurrence of the next event of its same class and at the same time initiate a chain reaction for the occurrence of other events, which makes up the entire system's dynamics. To accomplish this, every event that is entered or retrieved from the master information tables is defined in terms of relevant attributes like type of event, occurrence time, and any other information of interest that would permit setting up the logic which the system will follow in operation.

In formulating a simulation model it is desirable to associate some measure of performance with both the decision variables and the uncontrollable variables. Decision variables are those that are under control and can be freely manipulated to a certain extent (i.e. plant capacity, inventory parameters). The uncontrollable variables are beyond management control and are usually represented by probability distributions within the simulation (i.e. demand). These two types of variables must be precisely defined in every simulation study for they are vital parts in the process.

## SIMULATING A FEED MILL

One of the first decisions to be made in any simulation study is the language to be used for modeling the system. In this case GASP II-A (13), a Fortran based simulation language, was chosen because of the general availability of the Fortran compiler in almost every installation. GASP II-A is a set of twenty-six Fortran IV subprograms organized to assist in performing simulation studies. Common elements of simulation studies which are independent of particular problems have been specified and subprograms organized to perform these specialized tasks. Among the more important functions, GASP performs event schedule scanning and control, file maintenance, data collection, statistical computing and reporting, standard initialization, and error reporting. The typical arrangement of the GASP programs is presented in Fig. 1, where the interaction programmer-language can be observed.

Next, it is necessary to describe the system in terms of the different events that take place within its boundaries, and define precisely the nature of the decision variables, the uncontrollable variables, and the measures of performance that will be included in the model.

The system under study is a feed mill that will serve a market for bagged feed. The feeds are produced in either mash, crumble, or pellet form and are sold in several sizes of bags. For the purpose of this study the system will include the processes that take place from the mixing operation up to the dispatch of the finished products and demand flow. In other words, it is assumed a continous availability of raw ingredients and that a least-cost formulation has been performed previous to the mixing process. It is typical of every simulation model that its scope is determined by the particular problems the model is designed to solve. The primary objective here will be

**THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.**

**THIS IS AS RECEIVED FROM CUSTOMER.**

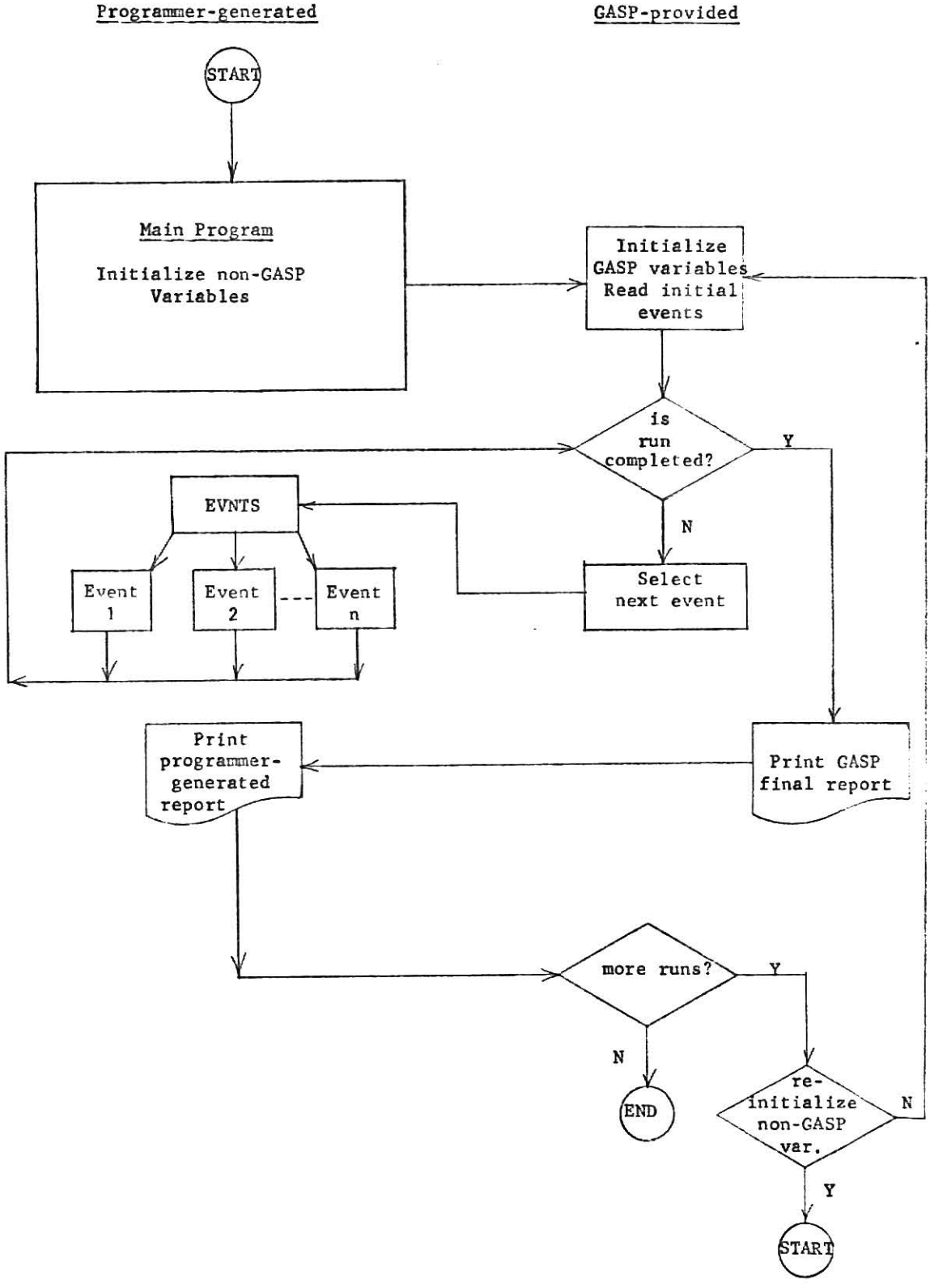



Fig. 1. GASP Programs Arrangement

the design of the basic components of a plant that will work linked to a warehouse according to a convenient inventory control system. Furthermore, the storage area required to hold a day's dispatch until it is picked up by the customers is considered beyond the boundaries of the system under consideration (in case information is available about dispatching time, amount and frequency it is recommended that this storage area be included in the system). Customers are willing to wait up to a known maximum number of hours (customer maximum lead time) before their orders are filled. Lost sales will be allowed, although a management policy is imposed upon the inventory control system in order to reduce stockouts to certain maximum levels.

Description of Events and Routines

TODAY (Fig. 2): this event acts like a master clock that updates the simulation time every day. The generation of demand also takes place in this routine. Orders are grouped by formula number and filed in a "demand backlog", from where they are removed later by the inventory control simulator called MASTER. More details pertaining to this event can be observed in Fig. 2.

MASTER (Fig. 3): this is not an event in itself but rather an inventory control and scheduling simulator. It removes orders from the "demand backlog", where they had been ranked on order size, low values first. Then, the order size is compared with the inventory position, that is the total of the amount actually in stock and the amount on-order for that particular feed formula, and a decision is made according to the particular case. When necessary, production orders are issued and their priorities assigned, filing them in a "production backlog" and ranking them on priority classes, high values first. Low priorities are given to feeds which reach reordering point after the day's delivery. High priorities are assigned when a customer has to wait for feed

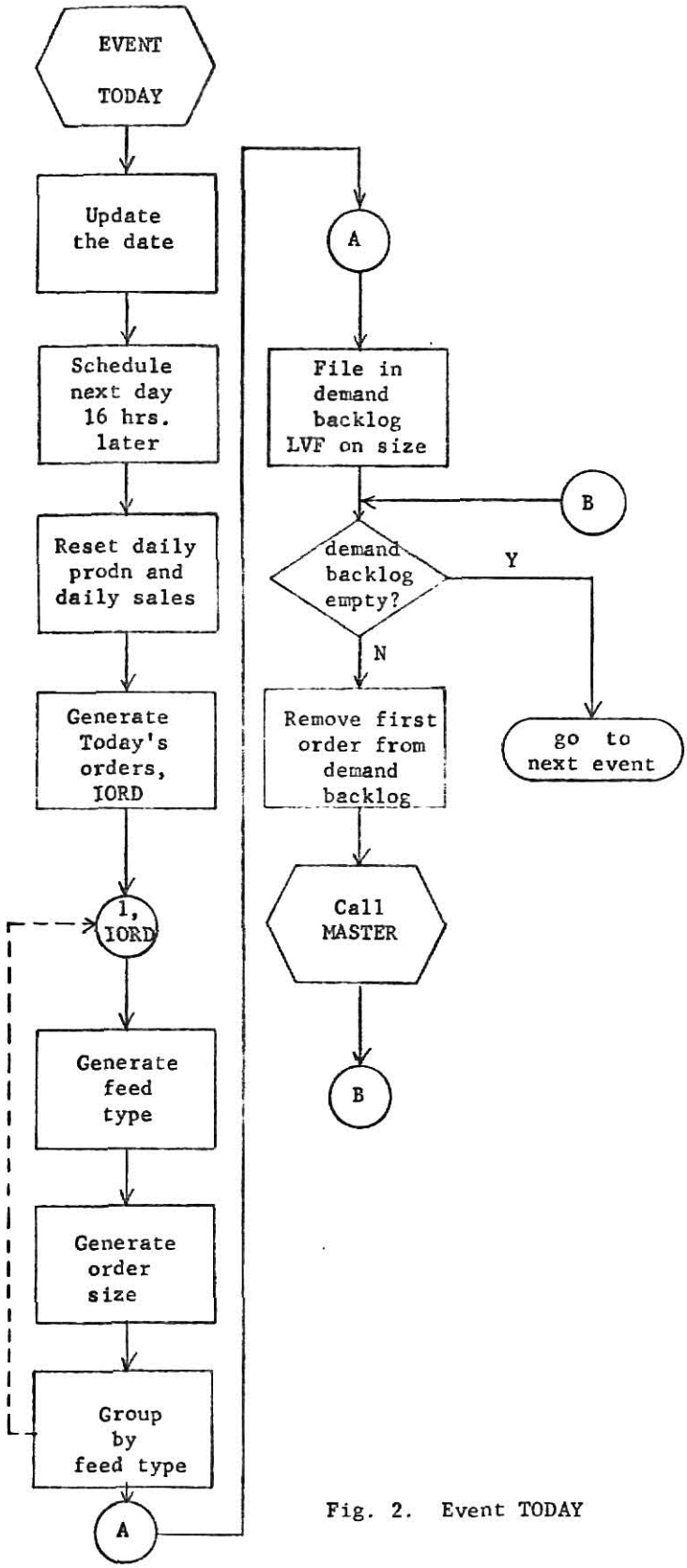



Fig. 2. Event TODAY

that will be produced. Production scheduling is performed on a first-come-first-serve basis, within priority classes. Fixed lot sizes are manufactured, according to a Q-model of inventory control.

In order for the reordering inventory system to be enforced, the maximum number of hours that customers are willing to wait (high priority order maximum lead time) will automatically set the maximum lead time for the replenishing orders (low priority) to be completed. Cancellation of replenishing orders will be made, when necessary.

Once the day's first order has been filed in the "production backlog", production activities can begin. If the mixer is idle, completion time for the first run is scheduled (event ENDMIX). For successive runs the time that the order has been waiting will be compared with its priority class maximum lead time. If greater, a plant stockout will be produced or a replenishing order cancellation will be effective. Otherwise, the logic process will continue. After the first order is scheduled in the different pieces of equipment (or in case the mixer is still busy for successive orders) the process of removing orders from the "demand backlog" continues and the inventory simulator functions are performed. Orders are either filled, delayed, or declared lost.

Before any pelleted order may be mixed, the routine checks for enough pellet mills available to deliver the order within the maximum lead time corresponding to the order's priority class. The check is made by considering order quantity, waiting time, and priority. This is the basic rule to determine the number of pellet mills to assign to a particular order. Next, the order is scheduled on the pellet mills in a fashion that reduces die changes to a minimum.

The route of the mix through the different pieces of conveying and processing equipment is simulated by including the corresponding delays, when

necessary; taking care that if any two delays superimpose the longest will be the effective one.

If the turn for a pelleted order has come and there are not enough mills available to process it, a plant stockout will be produced if the order is to be partially delivered, or a replenishing order cancellation will take place, if that is the case. After this, the simulator will remove the next order from the "production backlog" and the decision process will be repeated.

If not stockout is produced, the completion time of the order is scheduled (event ENDJOB) in the different mills and baggers. Even distribution of loads among mills is performed by simulation a physical distributor, and "readyness" time (event READY) is scheduled if part of the order is to be delivered.

If the order is for mash feed, the mixer must be idle to start any run. If so, mixing completion time, order completion time, and "readyness" time are scheduled. There will be enough bagging capacity to fill needs of feed being processed in either the mixer or the pellet mills.

When there is not enough feed in stock but on-order (from the previous day), customers will join the delivery waiting line until the feed is ready within the predefined priority class maximum waiting time.

Statistics are collected in all routines and events on relevant information (see measures of performance later). Cost accounting functions are performed as the different events take place.

The inventory control and scheduling functions contained in this routine are the kind of operations that would occur when on-line data processing is utilized. On-line means that the transactions are handled on real company time and not after hours. This is also typical of the Q-model of inventory which was chosen for this system of inventory control since it requires that perpetual inventory be kept.

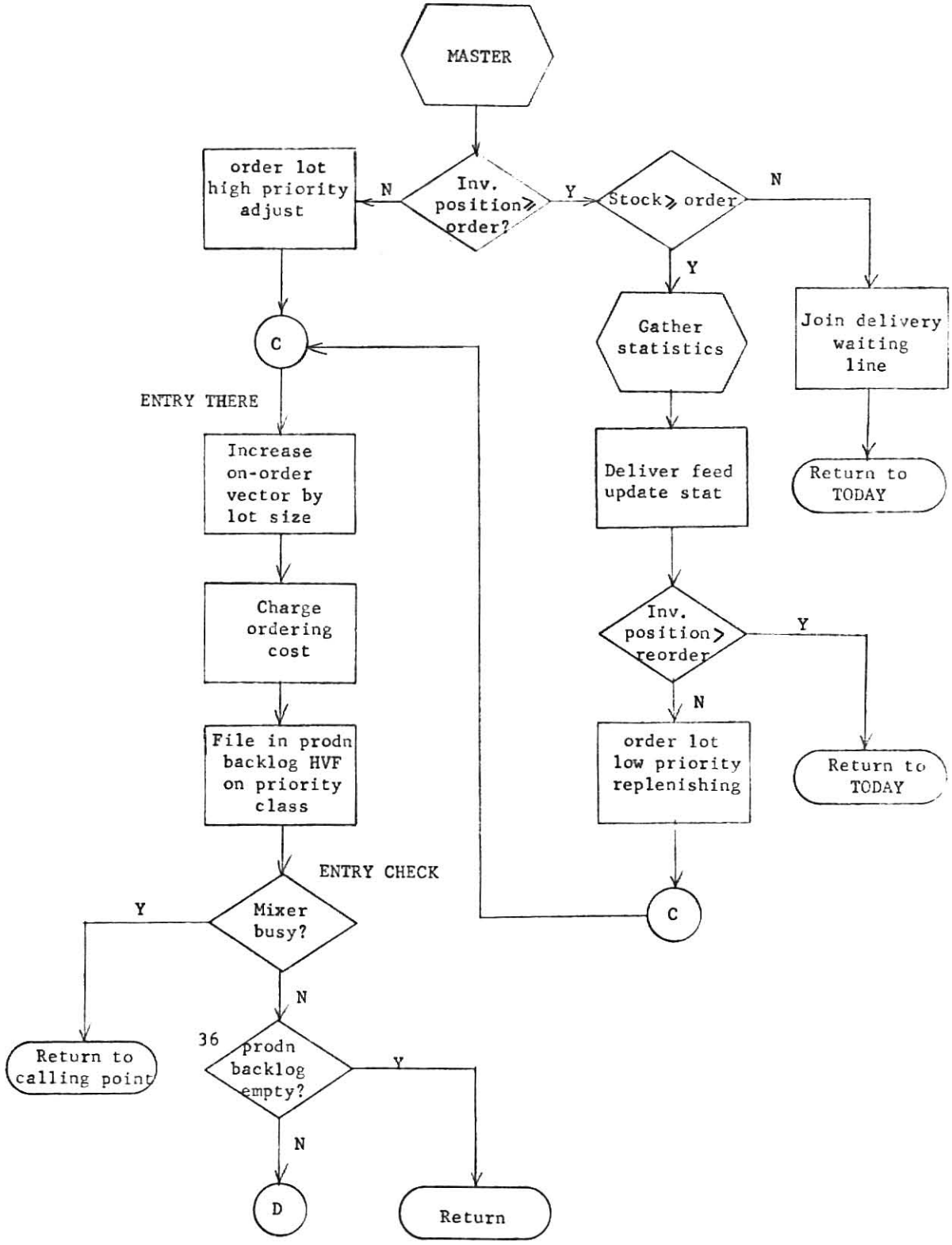



Fig. 3a. Routine MASTER

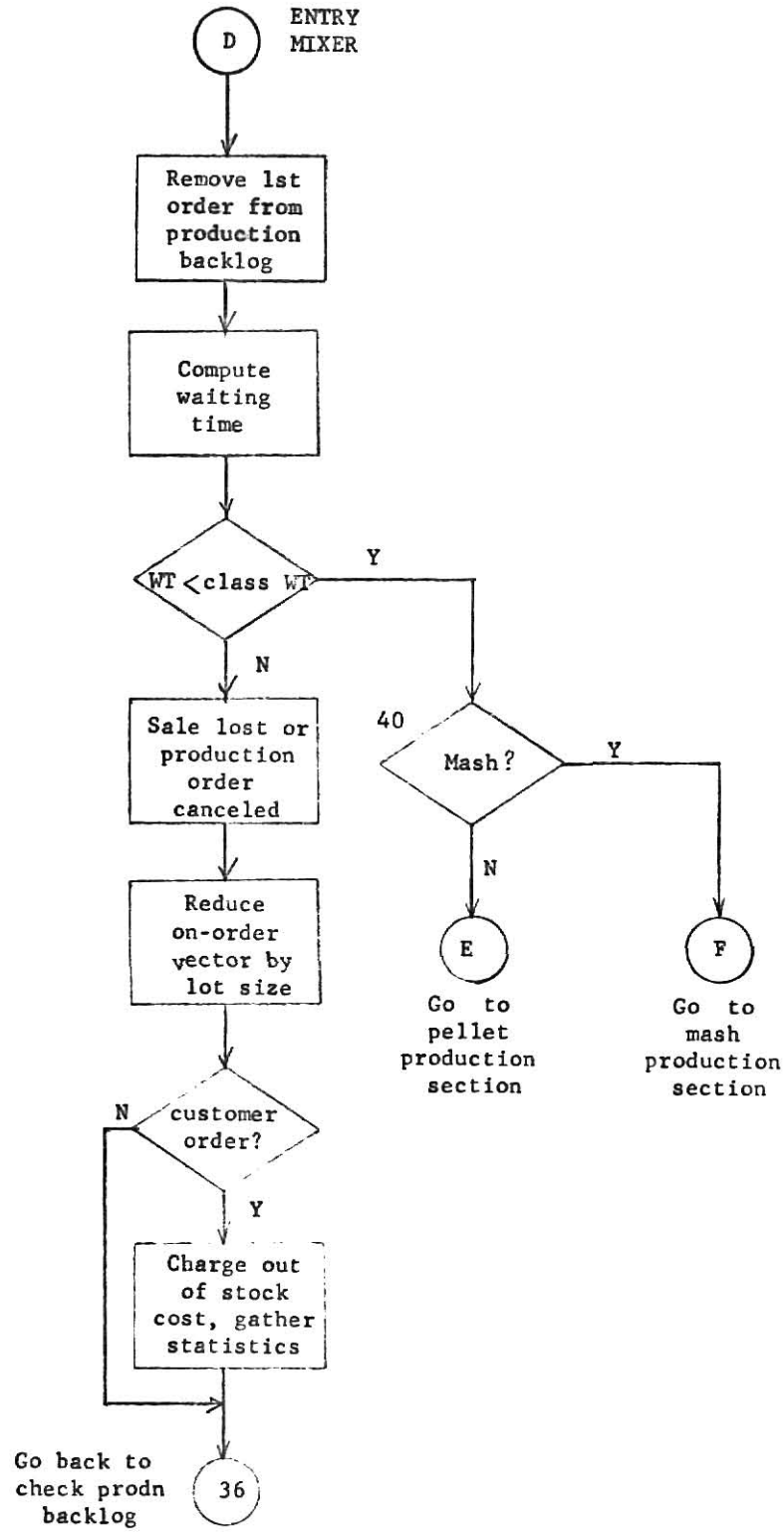



Fig. 3b. Routine MASTER (cont.)

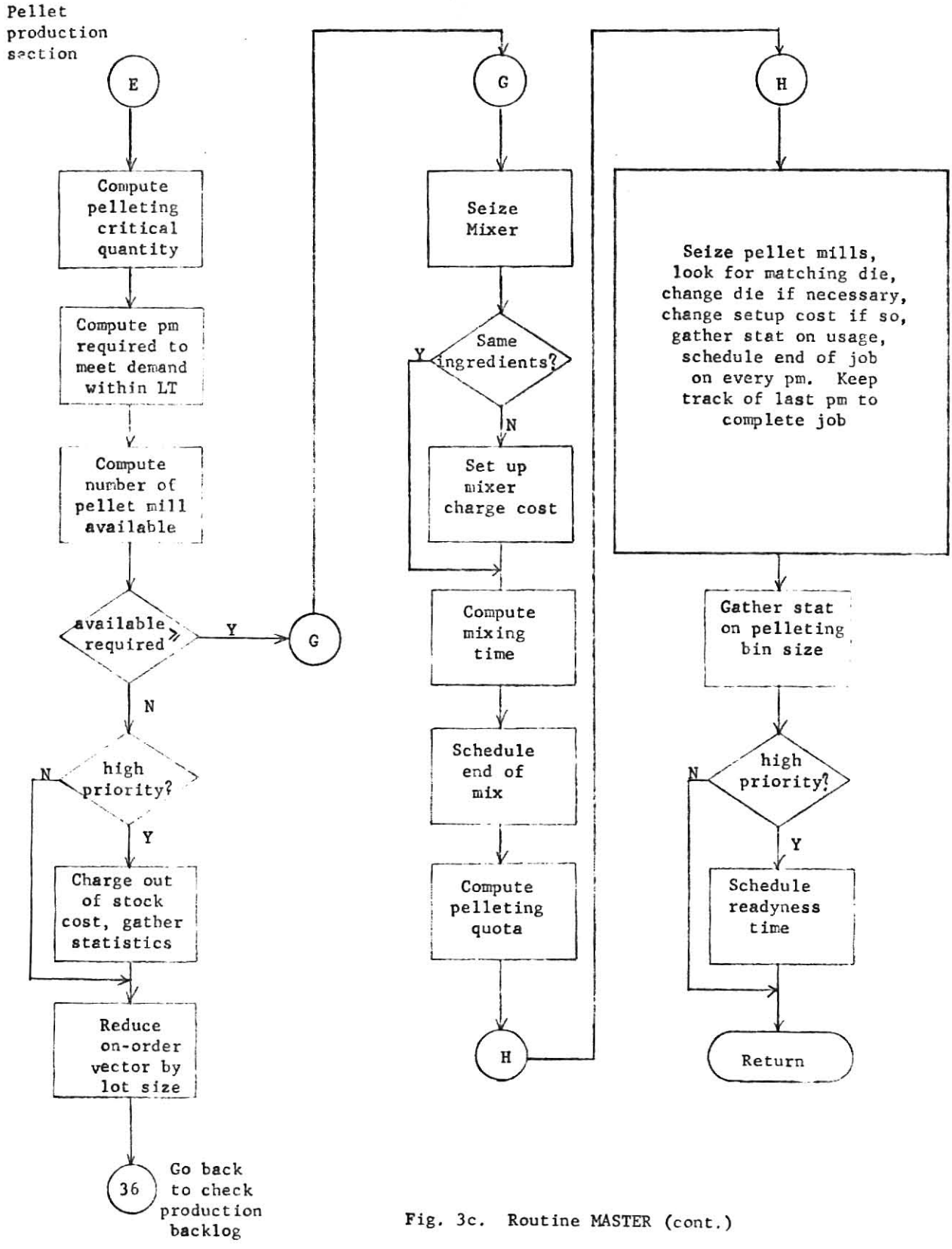



Fig. 3c. Routine MASTER (cont.)

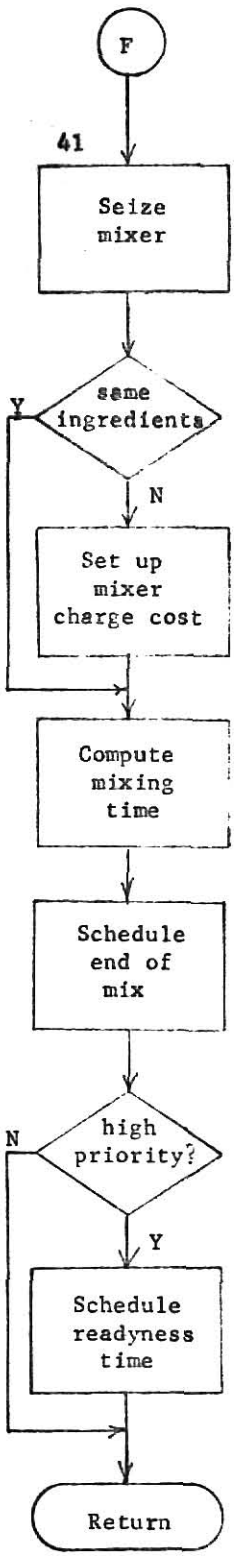



Fig. 3d. Routine MASTER (cont.)

ENDMIX (Fig. 4): this event indicates the end of a run on the mixer. The mixer is released and control is transferred to the section in the inventory simulator that processes the next order from the "production backlog". If there are none, the next event in the time sequence is executed.

ENDJOB (Fig 5): it simulates the completion of all operations on a particular order. The last machines in operation at that moment are freed and relevant statistics are collected. If there is any customer waiting for this type of feed his order is filled. A replenishing order with low priority is placed, if necessary. Then, control is transferred to the section in the inventory simulator (MASTER) that checks whether or not the mixer is idle.

RECORD (Fig. 6): this event is a recording and reporting routine that occurs at the end of the second shift every day. It reports daily values on stock-on-hand, running average stock, running average of stockouts, running average of mixer usage, and daily production and sales.

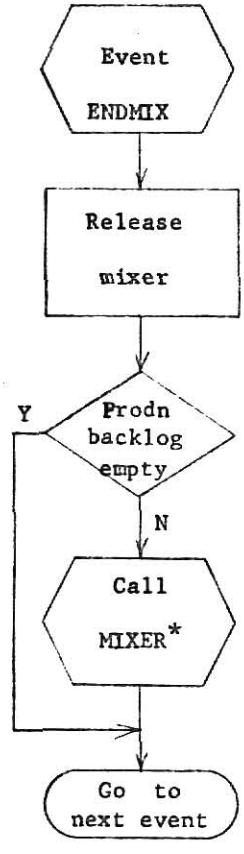
READY (Fig. 7): it indicates the time at which a production order (or part of a production order) to be delivered was completed. Statistics are collected on the time the order spent in the system.

OTPUT (Fig. 8): this routine prints out final simulation reports and plots of inventory costs and production-inventory parameters.

RESET: this routine sets the maximum value in stock to zero after twenty days of operation. It is intended to remove the influence of the starting conditions on the determination of warehouse size.

REPLE: this routine simply makes up a replenishing order and files it in the production backlog.

EVNTS: this routine consist of a simple "computed go to" that switches control to the event routine according to the event type that has been triggered by the executive routine GASP.




(*) Entry point in MASTER

Fig. 4. Event ENDMIX

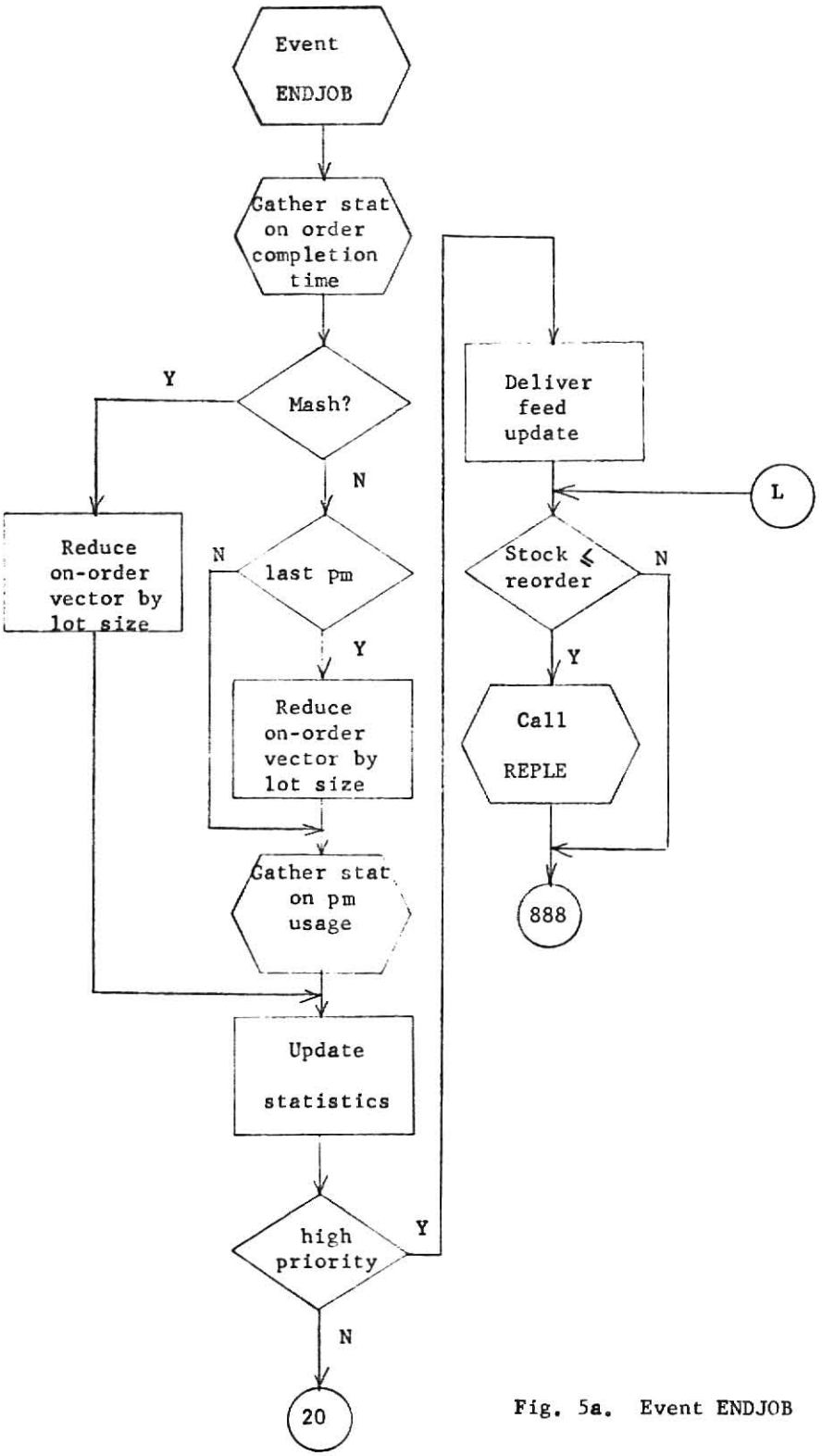



Fig. 5a. Event ENDJOB

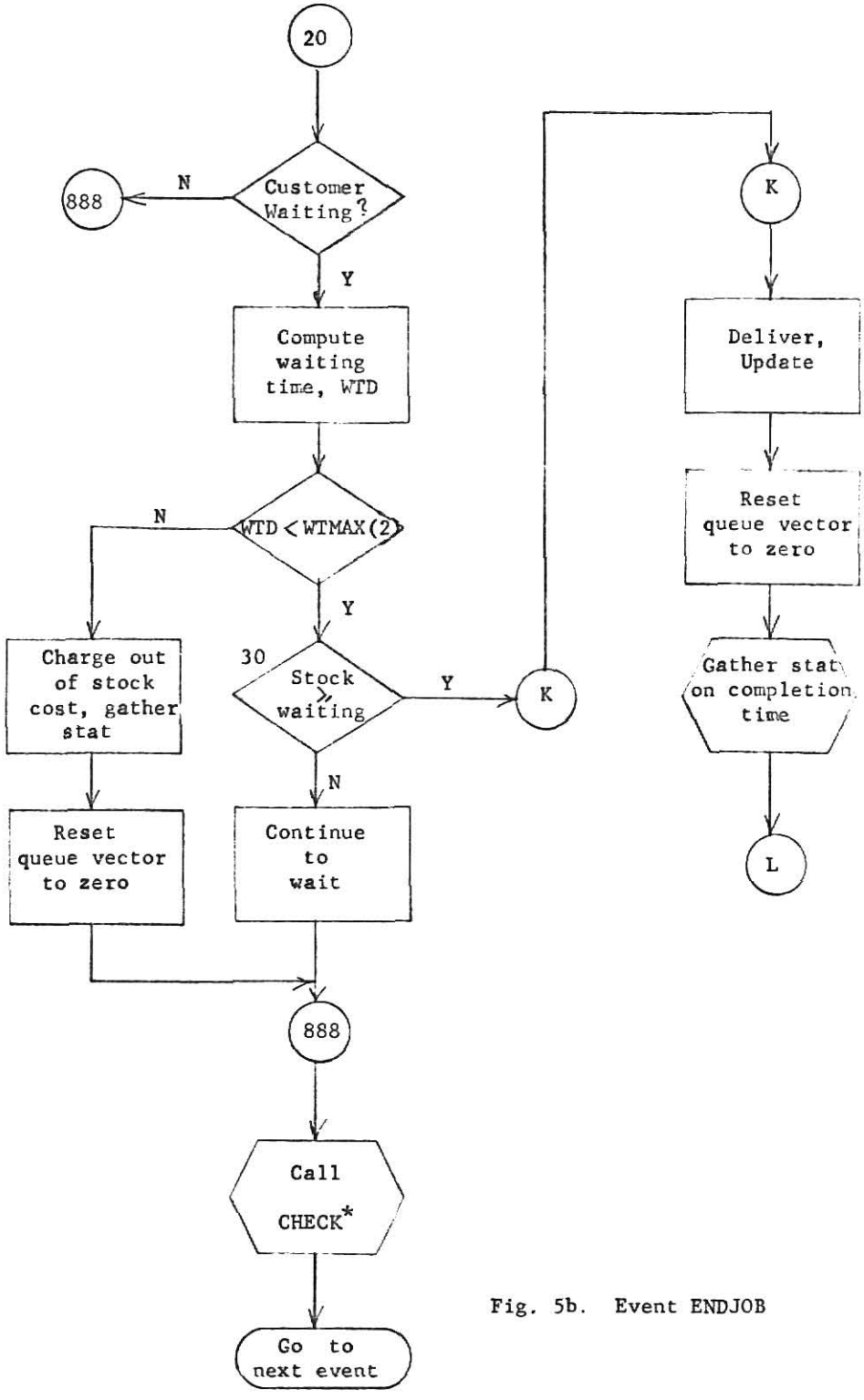



Fig. 5b. Event ENDJOB

(*) Entry point in MASTER

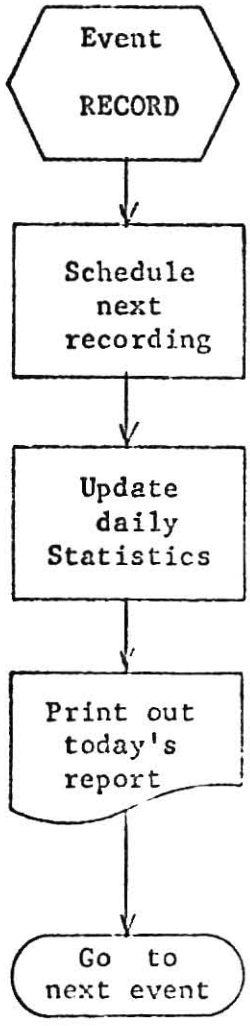



Fig. 6. Event RECORD

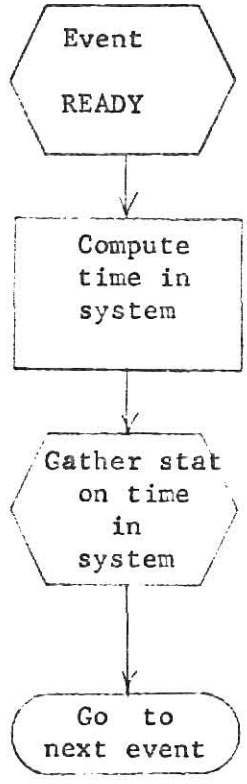



Fig. 7. Event READY

Fig. 8. Event OTPUT

The attributes used to describe every event and to furnish information about orders being processed are illustrated in Tables 1 and 2, according to the file employed.

Uncontrollable Variables

In general, uncontrollable variables are those variables over which the decision maker has not control whatsoever. A typical example of this kind of variable is the demand for finished goods in a non-integrated industrial environment. In our case, the demand for bagged feeds, either pellets, crumbles, or mash. Under economic phenomena where many elements contribute to the process and no seasonal trend is present, it seems safe to assume randomness in the demand process. Otherwise, it is necessary to determine the specific model that the phenomenon can be represented by and follow it when trying to obtain samples of the overall population (i.e. time-series models).

In this case, although it was not tested, the random characteristics of the process were apparent. Therefore, it was possible to use a random generator to extract samples of the variable of interest, that is the daily demand for feed. The random samples were fed into a simulation model of the system under study and its behavior observed.

For this particular model, historical sales data of a specific area were collected during one month covering approximately 10,000 sales transactions over 52 different feeds and more than 150 customers. Date, formula number, and order size (bags) were recorded.

Data processing of the observations was carried out to obtain the following statistical information:

a. Orders per day cumulative distribution, mean, standard deviation.

b. Relative ordering frequency of feeds.

Table 1

FIX POINT ATTRIBUTES (JTRIB'S)

| | Attribute Number | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| File | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1<br>EVENT FILE | event<br>type | | lot<br>size | | formula<br>pointer | pellet<br>mill<br>number | form<br>o-mash<br>≥1-pellet | last<br>pellet<br>mill<br>number |
| 2<br>DEMAND BACKLOG | formula<br>number | | | customer<br>order<br>size<br>(bags) | formula<br>pointer | | form<br>o-mash<br>≥1-pellet | |
| 3<br>PRODUCTION BACKLOG | formula<br>number | priority<br>class<br>1-low<br>2-normal<br>3-rush | lot<br>size<br>(bags) | customer<br>order<br>size<br>(bags) | formula<br>pointer | | form<br>o-mash<br>≥1-pellet | |
| 4<br>PLANT<br>(dummy file) | | | | | | | | |

Table 2

FLOATING POINT ATTRIBUTES (ATRIB'S)

| | Attribute Number | | | | | |
|---|---|---|---|---|---|---|
| File | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 EVENT FILE | event time | | amount to deliver (*) (bags) | amount to store (bags) | | |
| 2 DEMAND BACKLOG | | time order entered system | | | | |
| 3 PRODUCTION BACKLOG | | time order entered system | | | lot size (tons) | customer order (tons) |
| 4 PLANT (Dummy file) | | | | | | |

(*) It can be part of an order if pelleted feed.

c. Order size cumulative distributions, means, standard deviations.

d. Daily demand means and standard deviations.

Samples of this information are found in Appendix C.

Then, random deviate generators were built for the orders-per-day cumulative distribution, relative frequency of ordering, and order-size cumulative distributions according to the techniques for the construction of stochastic generators of discrete variables discussed by Schmidt and Taylor (14) and by Pritsker and Kiviat (13). Daily demand information was used to compute the inventory parameters for the model, namely, lot sizes, safety stocks, and reordering points.

These three groups of generators were included in the event TODAY and used as the source of statistical data to represent the uncontrollable variables in the model.

Controllable Variables

It is possible to group under this section all those variables that in one way or another constitute the underlying frame of the decision process. An adequate combination of these variables determines a rational policy.

The controllable variables used in the model were:

a. Mixer capacity.

b. Number and capacity of pellet mills.

c. Basic mill configuration.

d. Production scheduling rules.

e. Inventory control parameters.

Variables a, b, and c refer to the physical characteristics of the plant itself and in any design stage are manipulated to achieve a proper combination, or in an existing installation are supplied to the model to evaluate

usage rates under simulated conditions. Different values given to these variables throughout the runs performed are described in the section of experimental results.

Production scheduling rules are the set of regulations that will govern the machine assignment and load distribution problems within the model (see routine MASTER). They can be changed in successive runs until a satisfactory effect is obtained. In this case only one scheduling rule was tested: first-come-first-serve, within priority classes. Other alternatives that could be tested are: sequential rule, minimum imminent processing time, maximum imminent processing time and earliest start date (10).

To control the stock in a warehouse, definite inventory control policies must be formulated regarding when to reorder and how much to reorder. Formulating an inventory policy implies considerations about demand rates and demand variability patterns to balance the cost associated with carrying materials in stock and the cost of ordering and setup. On the other hand, carrying a reserve stock to prevent shortages due to demand fluctuations during lead time must counteract the cost of being out of stock.

When processing the raw data, it was intended to fit the daily demand for every feed to one of the known well-behaved distributions (normal, Poisson, exponential, gamma, lognormal) but very little significance was found in most cases. Thus, it was necessary to rely exclusively on the information contained in the first two moments of the unknown distributions, namely the means and the standard deviations. In order to determine lot sizes, reserve stocks, and reordering points, the general method of the Q-model for dynamic inventory problems under uncertainty were employed (16), for the case of a performance standard set by management. A brief summary regarding the theoretical fundamentals of this method is presented in Appendix A.

Measures of Performance and Statistics Collection

To understand the effects that a particular change of policy has on a system, statistics are to be collected on the measures of performances that the model will have as output.

The measures of performance used for this application were the following:

a. Inventory cost (setup, carrying, ordering, out-of-stock).

b. Mixer usage.

c. Pellet mills usage.

d. Warehouse size and usage.

e. Pelleting bin size.

f. Service quality (percentage of stockouts)

Other statistics collected were:

a. Histograms on:

   i) Warehouse order lead time.

   ii) Pellet mill running time.

   iii) Number of pellet mills used per order.

   iv) Pelleting bin size.

   v) Customer order completion time.

b. Mean, Standard deviation, maximum, minimum on:

   i) Customer order completion time.

   ii) Warehouse order lead time.

   iii) Mixer running time.

   iv) Pellet mill running time.

   v) Pelleting bin size.

Table 3

GASP STATISTICS COLLECTION

COLCT Statistics

| Number | Variable | Unit |
|---|---|---|
| 1 | Customer order completion time | 1 hour |
| 2 | Pelleting bin size | 1 ton |
| 3 | Replenishing order completion time | 1 hour |
| 4 | Not used | ---- |
| 5 | Mixer running time | 1 hour |
| 6 | Pellet mill running time | 1 hour |

Table 4

GASP STATISTICS COLLECTION

HISTO Statistics

| Number | Variable | Interval |
|---|---|---|
| 1 | Customer order completion time | 0.01 hour |
| 2 | Bin size | 10 tons |
| 3 | Number of pellet mills/order | 1 p.m. |
| 4 | Pellet mill running time | 1 hr. |
| 5 | Replenishing order completion time | 2 hr. |

TMST Statistics

| Number | Variable | Unit |
|---|---|---|
| 1-9 | Pellet mill status | --- |
| 10 | Time-integrated total stock | tons |

## SIMULATION PROCEDURE AND EXPERIMENTAL RESULTS

The simulation model was initially debugged with rather large capacities for the mixer and the pelleting group, taking into consideration that the pellet mill setup might consume a substantial part of the time available to produce an order. The values of the measures of performance were analyzed from run to run to determine the course of action to follow in successive runs. Three models were tried:

a) Model 1:

mixing capacity = 90 tons/hour

pelleting capacity = 90 tons/hour (9 mills)

b) Model 2:

mixing capacity = 70 tons/hour

pelleting capacity = 60 tons/hour (6 mills)

c) Model 3:

mixing capacity = 60 tons/hour

pelleting capacity = 50 tons/hour (5 mills)

As previously mentioned, the demand for feed (uncontrollable variable) was represented in the models by a series of cumulative probability functions from which samples were drawn to simulate the daily arrival of order to the feed mill (Appendix C).

It was assumed a production capacity of 10 tons/hour for pellet mills due to lack of information about the particular performance ratings per type of feed. It is known that the real performance of pellet mills varies within a considerable range depending upon feed characteristics (volumetric weight, moisture content, particle size--among others) and specific data regarding the

influence of these factors on production rates could easily be included in the model, making it more realistic.

A constant mixing cycle of 6 minutes was used. This includes: loading of ingredients and premix dumping, mixing time, and unloading of the mixer. The mixing cycle could be changed, if desired, by replacing one card in the main program of the simulation (see computer program printout); or converted to a variable depending upon the specific feed, reading in that information. Approximate conveying capacities from the mixer and to either the pelleting group (pelleting bins, mills, sifters, coolers, crumblers) or to the bagging group were provided for every model.

Some basic data regarding sales period, plant configuration, and unitary inventory costs were assumed, as follows:

working year = 260 days

working day = 16 hours

customer maximum waiting time = 8 hours

pellet cooling time (1 batch) = 10 minutes

change-of-die time = 24 minutes

mixer setup time = 6 minutes

mixer setup cost = $5/run

ordering cost (clerical) = $5/order

pellet mill change of formula cost = $5/change

pellet mill change of die cost = $10/change

carrying cost = $10 per ton-year

warehouse cost = $5 per ton-year

As far as the inventory parameters fed to the model were concerned, a Q-model based on a management standard of 10% maximum of stockouts on incoming orders for all feeds was used. Although apparently high, it is necessary to take in

consideration that when following management standards (Appendix A) the Chebyshev's inequality gives a rather conservative protection. Hence, the simulation model could also serve to test how conservative this approach is. Computation of the lot sizes was made according to the classical square root formula assuming the worst scheduling possible, that is that formula changes were always necessary between any two successive runs (assumption based on the fact that the Q-model does not permit convenient grouping of orders according to criteria that may minimize the number of setups). This was to be confirmed by collecting statistics on setup costs during the simulation to compute an average setup cost per order.

Although the unitary setup cost can be considered fixed for every machine, it is obvious that if adequate scheduling is performed, savings in total setup costs in a time period can be obtained. It was found that the actual setup cost per order was slightly lower than the unitary setup cost used for computing the lot sizes. Therefore, lot sizes were kept the same for every feed throughout the different runs, in which, again, very slight variations in setup costs per order were encountered. The inventory parameters employed can be observed in Table 5.

To "debug" the model the following steps were executed:

1. Preliminary runs were made to observe the behavior of the model and to make sure that the interrelationship among the parts of the system followed the real-world logic.

2. The measures of performance were studied throughout time to obtain indications that the system had reached steady-state conditions. This is a basic prerequisite in any simulation study and must be carried out before any conclusions are made in order for the model to be valid (5). One hundred nineteen days were sufficient for this purpose, as can be seen in figures 12, 13, and 14.

# ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

INVENTORY MODEL PARAMETERS

MANAGEMENT STANDARD = 10 % MAXIMUM STOCKOUTS
CHEBYSHEV'S CONSTANT = 3.162

INVENTORY CONTROL UNITARY COSTS
CARRY = 10 $/TON-YR  STORE = 5 $/TON-YR  MIX = 5 $/RUN  PELLET = 5 $/RUN  ORDER = 5 $/RUN

| FORMULA | MEAN BAGS/DAY | STD DEV BAGS/DAY | SAFETY STOCK BAGS | SAFETY STOCK TONS | SAFETY STOCK DAYS | REORDER POINT BAGS | REORDER POINT TONS | LOT SIZE BAGS | LOT SIZE TONS | RUNS/ YEAR | OUT OF STOCK DOLLARS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F-01 | 110.19 | 147.87 | 467 | 18.680 | 4.2 | 577 | 23.080 | 847 | 33.852 | 33.85 | 55.25 |
| F-02 | 182.33 | 204.76 | 647 | 25.880 | 3.5 | 829 | 33.160 | 1089 | 43.546 | 43.55 | 59.48 |
| F-03 | 112.71 | 116.27 | 367 | 14.680 | 3.3 | 479 | 19.160 | 856 | 34.237 | 34.24 | 42.96 |
| F-04 | 109.38 | 172.46 | 545 | 21.800 | 5.0 | 654 | 26.160 | 844 | 33.728 | 33.73 | 64.68 |
| F-05 | 76.90 | 212.87 | 673 | 26.920 | 8.8 | 749 | 29.960 | 708 | 28.280 | 28.28 | 95.21 |
| F-06 | 618.57 | 535.60 | 1693 | 67.720 | 2.7 | 2311 | 92.440 | 2006 | 80.207 | 80.21 | 84.47 |
| F-07 | 31.76 | 35.03 | 110 | 4.400 | 3.5 | 141 | 5.640 | 455 | 18.174 | 18.17 | 24.38 |
| F-08 | 38.67 | 51.16 | 161 | 6.440 | 4.2 | 199 | 7.960 | 615 | 24.561 | 16.37 | 39.52 |
| F-09 | 372.00 | 416.55 | 1317 | 52.680 | 3.5 | 1689 | 67.560 | 1905 | 76.179 | 50.79 | 103.75 |
| F-10 | 178.76 | 185.99 | 588 | 23.520 | 3.3 | 766 | 30.640 | 1078 | 43.117 | 43.12 | 54.56 |
| F-11 | 657.71 | 544.19 | 1720 | 68.800 | 2.6 | 2377 | 95.080 | 2533 | 101.293 | 67.53 | 101.93 |
| F-12 | 1077.33 | 891.09 | 2817 | 112.680 | 2.6 | 3894 | 155.760 | 3241 | 129.639 | 86.43 | 130.42 |
| F-13 | 33.81 | 45.93 | 145 | 5.800 | 4.3 | 178 | 7.120 | 469 | 18.752 | 18.75 | 30.98 |
| F-14 | 631.29 | 503.21 | 1591 | 63.640 | 2.5 | 2222 | 88.880 | 2481 | 99.238 | 66.16 | 96.21 |
| F-15 | 1863.30 | 1445.30 | 4570 | 182.800 | 2.5 | 6433 | 257.320 | 4263 | 170.492 | 113.66 | 160.84 |
| F-16 | 158.86 | 166.87 | 527 | 21.080 | 3.3 | 685 | 27.400 | 1245 | 49.782 | 33.19 | 63.60 |
| F-17 | 241.43 | 201.03 | 635 | 25.400 | 2.6 | 876 | 35.040 | 1535 | 61.370 | 40.91 | 62.15 |
| F-18 | 69.81 | 75.24 | 237 | 9.480 | 3.4 | 306 | 12.240 | 826 | 33.001 | 22.00 | 43.26 |
| F-19 | 54.38 | 113.64 | 359 | 14.360 | 6.6 | 413 | 16.520 | 729 | 29.126 | 19.42 | 74.03 |
| F-20 | 24.14 | 56.41 | 178 | 7.120 | 7.4 | 202 | 8.080 | 486 | 19.406 | 12.94 | 55.15 |
| F-21 | 550.48 | 1591.38 | 5032 | 201.280 | 9.1 | 5582 | 223.280 | 2317 | 92.669 | 61.78 | 325.83 |
| F-22 | 2656.38 | 2186.22 | 6913 | 276.520 | 2.6 | 9569 | 382.760 | 5090 | 203.567 | 135.71 | 203.77 |
| F-23 | 532.48 | 488.84 | 1545 | 61.800 | 2.9 | 2077 | 83.080 | 2279 | 91.141 | 60.76 | 101.77 |
| F-24 | 3325.86 | 2703.28 | 8546 | 341.920 | 2.6 | 11873 | 474.920 | 5695 | 227.779 | 151.85 | 225.18 |
| F-25 | 238.33 | 301.76 | 954 | 38.160 | 4.0 | 1192 | 47.680 | 1525 | 60.975 | 40.65 | 93.90 |
| F-26 | 171.00 | 294.77 | 932 | 37.280 | 5.5 | 1103 | 44.120 | 1292 | 51.649 | 34.43 | 108.29 |
| F-27 | 1155.24 | 1004.71 | 3177 | 127.080 | 2.8 | 4332 | 173.280 | 2741 | 109.611 | 109.61 | 115.94 |
| F-28 | 1005.14 | 853.67 | 2699 | 107.960 | 2.7 | 3704 | 148.160 | 2557 | 102.242 | 102.24 | 105.61 |
| F-29 | 20.00 | 36.33 | 114 | 4.560 | 5.7 | 134 | 5.360 | 361 | 14.422 | 14.42 | 31.86 |
| F-30 | 150.48 | 230.47 | 728 | 29.120 | 4.8 | 878 | 35.120 | 989 | 39.560 | 39.56 | 73.69 |
| F-31 | 2125.86 | 1830.20 | 5787 | 231.480 | 2.6 | 7972 | 318.880 | 3770 | 150.774 | 150.77 | 153.54 |
| F-32 | 412.10 | 364.89 | 1153 | 46.120 | 2.8 | 1565 | 62.600 | 2005 | 80.179 | 53.45 | 86.35 |
| F-33 | 1672.55 | 1497.03 | 4734 | 189.360 | 2.8 | 6406 | 256.240 | 3298 | 131.904 | 131.90 | 143.56 |
| F-34 | 459.71 | 402.46 | 1272 | 50.880 | 2.8 | 1731 | 69.240 | 1729 | 69.145 | 69.14 | 73.62 |
| F-35 | 19.05 | 88.32 | 279 | 11.160 | 14.6 | 298 | 11.920 | 352 | 14.076 | 14.08 | 79.37 |
| F-36 | 654.38 | 626.37 | 1980 | 79.200 | 3.0 | 2634 | 105.360 | 2063 | 82.496 | 82.50 | 96.04 |
| F-37 | 89.05 | 157.56 | 498 | 19.920 | 5.6 | 587 | 23.480 | 761 | 30.432 | 30.43 | 65.49 |
| F-38 | 61.71 | 56.07 | 177 | 7.080 | 2.9 | 238 | 9.520 | 634 | 25.333 | 25.33 | 28.00 |
| F-39 | 239.14 | 269.29 | 851 | 34.040 | 3.6 | 1090 | 43.600 | 1247 | 49.870 | 49.87 | 68.30 |
| F-40 | 50.29 | 61.32 | 193 | 7.720 | 3.8 | 243 | 9.720 | 572 | 22.870 | 22.87 | 33.92 |
| F-41 | 267.43 | 251.44 | 795 | 31.800 | 3.0 | 1062 | 42.480 | 1319 | 52.738 | 52.74 | 60.31 |
| F-42 | 35.86 | 70.46 | 222 | 8.880 | 6.2 | 257 | 10.280 | 592 | 23.652 | 15.77 | 56.52 |
| F-43 | 97.81 | 86.39 | 273 | 10.920 | 2.8 | 370 | 14.800 | 977 | 39.062 | 26.04 | 41.96 |
| F-44 | 205.57 | 169.44 | 532 | 21.280 | 2.6 | 737 | 29.480 | 1416 | 56.629 | 37.75 | 56.44 |
| F-45 | 51.52 | 45.33 | 143 | 5.720 | 2.8 | 194 | 7.760 | 709 | 28.350 | 18.90 | 30.34 |
| F-46 | 112.43 | 108.92 | 344 | 13.440 | 3.1 | 456 | 4.560 | 2094 | 20.940 | 13.96 | 24.67 |
| F-47 | 106.38 | 108.54 | 343 | 13.720 | 3.2 | 449 | 17.960 | 1019 | 40.737 | 27.16 | 50.55 |
| F-48 | 990.76 | 1197.58 | 3787 | 18.935 | 3.8 | 4777 | 23.885 | 8791 | 43.954 | 29.30 | 64.62 |
| F-49 | 800.05 | 995.69 | 3146 | 7.870 | 3.9 | 3948 | 9.870 | 11172 | 27.929 | 18.62 | 42.28 |
| F-50 | 19.95 | 31.63 | 100 | 4.000 | 5.0 | 119 | 4.760 | 442 | 17.641 | 11.76 | 34.02 |
| F-51 | 358.81 | 397.36 | 1256 | 12.560 | 3.5 | 1614 | 16.140 | 3055 | 30.543 | 30.54 | 41.14 |
| F-52 | 15.76 | 44.91 | 142 | 5.680 | 9.0 | 157 | 6.280 | 321 | 12.802 | 12.80 | 44.37 |

AGGREGATE SAFETY STOCK = 2821.3 TONS
EXPECTED RUNS PER DAY = 9.8

Table 5. Inventory Parameters

Once the basic model had been debugged and run (Model 1), its outputs were analyzed to determine possible changes in the controllable variables of the model that might improve the system performance. The same procedure was followed for Model 2 and for Model 3. The results obtained for the three models are summarized in Tables 6, 7, 8, 9, and 10.

All capacities considered here are meant to be installed capacities to serve the given market in a restricted planning horizon. This equipment may or may not be installed in a building that permits future expansion by addition of more equipment. By no means is it considered here that the capacities which this analysis presents as convenient will serve the needs of a future market. Nevertheless, if marketing projections were available, a simulation model with demand input affected by a growth term can be developed taking as a basis the models presented in this paper and a long-term or mid-range facility planning project can be worked out.

## Analysis of Results

The results obtained from Model 1 indicate that the usage of the plant facilities was rather low, or put in another way, the plant was oversized for the market characteristics. For the same reason rather large pelleting bins were necessary to hold the mix coming from the mixer. The scheduling rule determined that in most cases only one pellet mill per order was employed. The mixer running time per production order ranged from 8.4 minutes to almost 3 hours, with an average of 1 hour and a standard deviation of 0.63 hours; while the pellet mills were utilized between 1.77 and 17.08 hours with a mean of 8.05 hours and a standard deviation of 3.66 hours. More details can be observed in Tables 7, 8, 9, and 10 and in the corresponding frequency histograms (Appendix D). No stockout occurred in this model.

Table 6

SIMULATION RESULTS

| Performance measure | Model 1 | Model 2 | Model 3 |
|---|---|---|---|
| Mixer setup cost, dollars | 5,750 | 5,560 | 5,120 |
| Pelleting setup cost, dollars | 3,105 | 3,100 | 2,695 |
| Ordering cost, dollars | 5,905 | 7,045 | 9,035 |
| Out-of-stock cost, dollars | 59 | 2,180 | 17,561 |
| Carrying cost, dollars | 29,909 | 27,698 | 24,302 |
| Total inventory cost, dollars | 44,728 | 45,583 | 58,713 |
| Cost per ton sold, dollars | 0.43 | 0.44 | 0.61 |
| Mixer setup cost, dollars/order | 5.0 | 5.0 | 4.9 |
| Pellet mill setup cost, dollars/order | 4.6 | 4.5 | 4.4 |
| Mixer usage, % of time | 60.7 | 76.7 | 83.5 |
| Stockouts, % of orders | 0.0 | 0.8 | 4.5 |
| Average sales, tons/day | 875.3 | 861.1 | 806.1 |
| Warehouse size, tons | 4,957 | 4,753 | 4,457 |
| Ave. warehouse usage, % | 61.7 | 77.7 | 68.6 |
| Pelleting bin size, tons | 271 | 195 | 190 |

Table 7

SIMULATION STATISTICS

MODEL 1

| Statistic | Mean | Std. dev. | Max. | Min. |
|---|---|---|---|---|
| Customer order completion time, hrs. | 0.011 | 0.232 | 8.39 | 0.00 |
| Replenishing order completion time, hrs. | 11.22 | 5.27 | 20.45 | 0.29 |
| Mixer running time, hrs./run | 1.00 | 0.63 | 2.54 | 0.14 |
| Pellet mills running time, hrs./run | 8.05 | 3.66 | 17.05 | 1.77 |
| Pelleting bin size, tons | 79.07 | 38.95 | 270.5 | 17.7 |

Table 8

SIMULATION STATISTICS

MODEL 2

| Statistic | Mean | Std. dev. | Max. | Min. |
|---|---|---|---|---|
| Customer order completion time, hrs. | 0.082 | 0.599 | 8.12 | 0.00 |
| Replenishing order completion time, hrs. | 12.02 | 5.44 | 31.13 | 0.33 |
| Mixer running time, hrs./runs | 1.30 | 0.81 | 3.26 | 0.18 |
| Pellet mills running time, hrs./runs | 7.78 | 3.66 | 22.78 | 0.47 |
| Pelleting bin size, tons | 62.04 | 31.87 | 195.68 | 3.49 |

Table 9

SIMULATION STATISTICS

MODEL 3

| Statistic | Mean | Std. dev. | Max. | Min. |
|---|---|---|---|---|
| Customer order completion time, hrs. | 0.232 | 1.099 | 8.41 | 0.0 |
| Replenishing order completion time, hrs. | 11.83 | 5.66 | 29.73 | 0.36 |
| Mixer running time, hrs./runs | 1.51 | 0.92 | 4.00 | 0.21 |
| Pellet mills running time, hrs./runs | 7.89 | 4.37 | 22.78 | 1.47 |
| Pelleting bin size, tons | 64.31 | 39.92 | 190.33 | 10.72 |

Table 10

PELLETING REPORT

| Performance measure | Pellet mill number 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| MODEL 1 | | | | | | | | | |
| Usage, % of time | 70.8 | 66.9 | 54.9 | 38.6 | 31.5 | 22.5 | 12.3 | 5.5 | 2.9 |
| Ave. production, t/h | 9.52 | 9.53 | 9.53 | 9.41 | 9.32 | 9.26 | 8.72 | 8.61 | 7.09 |
| MODEL 2 | | | | | | | | | |
| Usage, % of time | 73.7 | 65.0 | 57.1 | 45.0 | 37.1 | 19.9 | | | |
| Ave. production, t/h | 9.50 | 9.43 | 9.30 | 9.27 | 9.30 | 9.26 | | | |
| MODEL 3 | | | | | | | | | |
| Usage, % of time | 74.2 | 67.8 | 57.0 | 43.1 | 27.8 | | | | |
| Ave. production, t/h | 9.47 | 9.51 | 9.50 | 9.43 | 9.16 | | | | |

It was logical then to reduce the plant capacity. In Model 2 a mixing capacity of 70 tons/hour and a pelleting group of 60 tons/hour (6 pellet mills) were employed. As a result of this change, there was a slight increase in the inventory total cost which raised the inventory cost per ton sold from $0.43 to $0.49 due primarily to the occurrence of stockouts in about one percent of the orders received during the period under study (119 working days). A considerable improvement was noticed in plant usage and warehouse utilization. A smaller warehouse was possible and the pelleting bin size was drastically reduced. If the overall system total cost is considered, the economic performance of this type of arrangement is better than what it looks. Although the total inventory cost increased by 2.3 per cent per ton sold, the capacity reduction, specifically the dropping of three pellet mills, will produce substantial savings that offsets the cost increase by a large amount.

When planning Model 3, it was considered advisable to further reduce capacity. A mixing group of 60 tons/hour and a pelleting group of only 5 pellet mills were supplied to the model. The level of stockouts mounted to 4.5 percent of the incoming orders (57.4 tons/day) and the inventory cost reached $0.61 per ton sold. Although the compensating effects of the new reduction in plant capacity might very likely offset the raise in inventory cost, the level of stockouts has climbed to a level which may deteriorate plant service in such a way that a sound judgment would not advise this facility arrangement.

Finally, a series of plots (Figs. 9, 10, 11, 12, 13, and 14) were made to give a pictorial comparison of the three models.

Running one of the models in Fortran IV level G takes approximately 12.5 minutes on the Kansas State University IBM 360/50 computer. Reduction in computer time can be achieved by obtaining an object deck of the debugged model

and simply making changes in the data for successive runs. It was found that running simulation models in Fortran IV level H also speeds up the process by about fifteen percent.

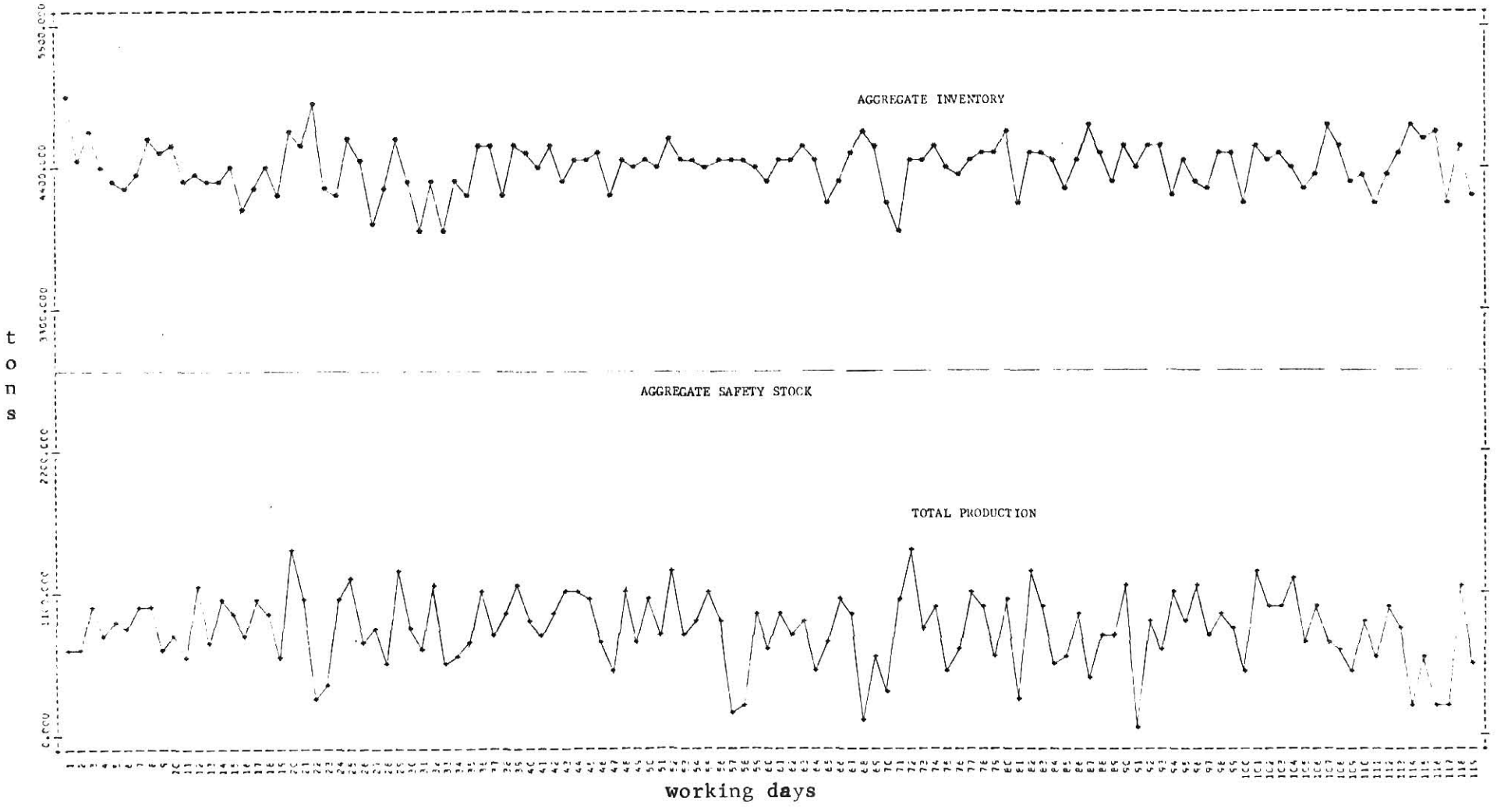



Fig. 9. Production-Inventory. Model 1

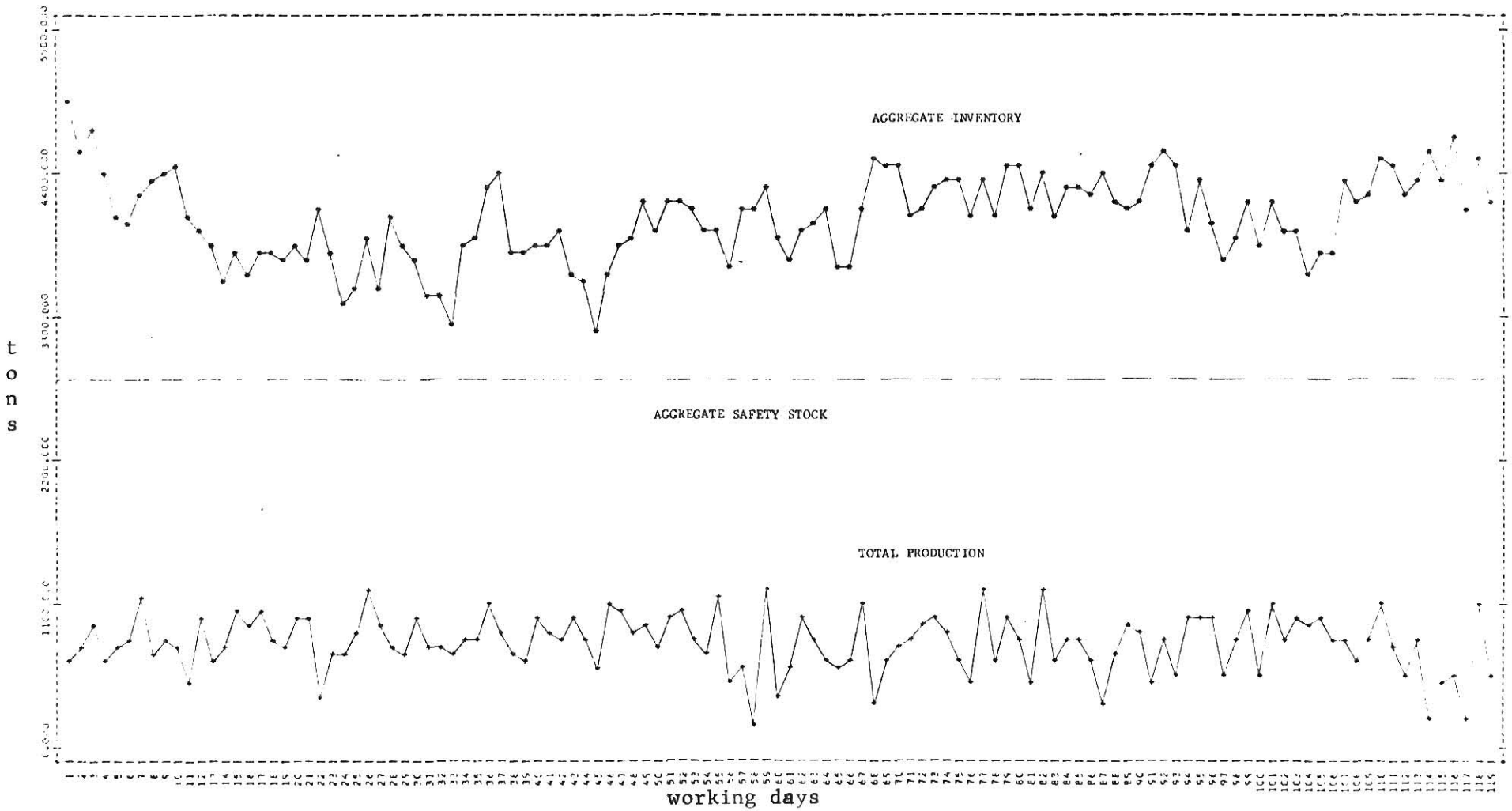



Fig. 10. Production-Inventory. Model 2

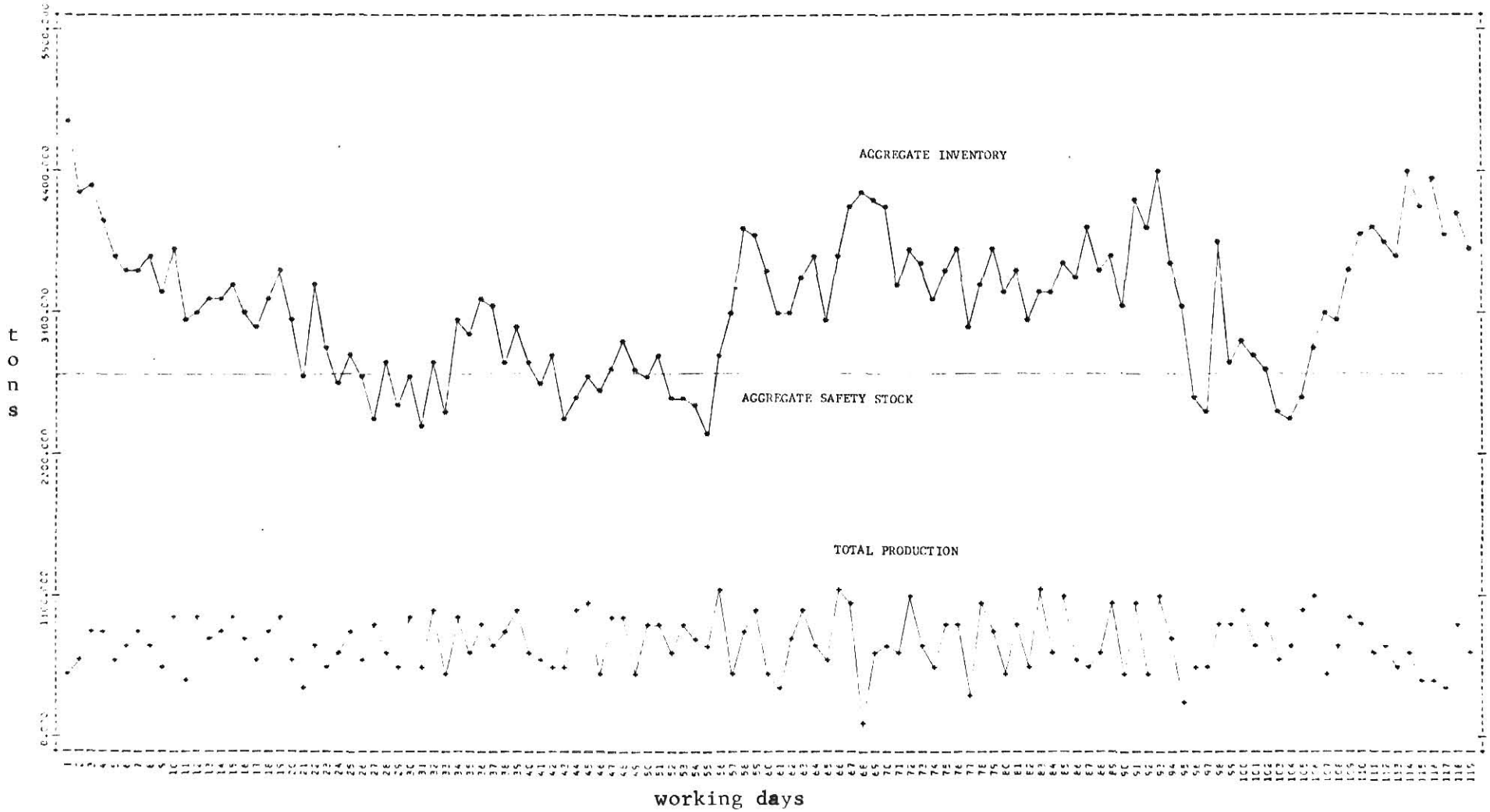



Fig. 11. Production-Inventory. Model 3

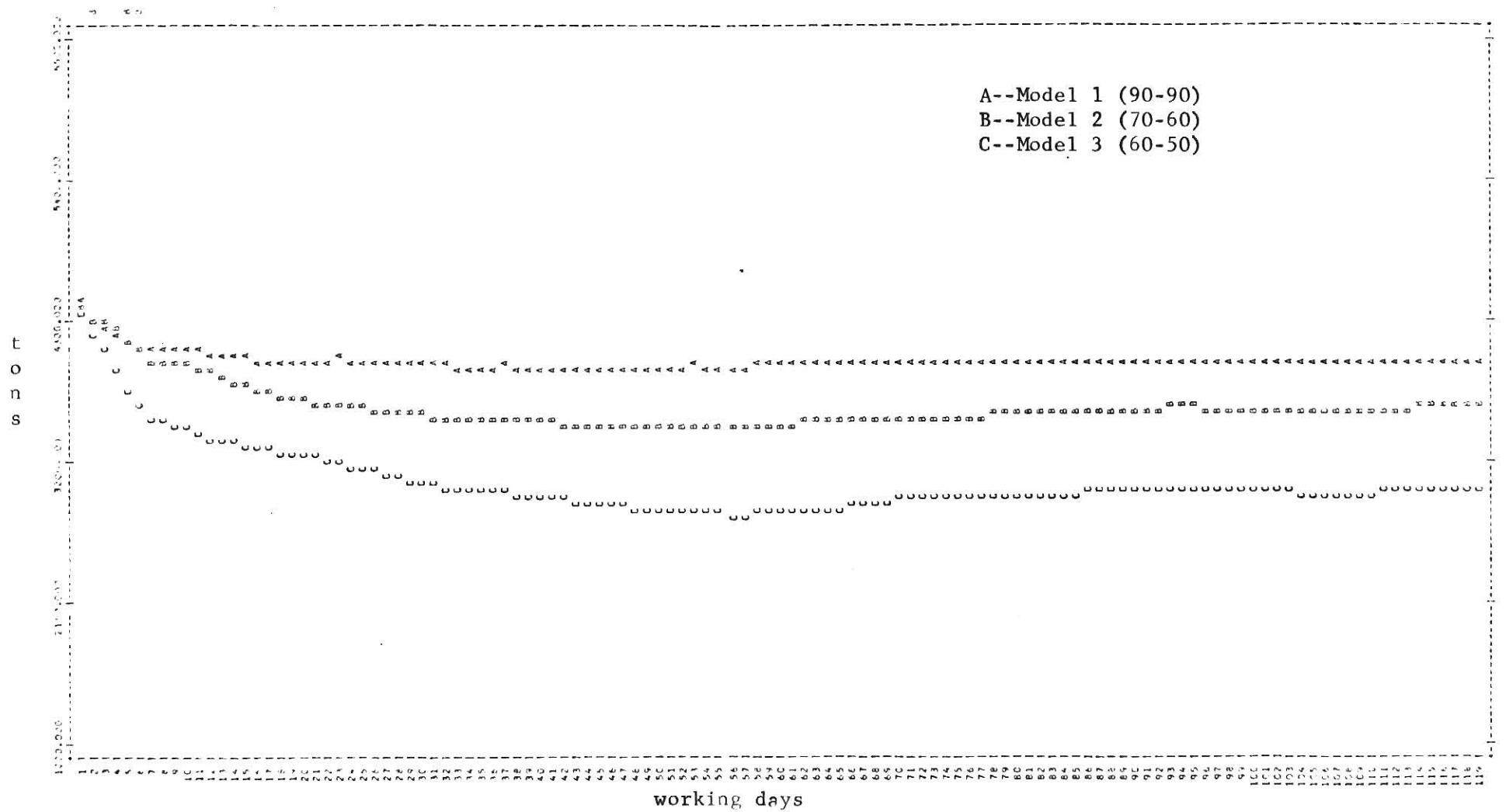



Fig. 12. Running Average on Total Stocks

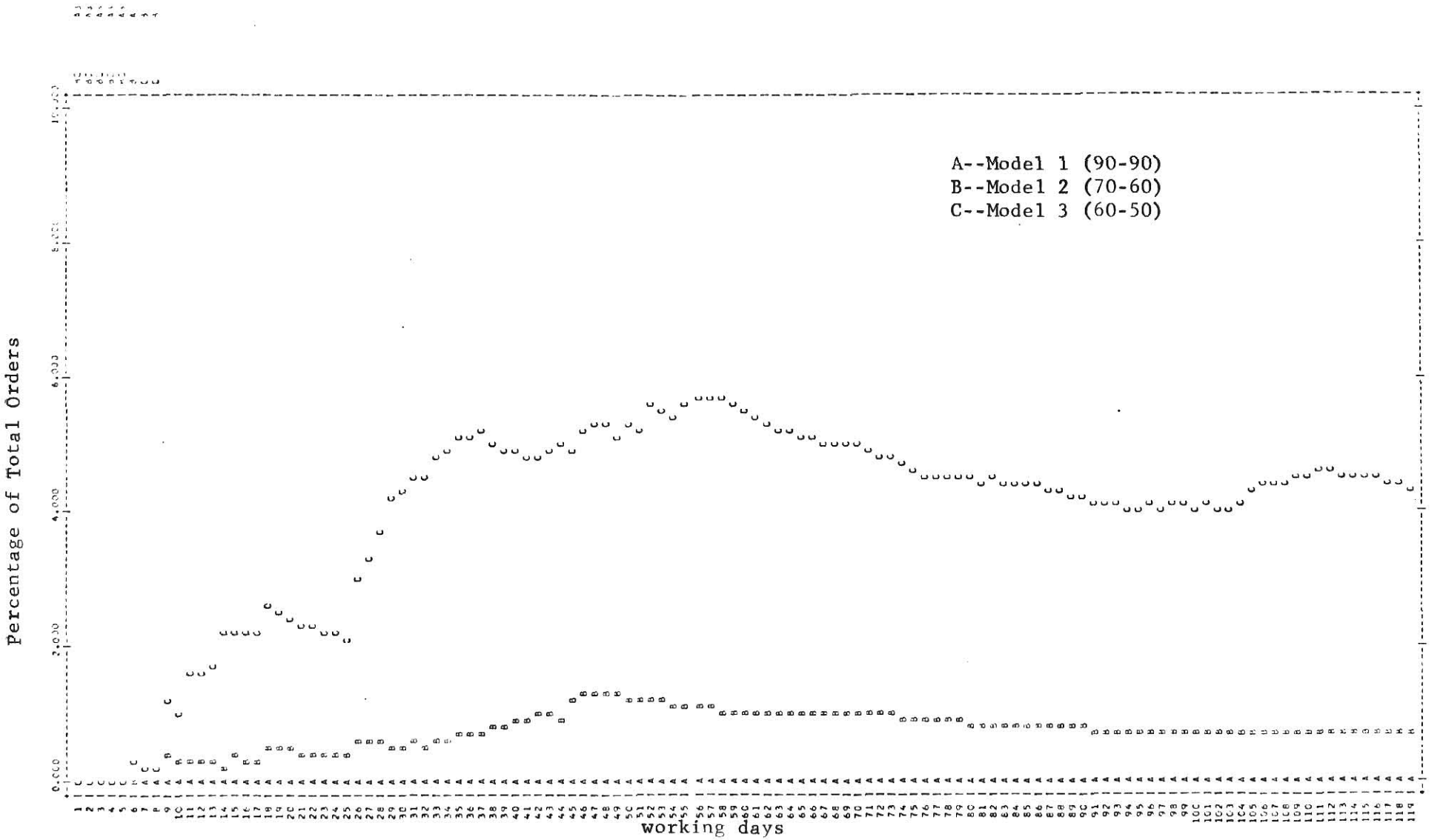



Fig. 13. Running Average on Stockouts Level

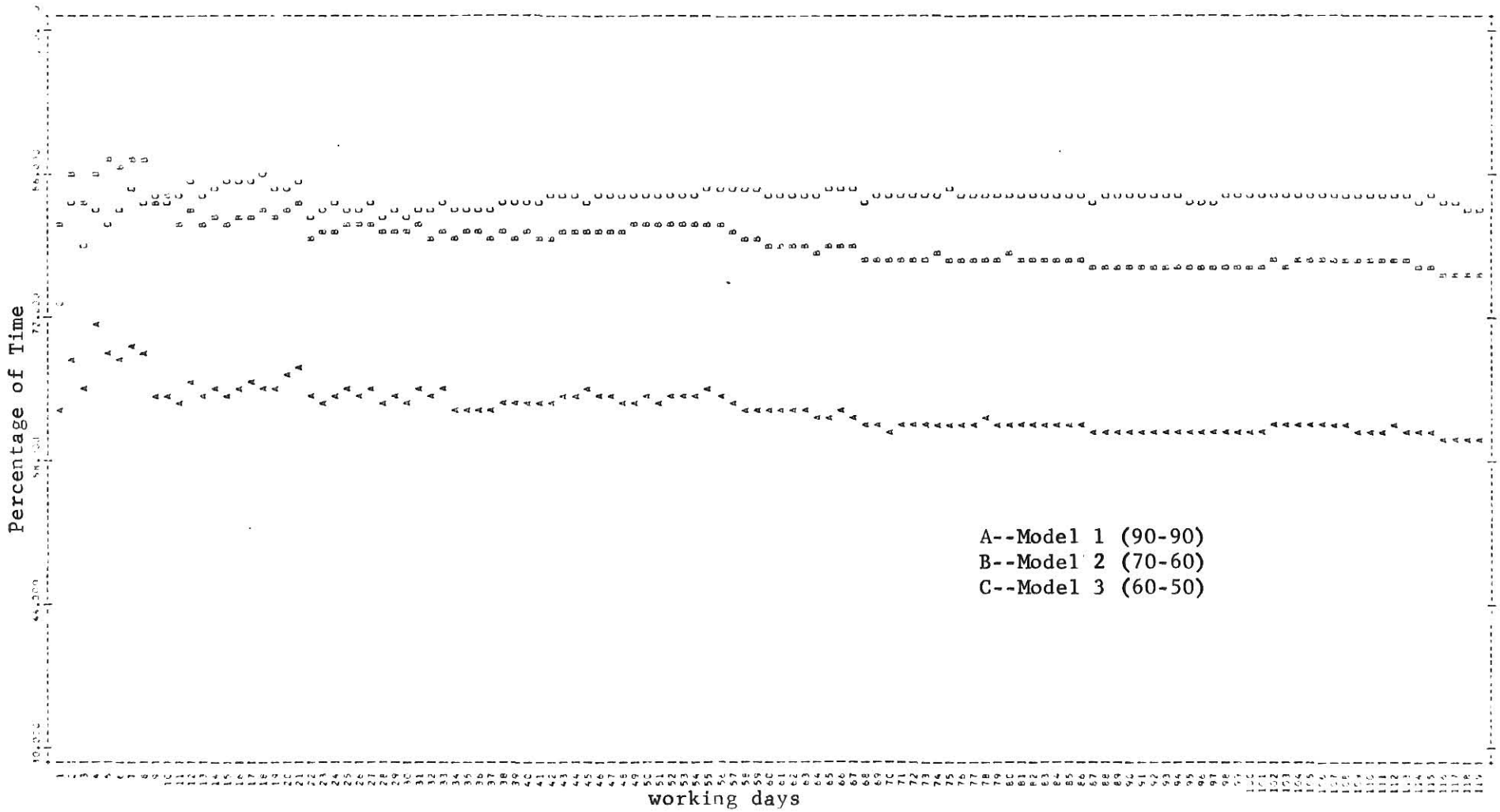



Fig. 14. Running Average on Mixer Usage

## SUMMARY

The main difficulties encountered in developing the simulation model of a feed mill were related to the planning stage and the modeling work of the system. It is of primary importance to define precisely the problems that the model is designed to solve. At the same time clear system boundaries must be defined and all pertinent information either deterministic or probalistic must be listed. Furthermore, those parameters that are under the analyst's control are to be unmistakably susceptible to such a control (controllable variable) and some sort of representation is to be provided for the uncontrollable variables.

It has been shown that analitical techniques of Operations Research such as inventory control can be employed within a simulation model to provide data, policies or general guidelines for the model operational characteristics. This enables the analyst to integrate these techniques even though they may seem specific to particular areas contained within the system under consideration.

Given a model of a feed plant for which key pieces of equipment are specified (mixer, pellet mills) it is possible by simulation to obtain statistical information about the usage of this equipment under specific production scheduling rules, given that the plant serves a specific market. Also, information leading to the design of other equipment (pelleting bins) or facilities (warehouse) can be obtained as a result of the simulation process. The system definition can be broaden to include entities that before were outside of the system and by an aggregation process more information can be gained about a larger system.

There has been a rapid development of specialized simulation languages in the general literature in recent years to facilitate the programmer's task

and reduce model debugging time and cost. In most of the cases, the language used is simply determined by the fact that the analyst knows it. Fortran seems to be a useful general-purpose type of language for simulation work and GASP-II A, a Fortran based simulation package contains a set of routines that facilitates the statistic-collection and event-scheduling activities.

Three different models with decreasing plant capacities were simulated for a random demand process during the same period of time and employing the same statistical samples. Statistics were collected on the measures of performance of the system in every case in order to judge the relative adequacy of the models. In Model 1 the plant capacity appeared to be oversized, while in Model 3 the level of stockouts raised to figures that may seem inconvenient in a real-life situation and may cause market deterioration. Model 2, with an intermediate capacity arrangement, seemed to be the best of the three. To simulate half a year of real operations every model run took about 10 minutes using Fortran IV level H in a IBM 360/50 computer.

If marketing projections were available and simulation studies were feasible in a feed mill, a long-term or mid-range facility planning project can be developed to forecast plant expansion needs in a growing market.

Simulation studies may be rather costly and long. It is suggested that a team approach be employed during the modeling stage. As in all Operation Research activities directed to improve business processes and operations, simulation studies must be economically justifiable.

# ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to his major professor, Dr. Harry B. Pfost, for his advice and assistance in the preparation of this thesis. To Drs. Louis E. Grosh and Charles W. Deyoe for their valuable suggestions and for being in the advisory committee and reviewing the manuscript.

Acknowledgment is also extended to Dr. William J. Hoover, Head of the Department of Grain Science and Industry for providing the means and conditions for the accomplishment of this work.

Many thanks are given to fellow graduate students, Fred R. Anstaett, Victor J. Alvarado, Reuben Kalinhoff, and Oscar A. Romer for providing the opportunities for challenging discussions. To Mrs. Beth Conrady for having the manuscript typed under a rather tight schedule.

Special thanks are given to the firm Protinal C. A., Valencia, Venezuela, and its Directors, for their financial support which made this work possible.

My gratitude to Lilita and Maria-Lila, my wife and daughter, for their moral support and for the hours that rightfully belonged to them.

# BIBLIOGRAPHY


1. Batra, S. "Simulation of Probabilistic Inventory Control Model." Master's Report. Industrial Engineering Department. Kansas State University. 1967.

2. Bowmann, E. H. and Fetter, R. B. "Analysis for Production and Operations Management." Richard D. Irwin, Inc. 1967.

3. Carrillo, R. "Problems of Inventory Control in Feed Mills." Master's Thesis. Grain Science Department. Kansas State University. 1969.

4. Cress, P., Dirksen, P., and Graham, J. W. "Fortran IV With Watfor." Prentice-Hall. 1968.

5. Emshoff, J. R. and Sisson, R. L. "Design and Use of Computer Simulation Models." The Macmillan Company. 1970.

6. Fetter, R. and Dalleck, W. "Decision Models for Inventory Management." Richard D. Irwin. 1961.

7. Hillier, F. S. and Lieberman, G. J. "Introduction to Operations Research." Holden-Day. 1967.

8. Hogg, R. and Craig, A. T. "Introduction to Mathematical Statistics." The Macmillan Company. 1969.

9. Nanda, H. "A Production Scheduling Simulator." Master's Thesis. Industrial Engineering Department. Kansas State University. 1963.

10. Naylor, T. H. et al. "Computer Simulation Techniques." John Wiley & Sons. 1968.

11. Pfost, H. B. (Editor) "Feed Manufacturing Technology." American Feed Manufacturing Association. 1970.

12. Plossl, F. W. and Wight, O. W. "Production and Inventory Control." Prentice-Hall. 1967.

13. Pritsker, A. and Kiviat, P. "Simulation With GASP II: A Fortran Based Simulation Language." Prentice-Hall. 1969.

14. Schmidt, J. W. and Taylor, R. E. "Simulation Analysis of Industrial Systems." Richard D. Irwin. 1970.

15. Stafford, J. H. and Snyder, J. C. "Simulator Program for Feed Manufacturing Inventory Control." Research Report 208. Agricultural Experiment Station. Purdue University.

16. Starr, M. K. and Miller, D. W. "Inventory Control: Theory and Practice." Prentice-Hall. 1962.

17. Sterling, T. and Pollack, S. "Introduction to Statistical Data Processing." Prentice-Hall. 1968.

18. Thierauf, R. "Decision Making Through Operations Research." John Wiley & Sons. 1970.

19. Wagner, H. M. "Principles of Operations Research." Prentice-Hall. 1969.

# APPENDIX A

# APPENDIX A

## Q-SYSTEMS FOR INVENTORY CONTROL

The Q-system of inventory control calls for a fixed order size and a variable order period. In terms of the analysis of this kind of system the basic point is that there is need only for a safety stock to protect against fluctuations in demand during the lag time between order and delivery.

In the discussion that follows about the two methods of solving the Q-model, it will be assumed that the information available regarding the demand distribution is restricted to the first two moments, namely, mean and standard deviation. In other words, the probability distribution is not known, which constitutes the overwhelming majority of everyday inventory problems.

### The Approximate Method

The so-called "approximate" method of analysis for the Q-model (16) consists of determining the quantity to order from the optimal lot-size equation and then determining separately the reserve stock for the lead time (time between placing an order and its delivery).

The optimal lot-size formula that is usually applied in inventory control problems with known demand states that

$$q = \sqrt{\frac{2DC_r}{C_c}}$$

Where,

D = demand during period under study, units

q = optimal lot-size, units

$C_r$ = ordering cost (clerical + setup), dollars per order

$C_c$ = carrying cost, dollars per unit per time period

If warehouse costs (depreciation) are considered in the design stage (6) this equation becomes

$$q = \sqrt{\frac{2DC_r}{C_c + 2C_s}}$$

Where,

$C_s$ = storage space cost, dollar per unit per time period

This formula tends to give a smaller lot-size which ultimately will result in less storage space.

To compute the safety stock size, it is assumed that this will be carried constantly in stock, and, therefore, the carrying cost for it can immediately be expressed on a time period basis. This carrying cost will be balanced against the cost of being out of stock. Alternatively, if the out of stock is difficult to measure, it may be satisfactory to use some performance standard established by management.

Management can set the standard that the level of stock should be sufficient to insure that there is not one out of stock more than a certain number of them, say n, in a hundred lead time periods (n per cent maximum of stockouts). Then, if the mean and standard deviation estimates of the demand during lead time are known, the management policy can be put in a probabilistic manner by using the Chebyshev's inequality, namely,

$$P\ (y - \bar{x} \geqslant ks) \leqslant \frac{1}{k^2}$$

Where,

y = demand during lead time, units

$\bar{x}$ = average demand during lead time, units

s = standard deviation of demand during lead time, units

k = safety stock factor

b = reordering point, units

This probability is to be less than or equal to n/100. Then, the demand during lead time can be computed as:

$$y = \bar{x} + ks \quad \text{(with a confidence greater than } 1 - (\frac{1}{k^2})$$

The minimum safety stock that will ensure the desired protection will be the demand excess over the average demand during lead time, that is

$$r = ks$$

Where,

$$k = \sqrt{\frac{n}{100}}$$

Then,

$$r = \sqrt{\frac{n}{100}}\ s$$

Therefore, the reordering point for the system will be

$$b = \bar{x} + \sqrt{\frac{n}{100}}\ s$$

This approach gives a rather conservative estimate of the safety stock required to meet the standard set by management because of the upper-bound probability given by the Chebyshev's inequality. Judgment should be exercise in determining the adequacy of the reserve stock level determined according to this formulation.

When management defines a specific service policy, it is automatically inputing a stockout cost, as can be seen in the next derivation.

The notation to be used will be as follows.

$TC_r$ = cost of carrying a safety stock, dollars

$N$ = expected number of orders during period

$K_o$ = inputed out of stock cost, dollars per stockout

$y$ = demand during lead time, units

$P(y > b)$ = probability of being out of stock in the lead time interval

The cost of carrying a safety stock can be equated to

$$TC_r = rC_c + NK_oP(y > b)$$

Where,

$$N = \frac{D}{q}$$

Employing the Chebysbev's inequality,

$$TC_r \leqslant ksC_c + NK_o\frac{1}{k^2}$$

and the maximum possible total cost will be

$$TC_r = ksC_c + NK_o\frac{1}{k^2}$$

In order to minimize this maximum with regard to k (safety stock factor) it must follow that

$$\frac{dTC_r}{dk} = 0$$

$$\frac{dTC_r}{dk} = sC_c - \frac{2NK_o}{k^3} = 0$$

Then, $TC_r$ (Max) will be minimum for

$$k = \sqrt[3]{\frac{2NK_o}{sC_c}} = \sqrt[3]{\frac{2DK_o}{qsC_c}}$$

Hence

$$K_o = \frac{sC_c q k^3}{2D} \quad \text{(dollars per stockout)}$$

This is the out of stock cost that the policy set by management (given by k) inputs to every stockout. Here, again, is up to the management to judge the correction of this magnitude.

The Exact Method

For the exact solution it is necessary to write one total cost equation which expresses both ordering cost and safety stock costs (3), as follows,

$$TC = \frac{D}{q}C_r + (\frac{q}{2} + b)C_c + (q + b)C_s + \frac{D}{q}K_oP(y > b)$$

Where,

$\frac{D}{q}C_r$ = ordering cost

$(\frac{q}{2} + b)C_c$ = expected carrying cost

$(q + b)C_s$ = storage cost

$\frac{D}{q}K_oP(y > b)$ = out of stock cost

Considering the case of a performance established by management, we can use again the Chebyshev's inequality.

$$TC \leqslant \frac{D}{q}C_r + (\frac{q}{2} + \bar{x} + ks)C_c + (q + \bar{x} + ks)C_s + \frac{D}{q}K_o\frac{1}{k^2}$$

The expression on the right-hand side of this equation is the maximum possible total cost for a given set of values for q and k. Then, to minimize this maximum cost it must follow that,

$$\frac{\partial TC}{\partial k} = 0 \quad \text{and} \quad \frac{\partial TC}{\partial q} = 0$$

or

$$s(C_c + C_s) - 2\frac{DK_o}{qk^3} = 0$$

$$\frac{C_c}{2} + C_s - \frac{D}{q^2}(C_r + \frac{K_o}{k^2}) = 0$$

This system of equations in q and k can be solved by a numerical procedure and the parameters of the system be known. Although more cumbersome than the "approximate" solution, this method can provide a system total cost which is closer to the true optimal.

Inventory Control for Perishable Products

It is of special interest to consider the situation of perishable products in a feed mill when setting up any inventory control system. In fact, all feedstuffs are perishable within nutritional constraints.

The application of techniques for computing the Q-model parameters, namely, reordering point and lot-size, is to be revised in order to protect against product spoilage.

Using the "approximate" method of solution to compute the lot-size, it is possible to determine the expected time between orders as the ratio

of the period length and the expected number of orders during the period. The period length is the same as the planning horizon that is used (a year, a month, etc.). The expected number of orders in the period is simply the total demand during the planning horizon divided by the optimal lot-size.

If the expected time between orders is shorter than the product shelf-life, the probability of spoilage can be neglected, provided that proper rotation of stocks is employed (first in-first out). Either the "approximate" method or "exact" method of solution will provide satisfactory results.

On the other hand, if the expected time between orders is greater than the shelf-life, measures should be taken to reduce spoilage to a minimum by ordering smaller quantities more frequently. Under these circumstances, the "approximate" solution is no longer valid and both the safety stock (which automatically defines the reordering point) and the lot-size must be determined simultaneously. This can be done by employing the "exact" approach, adding a spoilage-cost term to the general total cost equation.

A notation for the new term to be entered in the total cost equation will be defined as:

$z$ = demand during shelf-life, units

$f(z)$ = probability distribution function (pdf) of the demand during shelf-life

$K_s$ = spoilage unitary cost (finished feed cost), dollars per unit

When the consumption during shelf-life is less than the total made up by the reordering point and the lot-size, spoilage will occur. It will vary from zero if this total is exhausted up to a maximum where there is no consumption at all.

For a particular set of values for the reordering point "b" and lot-size "q", the amount that will spoil for a given level of demand "z" is

determined by the product $(b + q - z)\ f(z)$, where $f(z)$ is the corresponding ordinate (probability) at that level in the distribution of the demand during shelf-life. In order to find the expected total cost for spoilage, we have to integrate overall levels of demand during shelf-life which would cause feed to spoil, namely, all z's for which $z < b + q$, each of them multiplied by the probability of that level of demand. Mathematically,

$$\text{Expected spoilage cost} = K_s \int_{o}^{b+q} (b + q - z)\ f(z)dz$$

This cost applies to one shelf-life period. Then, if the planning horizon contains $n_s$ shelf-life periods, we can easily compute and include this spoilage cost in the total cost equation for the planning horizon.

The out of stock cost, in turn, can be expressed in terms of the distribution of the demand during lead time $f(y)$ as follows

$$\text{Out of stock cost} = \frac{D}{q}K_o \int_{b}^{\infty} f(y)dy$$

Then, including both terms, out of stock and spoilage cost, the total equation becomes

$$TC = \frac{D}{q}C_r + (\frac{q}{2} + b)C_c + (q + b)C_s + \frac{D}{q}K_o \int_{b}^{\infty} f(y)dy + n_s K_s \int_{o}^{b+q} (b + q - z)\ f(z)dz$$

Carrillo (3) suggested that if enough information were available to fit the demand distributions to either one of the typical distributions

(Poisson, Experimental, gamma, normal, lognormal) or to a set of polynomials, the method to optimize the total cost would be to differentiate this equation with respect to every one of the variables (reordering point and lot-size) and then equate the resulting equations to zero. A set of simulataneous equations will result, which would have to be solved by numerical approximation due to the nonlinearity of the terms involved.

Taking partial deviatives (*) with respect to "b" and "q" and equating to zero, the resultant system of equations is as follows,

$$\frac{\partial TC}{\partial q} = \frac{C_c}{2} + C_s + \frac{D}{q^2}\left[K_o F(b) - K_o - C_r\right] + n_s K_s F(b + q) = 0$$

$$\frac{\partial TC}{\partial b} = C_c + C_s - \frac{D}{q} K_o f(b) + n_s K_s F(b + q) = 0$$

Where,

$f(b)$ = probability value at $z = b$ in $f(y)$

$F(b)$ = cumulative probability up to $z = b$ in $f(z)$

$F(b + q)$ = cumulative probability up to $z = b + q$ in $f(z)$

---

(*) The Leibnitz rule is applied for differentiating under integral sign.

# APPENDIX B

MAIN

```
0001          DIMENSION NSET(11000),CSET(6000)
0002          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0003          COMMON/LILITA/IRECORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0004          NCRDR=1
0005          NPRNT=3
        C                                NUMBER OF FEEDS UNDER STUDY
0006          NFEEDS=52
        C                                NUMBER OF PELLET MILLS
0007          NPEL=9
        C                                MIXER CAPACITY,T/H
0008          RATIO=90.
        C                                CYCLE TIME,HRS
0009          CYCLE=0.10
        C                                ONE BATCH,TONS
0010          BEGIN=RATIO*CYCLE
        C                                CLEAR COUNTERS
0011          DO 77 I=1,NFEEDS
        C                                ITH FEED SALES,BAGS
0012          SALE(I)=0
        C                                STOCKOUTS
0013          KOUTS(I)=0
        C                                ON-ORDER VECTOR,BAGS
0014          KORDER(I)=0
        C                                WAITING ORDER,BAGS
0015          IWAIT(I)=0
        C                                DEMAND COUNTER
0016          IDEMAN(I)=0
        C                                WAITING-ORDER FORM ATTRIBUTE
0017       77 IWDIE(I)=0
0018          DO 78 I=1,NPEL
        C                                PELLET MILL STATUS 0-IDLE 1-BUSY
0019          MILL(I)=0
        C                                LAST MIX ON PELLET MILL
0020          ILAST(I)=0
        C                                CUMULATIVE PRODUCTION,TONS
0021       78 TONMIL(I)=0.
        C                                TOTAL STOCKOUTS
0022          NOUTS=0
        C                                CUMULATIVE TOTAL PRODUCTION,TONS
0023          TOTPRO=0.
        C                                CUMULATIVE TOTAL SALES,TONS
0024          TOTSAL=0.
        C                                MIXER RUNS
0025          NMIXER=0
        C                                PELLET MILL RUNS
```

```
0026          NMILL=0
        C                                TOTAL DEMAND COUNTER
0027          NDEMAN=0
        C                                CUMULATIVE LOST SALES,TONS
0028          XLOST=0.
        C                                CUMULATIVE MIXER SETUP TIME,HRS
0029          SETMIX=0.
        C                                CUMULATIVE ORDERING COST,$
0030          ORDER=0.
        C                                CUMULATIVE MIXER SETUP COST,$
0031          SETUP1=0.
        C                                CUMULATIVE PELLET MILL SETUP COST,$
0032          SETUP2=0.
        C                                CUMULATIVE OUT-OF-STOCK COST,$
0033          COUT=0.
        C                                UNIT ORDERING COST,$/ORDER
0034          COR=5.
        C                                UNIT OUT-OF-STOCK COSTS,$/STOCKOUT
0035          READ(NCRDR,1000)(COS(I),I=1,NFEEDS)
        C                                PM CHANGE OF FORMULA COST,$/CHANGE
0036          CMILL=5.
        C                                PM CHANGE OF DIE COST,$/CHANGE (ADDITIONAL)
0037          CDIE=5.
        C                                MIXER SETUP COST,$/SETUP
0038          CMIX=5.
        C                                HOLIDAYS/WORKING DAYS RATIO
0039          WORKNO=8./22.
        C                                CARRYING COST,$/TON-DAY
0040          CCARRY=10./360.
        C                                STORAGE COST,$/TON-DAY
0041          CSTORE=5./360.
        C                                DATE COUNTER
0042          IDATE=0
        C                                INITIALIZE LAST-MIX-ON-MIXER VARIABLE
0043          MIXOLD=111
        C                                1ST BATCH+CONVEYING TIME,HRS
0044          CONVP=1.5*CYCLE
        C                                CONVEY TO BAGGER,HRS
0045          CONVB=0.05
        C                                COOLING TIME,HRS
0046          COOL=0.166
        C                                INITIALIZE DIE SIZE ON PELLET MILLS
0047          DO 1001 I=1,NPEL
0048     1001 IDIE(I)=2
        C                                MINIMUN TO START PELLETING,TONS
0049          BASE=3.
        C                                NUMBER OF PLOTS
0050          KGRAPH=6
        C                                GRAPH TITLES
0051          READ(NCRDR,601)((JNOTE(I,J),J=1,80),I=1,KGRAPH)
        C                                NUMBER OF CARDS
0052          ICARD=NFEEDS/10*10+10
        C                                ORDERS/DAY CDF
0053          READ(NCRDR,10)(AORD(I),I=1,20)
        C                                TYPE CDF
0054          READ(NCRDR,12)(ATYPE(I),I=1,ICARD)
        C                                ORDERS/DAY DEVIATES,TYPE DEVIATES
0055          READ(NCRDR,11)(NORD(I),I=1,20),(NTYPE(I),I=1,ICARD)
```

```
C                                   SIZE CDF
      DO 15 I=1,NFEEDS
   15 READ(NCRDR,10)(ASIZE(I,J),J=1,40)
C                                   SIZE DEVIATES,BAGS
      DO 16 I=1,NFEEDS
   16 READ(NCRDR,11)(NSIZE(I,J),J=1,40)
C                                   LEAD TIMES,HRS
      READ(NCRDR,98)(WTMAX(I),I=1,2)
C                                   BAG SIZES,TON/BAG
      READ(NCRDR,12)(BAG(I),I=1,ICARD)
C                                   LOT SIZES,REORDER POINTS,BAGS
      READ(NCRDR,11)(LOT(I),I=1,ICARD),(IREORD(I),I=1,ICARD)
C                                   INITIALIZE STOCK ON HAND
      STOCK=0.
      DO 56 I=1,NFEEDS
C                                   START AT AVERAGE INVENTORY CONDITIONS
C                                   ITH FEED STOCK,BAGS
      KSTOCK(I)=IREORD(I)+0.50*LOT(I)
C                                   TOTAL STOCK,TONS
      STOCK=STOCK+KSTOCK(I)*BAG(I)
   56 CONTINUE
C                                   CALL GASP EXECUTIVE
      CALL GASP(NSET,QSET)
C                                   QUIT
      STOP
 1000 FORMAT(10F5.2)
  601 FORMAT(80A1)
   10 FORMAT(10F5.3)
   11 FORMAT(10I5)
   12 FORMAT(10F6.4)
   98 FORMAT(3F5.0)
      END
```

```
0001          SUBROUTINE EVNTS(K,NSET,QSET)
0002          DIMENSION NSET(1),QSET(1)
0003          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0004          COMMON/LILITA/IRECORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0005          GO TO (1,2,3,4,5,6,7),K
0006        1 CALL TODAY(NSET,QSET)
0007          RETURN
0008        2 CALL ENDJOB(NSET,QSET)
0009          RETURN
0010        3 CALL RECORD(NSET,QSET)
0011          RETURN
0012        4 CALL FIRST(NSET,QSET)
0013          RETURN
0014        5 CALL ENDMIX(NSET,QSET)
0015          RETURN
0016        6 CALL READY(NSET,QSET)
0017          RETURN
0018        7 CALL RESET
0019          RETURN
0020          END
```

```
0001          SUBROUTINE TODAY(NSET,QSET)
0002          DIMENSION NSET(1),QSET(1)
0003          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRLN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             5C,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0004          COMMON/LILITA/IREORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
        C                                UPDATE THE DATE
0005          IDATE=IDATE+1
        C                                RESET DAILY PRODN & DAILY SALES
0006          ASALE=0.
0007          SUMPRO=0.
        C                                SCHEDULE NEXT DAY TWO 8-HOUR SHIFTS LATER
0008          ATRIB(1)=TNOW+16.00
0009          CALL FILEM(1,NSET,QSET)
        C                                GENERATE TODAY'S NUMBER OF ORDERS
0010          XONE=DRAND(1)
0011          DO 10 I=1,20
0012          IF(XONE.LE.AORD(I))GO TO 15
0013       10 CONTINUE
0014          CALL ERROR(15,NSET,QSET)
0015       15 IORD=NORD(I)
        C                                RESET GROUPING VECTOR
0016          DO 20 N=1,NFEEDS
0017       20 IFEED(N)=0
        C                                GENERATE ORDER ATTRIBUTES
0018          DO 170 M=1,IORD
        C                                GENERATE FEED TYPE
0019          XTWO=DRAND(2)
0020          DO 160 J=1,NFEEDS
0021          IF(XTWO.LE.ATYPE(J))GO TO 200
0022      160 CONTINUE
0023          CALL ERROR(200,NSET,QSET)
0024      200 ITYPE=J
        C                                GENERATE ORDER SIZE,BAGS
0025          XTHREE=DRAND(3)
0026          DO 180 K=1,40
0027          IF(XTHREE.LE.ASIZE(ITYPE,K))GO TO 280
0028      180 CONTINUE
0029          CALL ERROR(280,NSET,QSET)
0030      280 ISIZE=NSIZE(ITYPE,K)
        C                                GROUP BY TYPES
0031          IFEED(ITYPE)=IFEED(ITYPE)+ISIZE
0032      170 CONTINUE
        C                                FILE IN DEMAND BACKLOG,LVF ON SIZE
0033          ATRIB(2)=TNOW
0034          DO 126 I=1,NFEEDS
        C                                IF NO ORDER,CHECK NEXT TYPE IN SEQUENCE
```

```
0035          IF(IFEED(I).EQ.0)GO TO 126
        C                                FORMULA NUMBER
0036          JTRIB(1)=NTYPE(I)
        C                                TODAY'S TOTAL DEMAND FOR ITH FEED,BAGS
0037          JTRIB(4)=IFEED(I)
        C                                POINTER
0038          JTRIB(5)=I
        C                                FORM ATTRIBUTE 0-MASH GE THAN 1-PELLET
0039          JTRIB(7)=(NTYPE(I)-(NTYPE(I)/100)*100)/10
        C                                FILE
0040          CALL FILEM(2,NSET,QSET)
0041      126 CONTINUE
        C                                IF DEMAND BACKLOG EMPTY,GO TO NEXT EVENT
0042      726 IF(NQ(2).EQ.0)GO TO 727
        C                                OTHERWISE,REMOVE FIRST ORDER FROM BACKLOG
0043          MFE2=MFE(2)
0044          CALL RMOVE(MFE2,2,NSET,QSET)
        C                                CALL INVENTORY SIMULATOR
0045          CALL MASTER(NSET,QSET)
        C                                GO TO CHECK DEMAND BACKLOG
0046          GO TO 726
        C                                GO TO NEXT EVENT
0047      727 RETURN
0048          END
```

```
0C01            SUBROUTINE MASTER(NSET,QSET)
0C02            DIMENSION NSET(1),QSET(1)
0C03            COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
               1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
               2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
               3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
               4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
               50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0C04            COMMON/LILITA/IREORD(100),WTMAX(5),A(10,360),
               1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
               2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
               3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
               4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
               5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
               6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
               7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
               8 NMILL,NMIXER,WORKNO
0C05          1 IJ=JTRIB(5)
0C06            NDEMAN=NDEMAN+1
0C07            IDEMAN(IJ)=IDEMAN(IJ)+1
          C                                CHECK STOCK ON HAND + ON ORDER VS. ORDER
0C08            IF((KSTOCK(IJ)+KORDER(IJ)).GT.JTRIB(4))GO TO 10
          C                                IF LOWER, PLACE LOT, ASSIGN HIGH PRIORITY
0C09            JTRIB(3)=LOT(IJ)
0C10            JTRIB(2)=2
          C        CORRECT LOT SIZE IF NECESSARY
0C11            IF(JTRIB(4).GT.JTRIB(3))JTRIB(3)=JTRIB(4)
0C12            ATRIB(5)=JTRIB(3)*BAG(IJ)
0C13            ATRIB(6)=JTRIB(4)*BAG(IJ)
0C14            GO TO 11
          C                                OTHERWISE, CHECK STOCK ON HAND VERSUS ORDER
0C15         10 IF(KSTOCK(IJ).GE.JTRIB(4))GO TO 12
          C                                IF LOWER, JOINT DELIVERY WAITING LINE
0C16            WAIT(IJ)=TNOW
0C17            IWAIT(IJ)=JTRIB(4)
0C18            IWDIE(IJ)=JTRIB(7)
0C19            RETURN
          C                                OTHERWISE, DELIVER, UPDATE, GATHER STAT.
0C20         12 CALL COLCT(0.,1,NSET,QSET)
0C21            TONOUT=JTRIB(4)*BAG(IJ)
0C22            CALL HISTO(0.,0.01,0.01,1)
0C23            CALL TMST(STOCK,TNOW,10)
0C24            STOCK=STOCK-TONOUT
0C25            KSTOCK(IJ)=KSTOCK(IJ)-JTRIB(4)
0C26            ASALE=ASALE+TONOUT
0C27            SALE(IJ)=SALE(IJ)+JTRIB(4)
          C                                REORDER POINT REACHED AFTER DELIVERY?
0C28            IF((KSTOCK(IJ)+KORDER(IJ)).GT.IREORD(IJ))RETURN
          C        IF SO, PLACE LOT SIZE, ASSIGN LOW PRIORITY
0C29            JTRIB(3)=LOT(IJ)
0C30            JTRIB(2)=1
0C31            JTRIB(4)=0
0C32            ATRIB(5)=JTRIB(3)*BAG(IJ)
0C33            ATRIB(6)=0.
          C                                ON ORDER VECTOR
0C34         11 KORDER(IJ)=KORDER(IJ)+JTRIB(3)
          C                                CHARGE ORDERING COST
0C35            ORDER=ORDER+COR
```

```
C                                   PUT IN PRODN BACKLOG, RANK ON PRIORITY,HVF
      CALL FILEM(3,NSET,QSET)
      ENTRY CHECK(NSET,QSET)
C                                   IS THE MIXER BUSY?
      IF(NQ(4).GT.0)RETURN
C                                   ANY ORDER IN PRODUCTION BACKLOG?
   36 IF(NQ(3).EQ.0)RETURN
      ENTRY MIXER(NSET,QSET)
C                                   PROCESS FIRST ORDER IN PRODN BACKLOG
   35 MFE3=MFE(3)
      CALL RMOVE(MFE3,3,NSET,QSET)
      IJ=JTRIB(5)
C                                   COMPUTE TIME ELAPSED SINCE ENTERING
      WT=TNOW-ATRIB(2)
C                                   IF WT< MAX LEAD TIME ALLOWABLE,GO ON
      IF(WT.LT.WTMAX(JTRIB(2)))GO TO 40
C                                   LOST SALE OR REPLENISHING ORDER CANCELED
      KORDER(IJ)=KORDER(IJ)-JTRIB(3)
      IF(JTRIB(4).EQ.0)GO TO 36
C                                   IF LOST SALE, CHARGE OUT OF STOCK COST
      NOUTS=NOUTS+1
      KOUTS(IJ)=KOUTS(IJ)+1
      COUT=COUT+COS(IJ)
      XLOST=XLOST+ATRIB(6)
C                                   ANY MORE ORDERS IN PRODN BACKLOG?
      GO TO 36
C                                   IS IT MASH FEED?
   40 IF(JTRIB(7).LT.1)GO TO 41
C                                   COMPUTE PELLETING CRITICAL QUANTITY
      IF(JTRIB(2).GT.1)GO TO 43
      TEST=ATRIB(5)
      GO TO 44
   43 TEST=ATRIB(6)
C                                   PM'S REQUIRED TO MEET DEMAND DURING LT
   44 NREQ=TEST/((WTMAX(JTRIB(2))-WT)*10.)+1.
C                                   PELLET MILLS AVALAIBLE
      NAVAL=0
      DO 45 I=1,NPEL
      IF(MILL(I).EQ.0)NAVAL=NAVAL+1
   45 CONTINUE
C                                   IS IT POSSIBLE TO MEET DEMAND DURING LT?
      IF(NAVAL.GE.NREQ)GO TO 46
      IF(JTRIB(4).EQ.0)GO TO 47
C                                   IF LOST SALE, CHARGE OUT OF STOCK COST
      NOUTS=NOUTS+1
      KOUTS(IJ)=KOUTS(IJ)+1
      COUT=COUT+COS(IJ)
      XLOST=XLOST+ATRIB(6)
C                                   CLEAR ON-ORDER VECTOR
   47 KORDER(IJ)=KORDER(IJ)-JTRIB(3)
      GO TO 36
C                                   IF POSSIBLE, MIX
   46 TSMIX=0.
      ANREQ=NREQ
      CALL HISTO(ANREQ,2.,1.,3)
C                                   IDENTIFY INGREDIENTS
      IMIX=JTRIB(1)/100
      NMIXER=NMIXER+1
```

```
C                                   IF NECESSARY, SET UP, CHARGE COST
      IF(MIXOLD.EQ.IMIX)GO TO 48
      SETUP1=SETUP1+CMIX
      TSMIX=TSMIX+0.10
C                                   GATHER STATISTICS
      SETMIX=SETMIX+TSMIX
C                                   STORE NAME OF INGREDIENTS
      MIXOLD=IMIX
C                                   MIXER BUSY (INCLUDIND SETUP TIME,IF ANY)
   48 CALL FILEM(4,NSET,QSET)
C                                   COMPUTE MIXING TIME
      TMIX=(ATRIB(5)-BEGIN)/RATIO
C                                   SCHEDULE END OF MIX
      TEMPX=TMIX+CYCLE
      CALL COLCT(TEMPX,5,NSET,QSET)
      JTRIB(1)=5
      ATRIB(1)=TNOW+TSMIX+TEMPX
      CALL FILEM(1,NSET,QSET)
C                                   COMPUTE PELLETING QUOTA
      AMOUNT=ATRIB(5)/ANREQ
      TEMPY=AMOUNT/10.
      TEMPMI=ATRIB(5)/RATIO
      SAVET=RATIO/ANREQ
      NREADY=0
      TOTDIE=0.
      KOUNT=0
C                                   SEIZE PELLET MILLS
      DO 49 I=1,NPEL
      TDIE=0.
      IF(MILL(I).GT.0)GO TO 49
C                                   IF NECESSARY, CHANGE DIE, CHARGE COST, GS
      IF(IDIE(I).EQ.JTRIB(7))GO TO 51
C                                   LOOK FOR MILL WITH MATCHING DIE SIZE
      IF((IREQ-NREADY).LT.(IAVAL-NREADY))GO TO 49
      SETUP2=SETUP2+CDIE
      TDIE=TDIE+0.40
      TOTDIE=TOTDIE+TDIE
      KOUNT=KOUNT+1
C                                   STORE CURRENT DIE SIZE
      IDIE(I)=JTRIB(7)
C                                   PUT PM NUMBER IN ORDER TAG
      IF(KOUNT.EQ.1)JTRIB(8)=I
C                                   GATHER STATISTICS
   51 NREADY=NREADY+1
      IF(NREADY.LT.IREQ)GO TO 1000
      IF(KOUNT.EQ.0)JTRIB(8)=I
 1000 TEMPM=MILL(I)
      CALL TMST(TEMPM,TNOW,I)
C                                   SET PELLET MILL TO BUSY
      MILL(I)=1
      NMILL=NMILL+1
      IF(ILAST(I).EQ.IMIX)GO TO 1001
      ILAST(I)=IMIX
      SETUP2=SETUP2+CMILL
C                                   SCHEDULE END OF JOB IN THIS PM
 1001 TEMPO=TSMIX+CONVP
      XTRIB=JTRIB(4)
      ATRIB(3)=XTRIB/ANREQ
```

```
0119          JTRIB(1)=2
0120          ATRIB(4)=AMOUNT/BAG(IJ)-ATRIB(3)
0121          JTRIB(6)=I
0122          IF(TDIE.GT.0.)TEMPO=TDIE
0123          CALL COLCT(TEMPY,6,NSET,QSET)
0124          CALL HISTO(TEMPY,1.,1.,4)
0125          ATRIB(1)=TNOW+TEMPO+TEMPY+COOL+CONVB
0126          CALL FILEM(1,NSET,QSET)
        C                                GATHER STATISTICS
0127          TONMIL(I)=TONMIL(I)+AMOUNT
0128          IF(NREADY.EQ.IREQ)GO TO 53
0129       49 CONTINUE
        C                                GATHER STATISTICS ON BIN SIZE
0130       50 IF(TOTDIE.EQ.0.)GO TO 52
0131          IF(TEMPMI.GT.0.40)GO TO 53
0132          BIN=AMOUNT
0133          GO TO 54
0134       53 BIN=SAVET*0.40+(SAVET-10.)*(TEMPMI-0.40)
0135          GO TO 54
0136       52 BIN=(SAVET-10.)*(TEMPMI-BASE/SAVET)+BASE
0137       54 CALL COLCT(BIN,2,NSET,QSET)
0138          CALL HISTO(BIN,10.,10.,2)
        C                                SCHEDULE GATHERING OF STAT ON MAX LT
0139          IF(JTRIB(4).EQ.0)RETURN
0140          JTRIB(1)=6
0141          ATRIB(1)=TNOW+TEMPO+ATRIB(6)/
             1 (AMOUNT*IREQ/(TEMPY+TOTDIE/IREQ))+COOL+CONVB
0142          CALL FILEM(1,NSET,QSET)
0143          RETURN
        C                                IF MASH, MIX
0144       41 TSMIX=0.
0145          NMIXER=NMIXER+1
0146          IMIX=JTRIB(1)/100
0147          IF(MIXOLD.EQ.IMIX)GO TO 55
0148          SETUP1=SETUP1+CMIX
0149          TSMIX=TSMIX+0.10
0150          SETMIX=SETMIX+TSMIX
0151          MIXOLD=IMIX
0152       55 CALL FILEM(4,NSET,QSET)
0153          TMIX=(ATRIB(5)-BEGIN)/RATIO
        C                                SCHEDULE END OF MIX
0154          TEMPX=TMIX+CYCLE
0155          CALL COLCT(TEMPX,5,NSET,QSET)
0156          JTRIB(1)=5
0157          ATRIB(1)=TNOW+TSMIX+TEMPX
0158          CALL FILEM(1,NSET,QSET)
        C                                SCHEDULE END OF JOB
0159          JTRIB(1)=2
0160          ATRIB(4)=JTRIB(3)-JTRIB(4)
0161          ATRIB(1)=ATRIB(1)+CONVB
0162          ATRIB(3)=JTRIB(4)
0163          CALL FILEM(1,NSET,QSET)
        C                                SCHEDULE GATHERING OF STAT, IF NECESSARY
0164          IF(JTRIB(4).EQ.0)RETURN
0165          JTRIB(1)=6
0166          ATRIB(1)=TNOW+TSMIX+ATRIB(6)/RATIO+CONVB
0167          CALL FILEM(1,NSET,QSET)
0168          RETURN
0169          END
```

```
0001          SUBROUTINE ENDMIX(NSET,QSET)
0002          DIMENSION NSET(1),QSET(1)
0003          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXOS,MAXNS,
             3ATRIB(10),ENQ(14),INQ(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0004          COMMON/LILITA/IRECORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),KAIT(100),IWAIT(100),IKDIE(100),ILAST(1 ),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
        C                                RELEASE MIXER
0005          MFE4=MFE(4)
0006          CALL RMOVE(MFE4,4,NSET,QSET)
        C                                IF ANY ORDER IN PRODN BACKLOG,PROCESS
0007          IF(NQ(3).GT.0)CALL MIXER(NSET,QSET)
0008          END
```

```
0C01          SUBROUTINE READY(NSET,QSET)
0C02          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0C03          COMMON/LILITA/IRFORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
        C                                COMPUTE TIME IN SYSTEM
0C04          ALEAD=TNOW-ATRIB(2)
        C                                GATHER STAT ON TIME IN SYSTEM
0C05          CALL COLCT(ALEAD,1,NSET,QSET)
0C06          CALL HISTO(ALEAD,0.01,0.01,1)
0C07          RETURN
0C08          END
```

```
0001          SUBROUTINE ENDJOB(NSET,QSET)
0002          DIMENSION NSET(1),QSET(1)
0003          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0004          COMMON/LILITA/IREORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TJNMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0005          IJ=JTRIB(5)
        C                                GATHER STAT ON ORDER COMPLETION TIME
0006          XLEAD=TNOW-ATRIB(2)
0007          CALL HISTO(XLEAD,2.,2.,5)
0008          CALL COLCT(XLEAD,3,NSET,QSET)
        C                                MASH? IF SO,JUMP
0009          IF(JTRIB(7).LT.1)GO TO 10
        C                                IF LAST PM,REDUCE ON-ORDER VECTOR BY LOT
0010          IF(JTRIB(8).EQ.JTRIB(6))KORDER(IJ)=KORDER(IJ)-JTRIB(3)
        C                                GATHER STAT ON PELLET MILL USAGE
0011          INDEX=JTRIB(6)
0012          TEMPM=MILL(JTRIB(6))
0013          CALL TMST(TEMPM,TNOW,INDEX)
0014          MILL(INDEX)=0
0015          GO TO 11
        C                                UPDATE STATISTICS
0016       10 KORDER(IJ)=KORDER(IJ)-JTRIB(3)
0017       11 CALL TMST(STOCK,TNOW,10)
0018          TEMPB=ATRIB(3)*BAG(IJ)
0019          STOCK=STOCK+ATRIB(4)*BAG(IJ)
0020          KSTOCK(IJ)=KSTOCK(IJ)+ATRIB(4)
0021          SUMPRO=SUMPRO+(ATRIB(4)+ATRIB(3))*BAG(IJ)
        C                                HIGH PRIORITY? IF NOT SO,JUMP
0022          IF(TEMPB.EQ.0.)GO TO 20
        C                                DELIVER FEED,UPDATE
0023          ASALE=ASALE+TEMPB
0024          SALE(IJ)=SALE(IJ)+ATRIB(3)
        C                                IF NECESSARY,ORDER REPLENISHING
0025          IF(KSTOCK(IJ).LE.IREORD(IJ))CALL REPLE(NSET,QSET)
0026          GO TO 888
        C                                IF NO CUSTOMER WAITS,JUMP
0027       20 IF(IWAIT(IJ).EQ.0)GO TO 888
        C                                COMPUTE WAITING TIME
0028          WTD=TNOW-WAIT(IJ)
        C                                IF STILL WITHIN CLASS LT,JUMP
0029          IF(WTD.LT.WTMAX(2))GO TO 30
        C                                CHARGE OUT OF STOCK COST,UPDATE
0030          NOUTS=NOUTS+1
0031          KOUTS(IJ)=KOUTS(IJ)+1
0032          COUT=COUT+COS(IJ)
0033          XLOST=XLOST+IWAIT(IJ)*BAG(IJ)
```

```
0C34          IWAIT(IJ)=0
0C35          GO TO 888
        C                                IF ENOUGH STOCK,JUMP
0C36       30 IF(KSTOCK(IJ).GE.IWAIT(IJ))GO TO 777
        C                                OTHERWISE,CONTINUE TO WAIT
0C37          GO TO 888
        C                                UPDATE STAT,DELIVER FEED
0C38      777 CALL TMST(STOCK,TNOW,1C)
0C39          STOCK=STOCK-IWAIT(IJ)*BAG(IJ)
0C40          KSTOCK(IJ)=KSTOCK(IJ)-IWAIT(IJ)
0C41          ASALE=ASALE+IWAIT(IJ)*BAG(IJ)
0C42          SALE(IJ)=SALE(IJ)+IWAIT(IJ)
0C43          IWAIT(IJ)=0
        C                                GATHER STAT ON COMPLETION TIME
0C44          ALEAD=TNOW-WAIT(IJ)
0C45          CALL COLCT(ALEAD,1,NSET,QSET)
0C46          CALL HISTO(ALEAD,0.01,0.01,1)
        C                                IF NECESSARY,ORDER REPLENISHING
0C47          IF(KSTOCK(IJ).LE.IREORD(IJ))CALL REPLE(NSET,QSET)
        C                                TRANSFER CONTROL TO MASTER
0C48      888 CALL CHECK(NSET,QSET)
        C                                GO TO NEXT EVENT
0C49          RETURN
0C50          END
```

```
0001          SUBROUTINE SAMPLE(NSET,QSET)
0002          DIMENSION NSET(1),QSET(1)
0003          COMMON ID,TI,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NUT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,MEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0004          COMMON/LILITA/IRECORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNDTE(10,80),MILL(10),IDTE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDTE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0005          IJ=JTRIB(5)
0006          JTRIB(1)=NTYPE(IJ)
0007          JTRIB(3)=LOT(IJ)
0008          JTRIB(2)=2
0009          JTRIB(4)=0
0010          JTRIB(6)=99
0011          ATRIB(5)=JTRIB(3)*BAG(IJ)
0012          ATRIB(6)=0.
0013          ATRIB(2)=TNOW
0014          CALL FILEM(3,NSET,QSET)
0015          KORDER(IJ)=KORDER(IJ)+JTRIB(3)
0016          ORDER=ORDER+COR
0017          RETURN
0018          END
```

```
0001          SUBROUTINE RECORD(NSET,QSET)
0002          DIMENSION NSET(1),QSET(1)
0003          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0004          COMMON/LILITA/IRECORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),CDS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,CDOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0005          ATRIB(1)=TNOW+16.00
0006          CALL FILEM(1,NSET,QSET)
        C                                END-OF-DAY STOCK
0007          A(1,IDATE)=STOCK
        C                                AVERAGE STOCK THROUGHOUT PERIOD
0008          A(2,IDATE)=SSUMA(10,2)/SSUMA(10,1)
        C                                AVERAGE PERCENTAGE OF STOCKOUTS
0009          XD=NOUTS
0010          XD=NDEMAN
0011          A(3,IDATE)=XO/XD*100.
        C                                AVERAGE MIXER UTILIZATION
0012          A(4,IDATE)= (ENQ(4)+NQ(4)*(TNOW-QTIME(4)))/(TNOW-TBEG)-
             1 SETMIX/(IDATE*16.)
        C                                DAILY PRODUCTION
0013          A(5,IDATE)=SUMPRO
        C                                DAILY SALES
0014          A(6,IDATE)=ASALE
0015          WRITE(NPRNT,10)IDATE,A(1,IDATE),A(2,IDATE),A(3,IDATE),A(4,IDATE),
             1 A(5,IDATE),A(6,IDATE)
0016       10 FORMAT(I10,F10.1,2F10.2,F10.3,2F10.1)
0017          TOTPRO=TOTPRO+SUMPRO
0018          TOTSAL=TOTSAL+ASALE
0019          RETURN
0020          END
```

```
0001          SUBROUTINE FIRST(NSET,QSET)
0002          DIMENSION NSET(1),QSET(1)
0003          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0004          COMMON/LILITA/IRECORD(100),WTMAX(5),ISAFE(100),IOBS(100),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,TFLAT,STD(100),IHOLD(100)
0005          SSUMA(10,1)=1.
0006          SSUMA(10,2)=STOCK
0007          WRITE(NPRNT,25)
0008       25 FORMAT('1',//,6X,'DATE',5X,'STOCK',2X,'STOCK(+)',7X,'OOS',5X,'MIXE
             1R',5X,'PRODN',5X,'SALES'/)
0009          RETURN
0010          END
```

```
0001          SUBROUTINE OTPUT(NSET,QSET)
0002          DIMENSION NSET(1),QSET(1)
0003          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NORDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0004          COMMON/LILITA/IRECORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0005          DIMENSION X(360),AVP(10),AVPR(10)
C                                   PLOT GRAPHS
0006          DO 15 I=1,KGRAPH
0007          DO 20 J=1,IDATE
0008       20 X(J)=A(I,J)
0009          CALL MAX(IDATE,X,J,XLARGE)
0010          IF(XLARGE.EQ.0.)GO TO 15
0011          WRITE(3,1)(JNOTE(I,J),J=1,80)
0012          CALL MIN(IDATE,X,J,SMALL)
0013          DOWN=XLARGE-SMALL
0014          IF(DOWN.EQ.0.)GO TO 15
0015          SCALE=50./DOWN
0016          CALL PLOT(X,SCALE,SMALL)
0017       15 CONTINUE
C                                   PRINT OUT INVENTORY COST REPORT
0018          CARRY=(A(2,IDATE)*CCARRY+SSUMA(10,5)*CSTORE)*IDATE*(1.+WORKNO)
0019          COST=ORDER+SETUP1+SETUP2+CARRY+COUT
0020          CSET1=SETUP1/NMIXER
0021          CSET2=SETUP2/NMILL
0022          XCOST=100./COST
0023          P1=SETUP1*XCOST
0024          P2=SETUP2*XCOST
0025          P3=ORDER*XCOST
0026          P4=COUT*XCOST
0027          P5=CARRY*XCOST
0028          WRITE(NPRNT,100)IDATE,SETUP1,P1,SETUP2,P2,ORDER,P3,COUT,P4,CARRY,
             1 P5,COST,CSET1,CSET2
C             PRINT PRODUCTION-INVENTORY REPORT
0029          AVSALE=TOTSAL/IDATE
0030          RATE=TOTPRO/(SETMIX+A(4,IDATE)*IDATE*16.)
0031          AVLEAD=SUMA(1,1)/SUMA(1,3)
0032          AVTIME=(1.-A(4,IDATE))*16.-SETMIX/IDATE
0033          AVLOST=XLOST/IDATE
0034          AVPRO=TOTPRO/IDATE
0035          AVOUT=A(3,IDATE)
0036          A(4,IDATE)=A(4,IDATE)*100.
0037          WRITE(NPRNT,101)IDATE,TOTSAL,AVSALE,A(2,IDATE),SSUMA(10,5),
             1 SSUMA(10,4),A(4,IDATE),RATE,AVLEAD,AVTIME,AVOUT,AVLOST,AVPRO
C                                   PRINT PELLETING REPORT
0038          TOTON=0.
0039          DO 499 I=1,NPEL
```

```
0040          499 AVPR(I)=0.
0041              TOTIME=0.
0042              DO 500 I=1,NPEL
0043              AVP(I)=TONMIL(I)/IDATE
0044              IF(SSUMA(I,1).EQ.0.)GO TO 500
0045              TEMPT=SSUMA(I,2)/SSUMA(I,1)*IDATE*16.
0046              AVPR(I)=TONMIL(I)/TEMPT
0047              TOTON=TOTON+TONMIL(I)
0048              TOTIME =TOTIME+TEMPT
0049          500 CONTINUE
0050              AVTOT=TOTON/TOTIME
0051              WRITE(3,502)(AVP(I),I=1,NPEL),(AVPR(I),I=1,NPEL),AVTOT
0052            1 FORMAT('1',80A1,///,7X,'DAY',5X,'VALUE'//)
0053          100 FORMAT('1',//////////,33X,'***** INVENTORY COST REPORT *****'////,
                 1 41X,I3,'-DAY SIMULATION',///,
                 2 30X,'MIXER SETUP COST',4X,'$',F10.2,F10.1,' %',//,
                 3 30X,'P M SETUP COST',6X,'$',F10.2,F10.1,' %',//,
                 4 30X,'ORDERING COST',7X,'$',F10.2,F10.1,' %',//,
                 5 30X,'OUT-OF-STOCK COST',3X,'$',F10.2,F10.1,' %',//,
                 6 30X,'CARRYING COST',7X,'$',F10.2,F10.1,' %',//,
                 7 37X,'TOTAL COST',3X,'$',F10.2,/////,
                 830X,'AVERAGE MIXER SETUP COST',7X,'$',F5.1,//,
                 930X,'AVERAGE PELLETING SETUP COST',3X,'$',F5.1)
0054          502 FORMAT(/////,51X,'PELLET MILL EFFICIENCY REPORT'    ///,27X,
                 1'#1',8X,'#2',8X,'#3',8X,'#4',8X,'#5',8X,'#6',8X,'#7',8X,'#8',8X,
                 2'#9'//,51X,'AVERAGE PRODUCTION(TONS/DAY)'//,20X,9F10.2,//,51X,
                 3'AVERAGE PRODUCTION RATE(T/H)'//,20X,9F10.2,//,51X,'OVERALL AVERAG
                 4E PRODUCTION RATE = ',F4.2,' TONS/HR')
0055          101 FORMAT('1',////,29X,'***** PRODUCTION-INVENTORY PARAMETER REPORT *
                 1****'////,40X,I3,'-DAY SIMULATION',///,30X,'TOTAL SALES',19X,F10.3
                 2,' TONS',//,30X,'AVERAGE SALES PER DAY',9X,F10.3,' TONS/DAY',//,
                 3 30X,'AVERAGE TOTAL STOCK',11X,F10.3,' TONS',//,30X,'MAXIMUN STOCK
                 4 ON HAND',9X,F10.3,' TONS',//,30X,'MINIMUN STOCK ON HAND',9X,
                 5F10.3,' TONS',//,30X,'AVERAGE MIXER UTILIZATION'  ,5X,F10.2,' %',
                 6//,30X,'AVERAGE PRODUCTION RATE',7X,F10.1,' TONS/HOUR',//,
                 730X,'AVERAGE LEAD TIME',13X,F10.3,' HOURS',//,30X,'AVERAGE TIME LO
                 8ST',13X,F10.2,' HOURS/DAY',//,30X,'AVE. PERCENTAGE OF STOCKOUTS',
                 9 2X,F10.2,' %',//,30X,'AVERAGE LOST SALES',13X,F10.1,' TONS/DAY',
                 1//,30X,'AVERAGE DAILY PRODUCTION',7X,F10.1,' TONS')
0056              RETURN
0057              END
```

```
0001          SUBROUTINE RESET
0002          COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
             1NDQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
             2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
             3ATRIB(10),ENQ(14),INN(14),JCELS(5,22),KRANK(14),MAXNQ(14),MFE(14),
             4MLC( 14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
             50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14)
0003          SSUMA(10,5)=0.
0004          RETURN
0005          END
```

```
0C01          SUBROUTINE PLOT(X,SCALE,AMINI)
0C02          COMMON/LILITA/IRECORD(1C0),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(10C),ACRD(2C),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0C03          DIMENSION X(360),BLANK(100)
0C04          DATA STAR/'*'/,BLANK/1C0*' '/
0C05          DO 2C I=1,IDATE
0C06          NBLANK=(X(I)-AMINI)*SCALE
0C07          IF(NBLANK.EQ.0)GO TO 21
0C08          WRITE(NPRNT,90)I,X(I),(BLANK(J),J=1,NBLANK),STAR
0C09          GO TO 20
CC10       21 WRITE(NPRNT,91)I,X(I),STAR
0011       20 CONTINUE
0C12       90 FORMAT(I10,F10.2,20X,8CA1,A1)
0C13       91 FORMAT(I10,F10.2,21X,A1)
0C14          RETURN
0C15          END
```

```
0001          SUBROUTINE MIN(N,X,J,SMALL)
0002          COMMON/LILITA/IREORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JNOTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0003          DIMENSION X(360)
0004          SMALL=X(1)
0005          J=1
0006          IF(N.EQ.1)GO TO 7
0007          DO 6 I=2,N
0008          IF(X(I).GE.SMALL)GO TO 6
0009          SMALL=X(I)
0010          J=I
0011        6 CONTINUE
0012        7 RETURN
0013          END
```

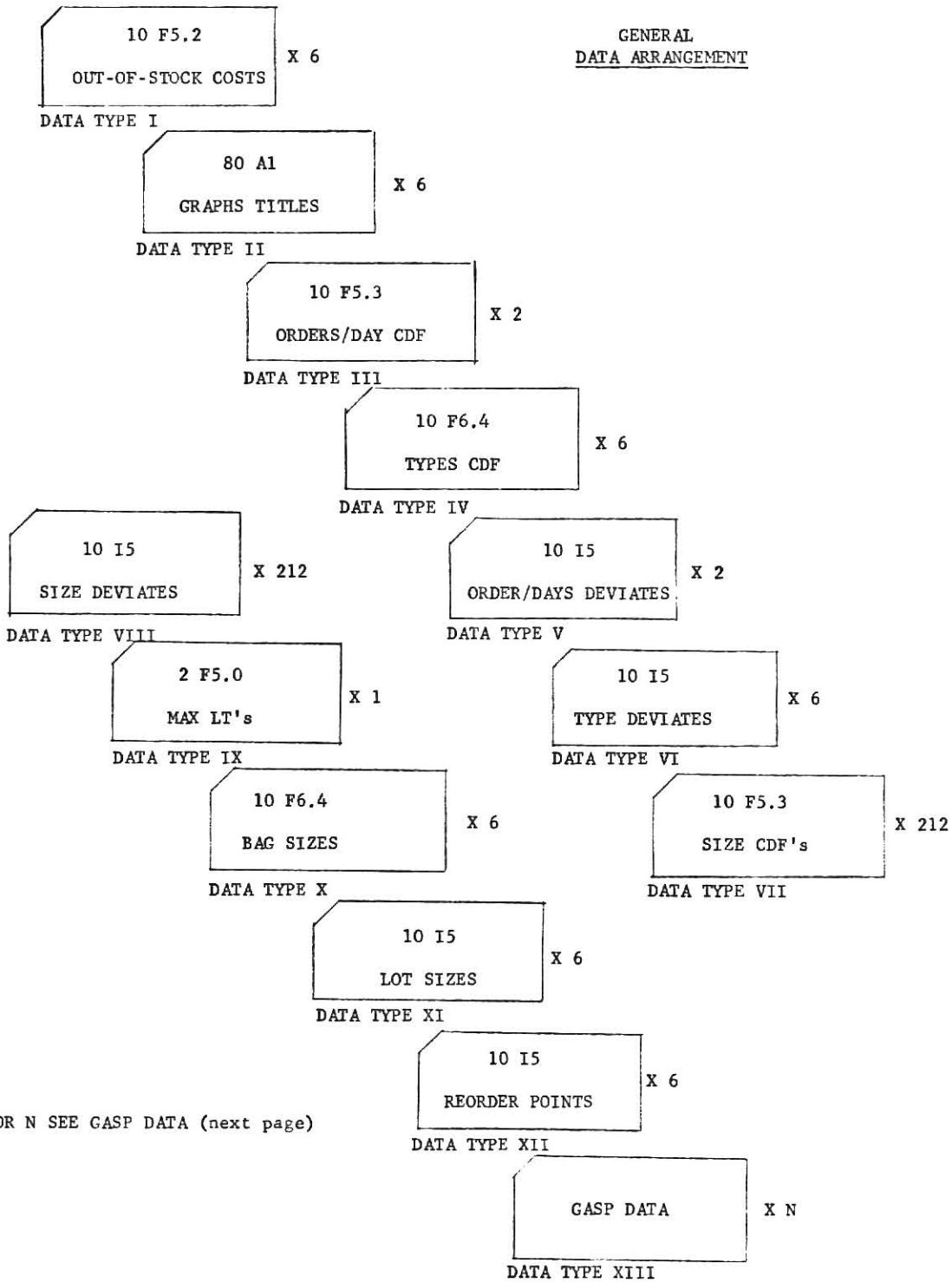
```
0C01          SUBROUTINE MAX(N,X,J,XLARGE)
0C02          COMMON/LILITA/IREORD(100),WTMAX(5),A(10,360),
             1 SALE(100),BAG(100),IDEMAN(100),KSTOCK(100),KOUTS(100),AORD(20),
             2 NORD(20),ATYPE(100),NTYPE(100),ASIZE(100,40),NSIZE(100,40),
             3 IFEED(100),LOT(100),JADTE(10,80),MILL(10),IDIE(10),TONMIL(10),
             4 KORDER(100),COS(100),WAIT(100),IWAIT(100),IWDIE(100),ILAST(10),
             5 COR,STOCK,CMIX,CDIE,CCARRY,CSTORE,ORDER,CMILL,COUT,RATIO,IDATE,
             6 SUMPRO,ASALE,NFEEDS,NPEL,XLOST,BASE,SETMIX,NOUTS,NDEMAN,KGRAPH,
             7 MIXOLD,COOL,CONVP,CONVB,BEGIN,CYCLE,TOTPRO,TOTSAL,SETUP1,SETUP2,
             8 NMILL,NMIXER,WORKNO
0C03          DIMENSION X(360)
0C04          XLARGE=X(1)
0C05          J=1
0C06          IF(N.EQ.1)GO TO 7
0C07          DO 6 I=2,N
0C08          IF(X(I).LE.XLARGE)GO TO 6
0C09          XLARGE=X(I)
0C10          J=I
0C11        6 CONTINUE
0C12        7 RETURN
0C13          END
```

```
$JOB            OJH,RUN=CHECK,TIME=3,LINES=66,PAGES=10
      DIMENSION A(1000),STOCK(119),PRODN(119)
 1000 READ(1,1,END=500)STOCK
      READ(1,1)PRODN
    1 FORMAT(20F4.0)
      DO 10 I=1,119
      A(I*2-1)=STOCK(I)
   10 A(I*2)=PRODN(I)
      CALL GRAPH(A)
      GO TO 1000
  500 STOP
      END
```

```
      SUBROUTINE GRAPH(DATA)
      DIMENSION DATA(1000),FORMAT(20),X(10),ALINE(105),T(6),TITLE(320)
      DATA LINE/'|'/,ALINE/1*'+',1*'-',1*'|',19*'-',1*'|',19*'-',1*'|',
     1 19*'-',1*'|',19*'-',1*'|',19*'-',1*'|',1*'-',1*'+'/
      DATA BLANK/' '/
      COMMON A(10),B(100),ANEW(9),OLD(9),JK
      READ(1,1)NVAR,NOBS,LIMIT
      READ(1,2)(A(I),I=1,NVAR)
      READ(1,400)TITLE
      NK=NVAR*NOBS
      IF(LIMIT.GT.0)GO TO 77
      CALL MAX(NK,DATA,J,BIG)
      CALL MIN(NK,DATA,J,SMALL)
      GO TO 78
   77 READ(1,401)BIG,SMALL
   78 RANGE=BIG-SMALL
      UNIT=RANGE/5.
      DO 8 J=1,6
      XJ=J-1
    8 T(J)=SMALL+XJ*UNIT
      WRITE(3,339)TITLE,T
      WRITE(3,133)ALINE
      DO 20 J=1,NOBS
      JK=0
      DO 88 I=1,9
      ANEW(I)=BLANK
   88 OLD(I)=BLANK
      DO 10 I=1,100
   10 B(I)=BLANK
      IJ=(J-1)*NVAR+1
      IK=IJ+NVAR-1
      K=0
      DO 15 I=IJ,IK
      K=K+1
      XK=DATA(I)
   15 CALL SET(XK,K,RANGE,SMALL)
      WRITE(3,16)J,LINE,B,LINE,ANEW,OLD
   20 CONTINUE
      WRITE(3,133)ALINE
      RETURN
    1 FORMAT(2I5,F1)
    2 FORMAT(10A1)
   16 FORMAT(I6,1X,A1,1X,100A1,2X,A1,1X,9A1,1X,9A1)
  133 FORMAT(7X,105A1)
  401 FORMAT(2F10.4)
  339 FORMAT('1',3(80A1,/),//,80A1,/,3X,F10.3,5(10X,F10.3))
  400 FORMAT(80A1)
      END
```

```
      SUBROUTINE SET(X,I,R,S)
      COMMON A(10),B(100),ANEW(9),OLD(9),JK
      DATA BLANK/' '/
      J=100.*(X-S)/R+0.5
      IF(J.EQ.0)J=1
      IF(B(J).EQ.BLANK)GO TO 10
      JK=JK+1
      ANEW(JK)=A(I)
      OLD(JK)=B(J)
   10 B(J)=A(I)
      RETURN
      END
```

GENERAL
DATA ARRANGEMENT




FOR N SEE GASP DATA (next page)

THIS BOOK WAS BOUND WITH TWO PAGES NUMBERED 94. THESE PAGES ARE DIFFERENT.

THIS IS AS RECEIVED FROM CUSTOMER.

```
ECHO CHECK ON INPUT DATA

   NAME        NPROJ  MON NDAY  NYR  NRUNS
OJ HERNANDEZ    1     11   15  1970    1

   NPRMS NHIST NCLCT NSTAT  ID     IM   NOQ   MXC   IMM
     0     5    10    10   500     8    4    22    6

NCELS
   20   20   20   20   20

KRANK
    1  104  102    1

INN
    1    1    2    1

   MSTOP JCLR  NORPT  NEP     TBEG      TFIN    NSEED
     1    1     0     1     0.0    1903.999   3

SEEDS FOR RANDOM NUMBER GENERATION
  75453  68797  88985

FILE NO., ATTRIBUTES
        -1         0         0         0         0         0         0         0
         0
         1         4         0         0         0         0         0         0
         0
    0.0       0.0       0.0       0.0       0.0       0.0
         1         1         0         0         0         0         0         0
         0
    0.0       0.0       0.0       0.0       0.0       0.0
         1         3         0         0         0         0         0         0
         0
   15.9990    0.0       0.0       0.0       0.0       0.0
         1         7         0         0         0         0         0         0
         0
  320.0000    0.0       0.0       0.0       0.0       0.0
```

# APPENDIX C

ORDERS/DAY EMPIRICAL DISTRIBUTION

(All Feeds)

Class width = 20 orders/day

| Class Limit orders/day | Frequency | Probability | Cumulative Probability |
|---|---|---|---|
| 20 | 1 | 0.048 | 0.048 |
| 160 | 1 | 0.048 | 0.095 |
| 360 | 1 | 0.048 | 0.143 |
| 380 | 1 | 0.048 | 0.190 |
| 400 | 1 | 0.048 | 0.238 |
| 420 | 1 | 0.048 | 0.286 |
| 440 | 2 | 0.095 | 0.381 |
| 460 | 3 | 0.143 | 0.524 |
| 480 | 1 | 0.048 | 0.571 |
| 500 | 3 | 0.143 | 0.714 |
| 520 | 1 | 0.048 | 0.762 |
| 540 | 2 | 0.095 | 0.857 |
| 580 | 1 | 0.048 | 0.905 |
| 640 | 1 | 0.048 | 0.952 |
| 740 | 1 | 0.048 | 1.000 |

RELATIVE FREQUENCY OF FEED ORDERING

(Types Empirical Distribution)

Base = 1

| Formula | Probability | Cumulative | Formula | Probability | Cumulative |
|---|---|---|---|---|---|
| F-1 | 0.0092 | 0.0092 | F-27 | 0.0444 | 0.5090 |
| F-2 | 0.0135 | 0.0226 | F-28 | 0.0492 | 0.5582 |
| F-3 | 0.0060 | 0.0286 | F-29 | 0.0007 | 0.5589 |
| F-4 | 0.0020 | 0.0307 | F-30 | 0.0025 | 0.5615 |
| F-5 | 0.0010 | 0.0317 | F-31 | 0.0594 | 0.6209 |
| F-6 | 0.0350 | 0.0667 | F-32 | 0.0331 | 0.6540 |
| F-7 | 0.0060 | 0.0727 | F-33 | 0.0293 | 0.6833 |
| F-8 | 0.0059 | 0.0786 | F-34 | 0.0186 | 0.7018 |
| F-9 | 0.0081 | 0.0866 | F-35 | 0.0001 | 0.7019 |
| F-10 | 0.0181 | 0.1048 | F-36 | 0.0102 | 0.7121 |
| F-11 | 0.0384 | 0.1432 | F-37 | 0.0009 | 0.7130 |
| F-12 | 0.0374 | 0.1806 | F-38 | 0.0149 | 0.7279 |
| F-13 | 0.0035 | 0.1841 | F-39 | 0.0073 | 0.7353 |
| F-14 | 0.0402 | 0.2243 | F-40 | 0.0080 | 0.7432 |
| F-15 | 0.0508 | 0.2750 | F-41 | 0.0275 | 0.7707 |
| F-16 | 0.0084 | 0.2834 | F-42 | 0.0011 | 0.7719 |
| F-17 | 0.0195 | 0.3029 | F-43 | 0.0170 | 0.7889 |
| F-18 | 0.0088 | 0.3116 | F-44 | 0.0331 | 0.8220 |
| F-19 | 0.0016 | 0.3133 | F-45 | 0.0198 | 0.8418 |
| F-20 | 0.0009 | 0.3142 | F-46 | 0.0232 | 0.8650 |
| F-21 | 0.0064 | 0.3206 | F-47 | 0.0287 | 0.8938 |
| F-22 | 0.0660 | 0.3865 | F-48 | 0.0544 | 0.9482 |
| F-23 | 0.0124 | 0.3990 | F-49 | 0.0225 | 0.9707 |
| F-24 | 0.0615 | 0.4604 | F-50 | 0.0015 | 0.9723 |
| F-25 | 0.0021 | 0.4626 | F-51 | 0.0268 | 0.9991 |
| F-26 | 0.0019 | 0.4645 | F-52 | 0.0009 | 1.0000 |

ORDER SIZE EMPIRICAL DISTRIBUTION (SAMPLE)

Formula F-21 (Simulation code 30520)

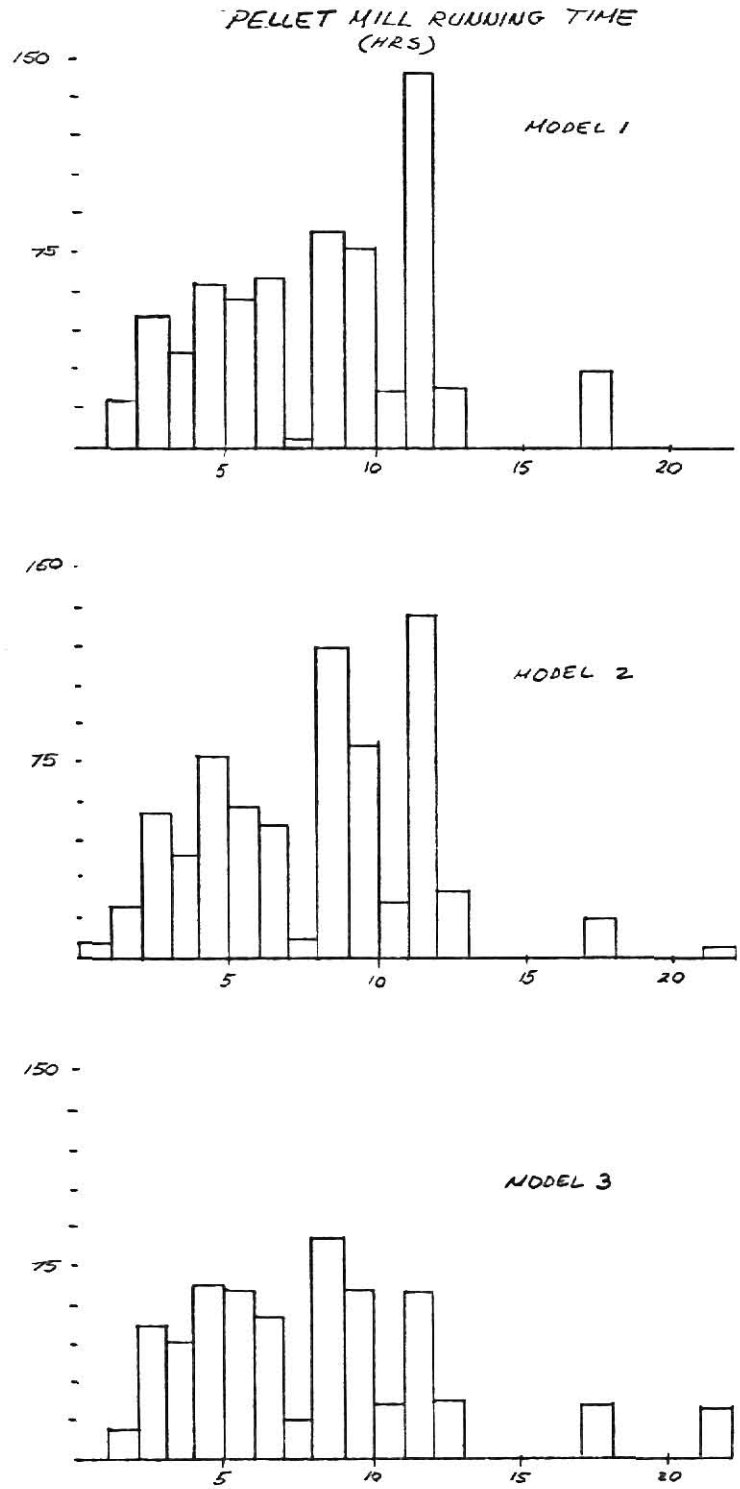
Mean Estimate = 124.06 bags/order  Std. Deviation = 115.07 bags/order

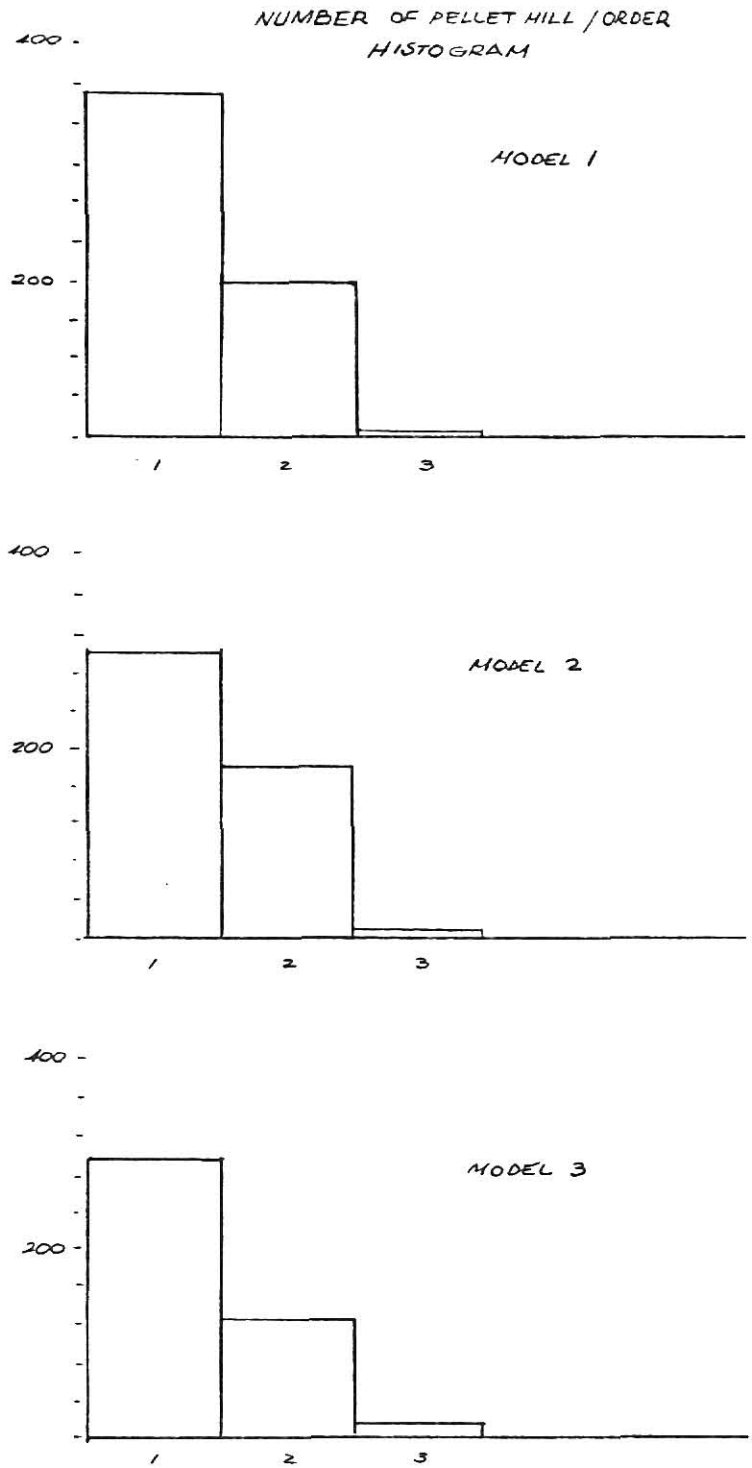
Sample size = 563 orders

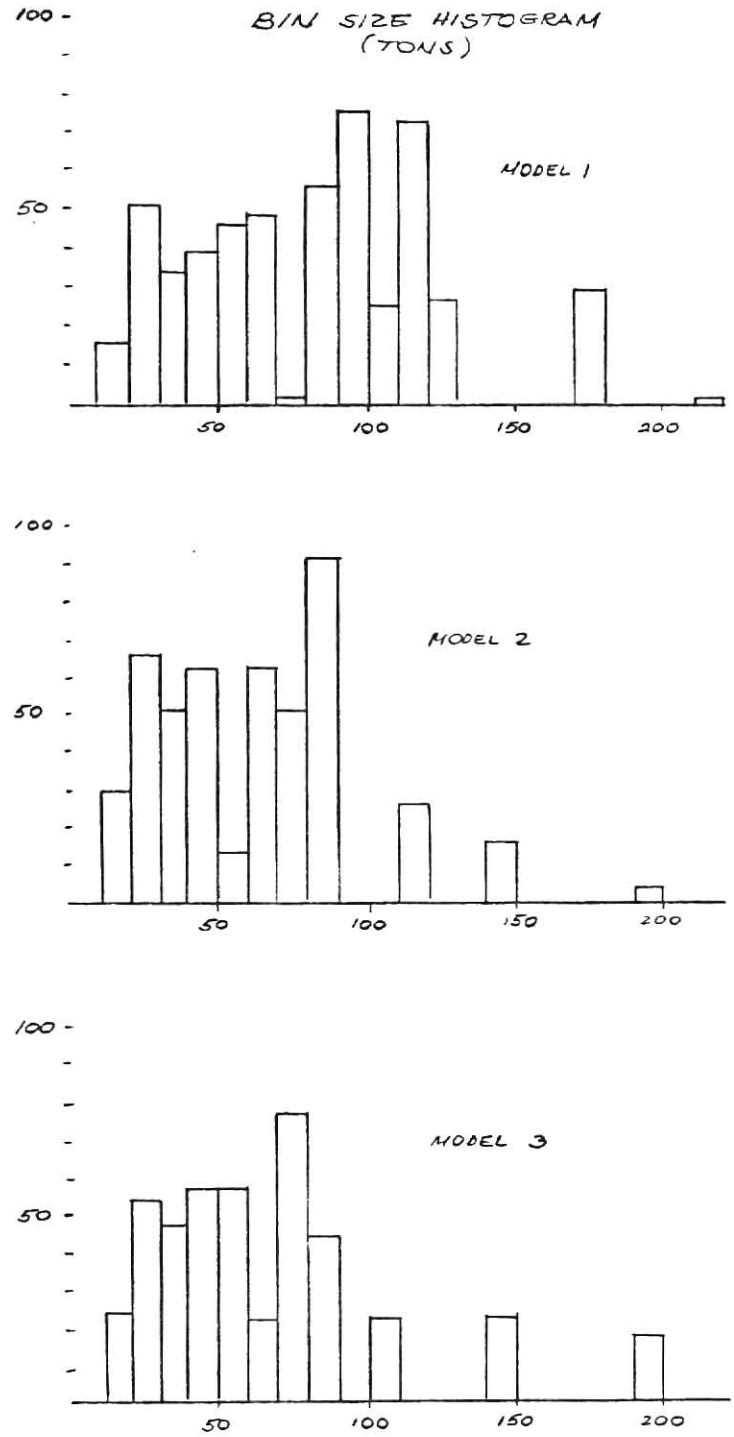
Class width = 10 bags/order

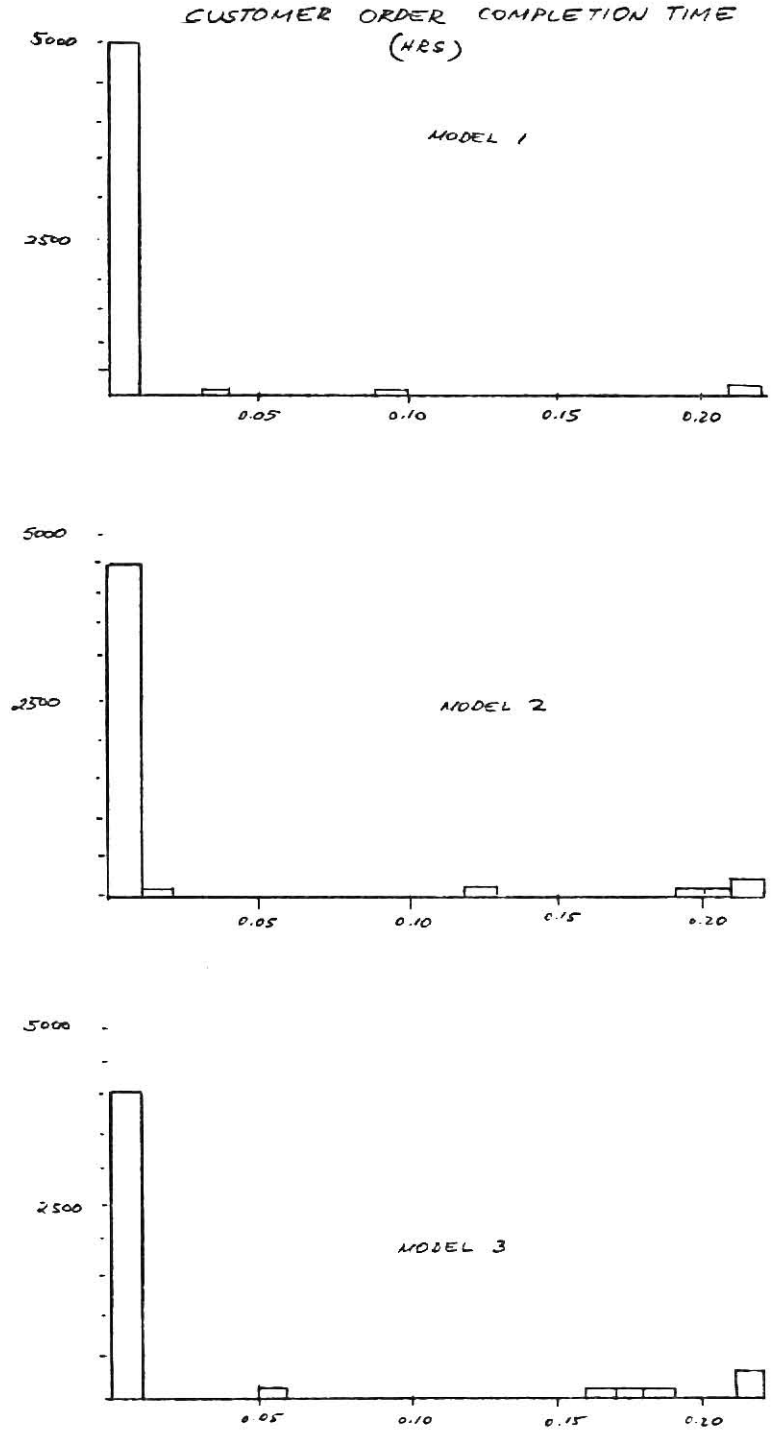
| Class Limit (bags/order) | Frequency | Probability | Cumulative |
|---|---|---|---|
| 10 | 78 | 0.138 | 0.139 |
| 20 | 32 | 0.057 | 0.195 |
| 30 | 32 | 0.057 | 0.252 |
| 40 | 20 | 0.036 | 0.288 |
| 50 | 47 | 0.084 | 0.371 |
| 60 | 18 | 0.032 | 0.403 |
| 70 | 15 | 0.027 | 0.430 |
| 80 | 17 | 0.030 | 0.460 |
| 90 | 9 | 0.016 | 0.476 |
| 100 | 49 | 0.087 | 0.563 |
| 110 | 6 | 0.011 | 0.574 |
| 120 | 22 | 0.039 | 0.613 |
| 130 | 10 | 0.018 | 0.631 |
| 140 | 9 | 0.016 | 0.647 |
| 150 | 38 | 0.068 | 0.714 |
| 160 | 8 | 0.014 | 0.728 |
| 170 | 5 | 0.009 | 0.737 |
| 180 | 7 | 0.012 | 0.750 |
| 190 | 3 | 0.005 | 0.755 |
| 200 | 44 | 0.078 | 0.833 |
| 210 | 2 | 0.004 | 0.837 |
| 220 | 1 | 0.002 | 0.838 |
| 240 | 3 | 0.005 | 0.844 |
| 260 | 18 | 0.032 | 0.876 |
| 280 | 2 | 0.004 | 0.879 |
| 290 | 1 | 0.002 | 0.881 |
| 300 | 26 | 0.046 | 0.927 |
| 320 | 1 | 0.002 | 0.929 |
| 330 | 3 | 0.005 | 0.934 |
| 340 | 2 | 0.004 | 0.938 |
| 350 | 5 | 0.009 | 0.947 |
| 360 | 2 | 0.004 | 0.950 |
| 370 | 1 | 0.002 | 0.952 |
| 380 | 1 | 0.002 | 0.954 |
| 400 | 17 | 0.030 | 0.984 |
| 450 | 1 | 0.002 | 0.986 |
| 460 | 1 | 0.002 | 0.988 |
| 500 | 7 | 0.012 | 1.000 |

# APPENDIX D

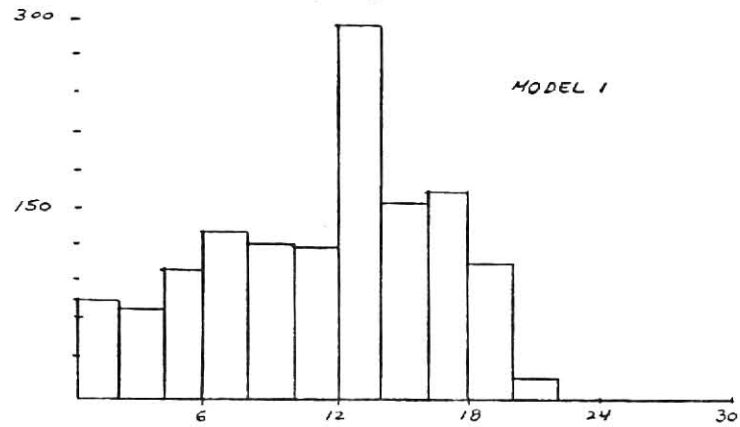

PELLET MILL RUNNING TIME
(HRS)
150
75
MODEL 1
5
10
15
20
150
75
MODEL 2
5
10
15
20
150
75
MODEL 3
5
10
15
20

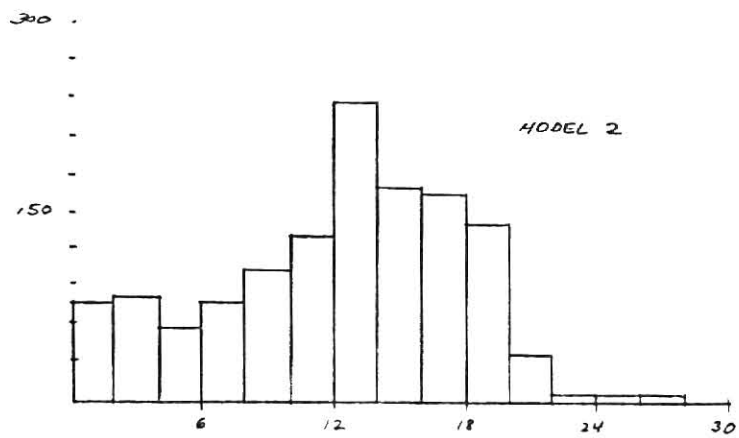

NUMBER OF PELLET HILL / ORDER
HISTOGRAM
400
200
MODEL 1
1
2
3
400
200
MODEL 2
1
2
3
400
200
MODEL 3
1
2
3

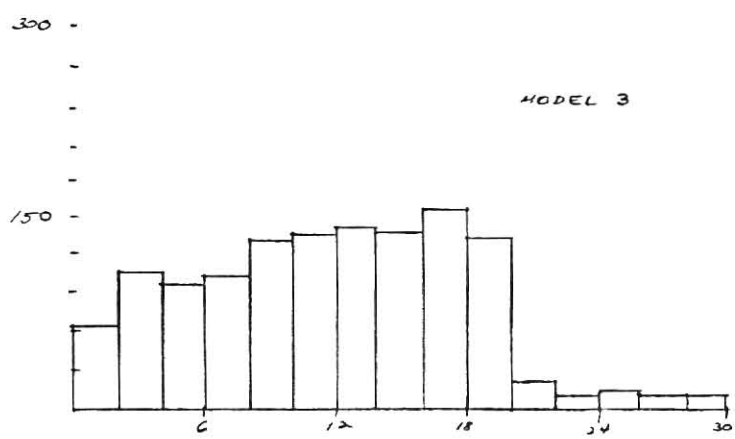

BIN SIZE HISTOGRAM
(TONS)
MODEL 1
MODEL 2
MODEL 3
100
50
50
100
150
200


CUSTOMER ORDER COMPLETION TIME
(HRS)
5000
2500
MODEL 1
0.05
0.10
0.15
0.20
5000
2500
MODEL 2
0.05
0.10
0.15
0.20
5000
2500
MODEL 3
0.05
0.10
0.15
0.20

REPLENISHING ORDER COMPLETION TIME (HRS)



# DESIGN OF FEED MILL FACILITIES BY COMPUTER SIMULATION

by

OMAR J. HERNANDEZ

Ingeniero Industrial, Universidad de Carabobo
Venezuela, 1966

---

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Grain Science and Industry

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971

A simulation model is a vehicle from which data for further analysis are collected when it is too unsafe, costly or uncertain to attempt a policy change in a business situation or to make a decision on an industrial project to be developed. In many cases, the actual simulation model is used in an interactive fashion with classical Operations Research techniques which can dictate the various policies to simulate.

This thesis has as a main objective the construction of a basic model that simulates the operation of a plant for the production of feedstuffs in bags. A search for an adequate combination of production facilities, inventory system, and finished products warehouse is initiated; provided that management standards regulate the service quality and a specific production acheduling rule is enforced. The problem is formulated in terms of a probabilistic distribution of demand, and a Q-model for dynamic inventory problems under uncertainty is employed for deciding on production lots and safety stocks.

Simulation techniques make possible the evaluation of usage rates of vital pieces of equipment (mixer, pellet mills) in a feed mill model. Also, information leading to the design of complementary equipment such as pelleting bins, or facilities like the finished feeds warehouse, can be obtained as a result of the simulation process.

Three models were implemented changing plant capacities according to an analysis of the measures of performance after every run. This approach can be applied in the design stage of a plant conceived for a restricted planning horizon or in making long-term or mid-range facility planning studies for a growing market situation.

GASP-IIA, a Fortran based set of subroutines organized for assisting in simulation studies, was employed. Running the model in Fortran IV level H to simulate half a year of real operations took approximately 10 minutes on the IBM 360/50 computer of the Kansas State University Computing Center.