A GASP SIMULATION STUDY OF A MULTI-PRODUCT INVENTORY SYSTEM

by 680

SAMBAMURTY ARISETTY

B. E. (Mechanical), Andhra University, India, 1966

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1969

Approved by

S. E. Grosh

Major Professor

TABLE OF CONTENTS

1. INTRODUCTION

This study is concerned with the use of GASP simulation language to simulate a multi-product inventory system, and to compare two alternative inventory management systems, the (s,S) system and the one based on adaptive forecasting using exponential smoothing. The objectives of this study are to: (1) simulate a multi-product inventory system, the products having different demand rates, (2) determine the most effective parameter values for use in exponential smoothing formula, (3) make available a version of GASP II usable on the IBM 360, the original version being written for GE 225 and (4) quantify in economic terms the expected savings derived from the application of adaptive forecasting technique to the inventory system. This inventory problem is taken from Starr and Miller (20).

It is assumed that customers follow an exponential arrival pattern with a normally distributed demand quantity. A delay has been incorporated between the warehouse and the posting department. The posting delay is assumed to follow a normal distribution. The lead time, the period from the time the order point is reached until the ordered material is available in inventory, is also assumed to follow a normal distribution. The process generating demands does not change with time. In particular, this implies that the mean rate of demand remains constant over time. In particular, this implies that the mean rate of demand remains constant over time. The inventory considered is a typical case of a stock of products replenished by purchases. The products are assumed to possess different usage rates and that the demand for each of these items is independent of the demand for other items. Customer demand continuously depletes the available stock of each item, and at the time that the inventory reaches a specified reorder level, an order is placed for the item.

GASP II was evolved at the Arizona State University by Pritsker and Kiviat (19). The original language and program were developed at the Applied Research Laboratory of the United States Steel Corporation by Kiviat (12). GASP is a computer program expressly designed for use in simulation studies of industrial systems to meet the growing demands for an efficient and easily understandable simulation language. The primary objectives were to promote the correct use of the simulation methodology by providing a workable tool to the prospective users and to reduce the long time usually required for the system studies by improving the communications between the engineer and the computer programmer. The language was based on a flowcharting description that used a small set of special symbols and conventions which were later related to particular GASP language statements. The applications of GASP to simulation projects include areas in steel manufacturing and transportation. GASP was written before the advent of SIMSCRIPT and probably owes much of its existence to this fact (12).

Packer (18) used IBM's IMPACT system (Inventory Management Planning and Control Technique) to simulate an inventory system to estimate cost savings resulting from adaptive forecasting as compared to optimal lot-size or economic order quantity. Batra (1) simulated a single-item inventory model using FORTRAN to investigate the effects of changes in the variances of demand and order lead time distribution on the actual demand during review period distribution and demand during lead time distribution.

2. INVENTORY MANAGEMENT

Inventory theory is defined as finding input (replenishment) and output (demand) functions for an inventory (defined as an ideal resource of any kind) that maximize a given measure of effectiveness subject to certain restrictions (20).

The inventory control system is a day-to-day operating system. The inventory control system maintains a record of stock status on hand and on order. It processes transactions about receipts, disbursements and adjustments. It can check the stock status against one or more control numbers. If the comparison passes a logical test, the inventory control system will generate a replenishment order. The inventory control system can be analyzed as a set of formal procedures. The implementation of these procedures would involve a computer, or manually posted records, or even the physical stock itself. The role of an order point in the inventory control system is to trigger the release of another requisition for more stock. A customer demand is subtracted from the quantity on hand for the item. If the quantity on hand is less than the demand, the latter is a stockout and a stockout is recorded. But such a demand is added to the accumulative demand so that we have a record from which to forecast what the future demand would be. Whenever an order is placed to replenish the stock, the quantity is added to the system stock (stock recorded in book and the stock on order). When the resulting stock is received at the warehouse, the quantity received is added to the physical balance on hand. The quantity received would be the same as ordered.

The inventory management system exists to set numerical values on the control numbers required in the inventory control system to decide when and how much to order. The economic order quantity is an example of such a control number. The inventory management system is concerned with the procedure that produces numerical values from observable characteristics of the item and current management policy. The inputs to the inventory management system include: 1) the forecast of demand and the mean absolute deviation of forecast errors; 2) item characteristics such as cost data, bulk, lead time, competetive classification or essentiality, and so on, and 3) management policy expressed as numerical values for the policy variables, such as carrying charge, the cost to expedite, or the desired item service. The alternative inventory management systems center on the formulas that express the output control numbers as functions of all of the input data.

Alternative Inventory Management Systems

The inventory control systems, at least for single items, need numerical values for two control parameters. These may be: 1) the order point and the order quantity; 2) the order point and the operating level; 3) the interval between orders and the operating level, or 4) the interval between orders and the order quantity.

The system will operate with any numerical values at all. If the order points are too small, there will be excessive shortages. If the order points or the order quantities are too large, then excessive stocks will be built up. The system will operate, but the consequences may not be pleasing to the management. The aim of successful inventory management is to compute numerical values, which when used in the control system would produce optimal results. The relevant results can be expressed as:

1. capital investment in the inventory;
2. customer service whether measured by frequency or by seriousness of shortages and
3. the operating costs of running the system.

The (s,S) System

When the available stock at any opportunity to order can be actually less than the order point, it is more economical to order up to an operating level than to order a fixed-order quantity. As the stock jumps, it is possible for it to be above reorder point on one review so that no order is triggered. At the next review the stock will be appreciably below the order point. In that case, the (s,S) system will order a larger quantity than an order-point, order quantity system would (3). The (s,S) policy requires that two appropriate numbers, s and S be specified, with $0 \leq s \leq S$. The rule is that when the available stock falls to, or below, the order point s, order the difference between the operating level S and the available stock. The order point is the sum of the safety stock and the expected demand during the lead time. The lead time includes not only the replenishment lead time, but also the review period between successive opportunities to order. An order quantity is placed equal to:

$$Q = S - (\text{Stock on hand} + \text{stock on order})$$

Here S is the maximum quantity that has been decided upon by management. The approximate formula used to find the maximum system stock for each item, when the lead time is a random variable, in terms of the average annual variable cost is given by (11):

$$F\left[\frac{S-\bar{x}}{\sigma_{\bar{x}}}\right] = \frac{CC_cT}{C_s + CC_cT}$$

$$F(y) = \text{Cumulative normal} = \int_{-\infty}^{y} f(t)\,dt$$

where

S = Inventory operating level, number of units

$\bar{x}$ = Expected demand during lead time

$\sigma_{\bar{x}}$ = Standard deviation of demand during lead time

C = Cost of item, dollars per unit

C_c = Carrying rate, percentage per year

T = Review period, fraction of year

C_s = Cost of a stockout, dollars per unit

Using normal tables, all the other values known the value of S can be found.

The reorder level s is set at the current estimate of 'maximum reasonable demand during a lead time' and is obtained by multiplying the forecast of expected demand by the lead time and adding a certain quantity for safety stock; i.e.,

Order point = (forecast/period) * (lead time periods) + safety stock. The safety stock is determined by a balance between the cost of a stockout and the cost of carrying the excess inventory. It is logical to establish the safety stock as a function of the success attained in forecasting demand during a lead time, the period from the time the order point is reached until the ordered product is available in inventory. Demand during a lead time implies the joint variability of demand and lead time.

Exponential Smoothing

As the demands placed against inventories under study are probabilistic in nature, it is necessary to develop a time series from the past data to be used as a basis for forecasting demand. A forecast for time t is sought that equals the true demand plus a weighted sum of the noise series of random fluctuations. The noise in each time period is assumed to be from independent probability distributions of mean zero and to attempt to minimize the effect of the random distributions. It is desireable, especially in a system having thousands of items, to have an adaptive technique that uses the most current data to estimate the parameters of the forecasting time series.

Exponential smoothing is one technique for forecasting that meets the above requirements. In this weights are assigned to observations in indirect proportion to their age, thus in a changing process recent data is more valid than older data. In order to determine the new forecast we need only the current forecast, a smoothing constant, and the new observation, thus eliminating the need for carrying large lists of past data. In some data-processing installations, it is a disadvantage to have to carry all the past data necessary to compute a moving average (4), the latter technique is the arithmetic average of the N most recent observations.

In exponential smoothing, the new forecast is the old forecast plus a fraction of the difference between the new observation and the old forecast. The fraction, by which the difference between the new observation and the old forecast is discounted, is designated by the Greek letter alpha, α.

If our estimate of the average demand prior to the current experience is $\bar{D}_{t-1}$ and in the current period t, we experience a demand D_t, then our new estimate or forecast is given by:

$$\bar{D}_t = \alpha D_t + (1 - \alpha)\, \bar{D}_{t-1} \qquad (0 \leq \alpha \leq 1)$$

This operation performed on any sequence of observations is called exponential smoothing. Obviously, with higher alpha factors the forecast is more responsive to the most recent demand. If the alpha factor were 1.0, the new forecast would equal the most recent demand, a factor of 0.5 approximates a three-month moving average, 0.25 about a seven-month moving average, and 0.10 about a 19 month moving average. The determination of the alpha factor is rather complex and often difficult because of the large quantity of data required. An alternative to the analysis is to 'try out' a group of alpha factors and just pick the one which would 'work best'. 'Working best' implies minimizing the overall cost associated with the inventory system. It may involve judgemental estimates of cost parameters.

Exponential smoothing is accurate. The smoothing function minimizes the weighted sum of squared residuals. The computations are simple, requiring a minimum of arithmetic. Finally, exponential smoothing is quite flexible.

Exponential smoothing always requires a previous value of the smoothing function. When the process is started, there must be some value that can be used as the best initial value. If there is no data available to average, a prediction of the average is required. The prediction may be what the process is intended to do - as in the intended sales of a new product. In other cases, the prediction can be based on similarity with other processes that have been observed for some time, as in the case of a new product added to the inventory.

In this study we have chosen the operational characteristics of the inventory system - i.e., inventory level, purchase orders placed and the stockouts experienced as the basic premise of evaluation of the two alternative inventory management systems.

The relevant costs for our inventory problem are: 1) procurement cost (order), 2) carrying cost and 3) out-of-stock cost (lost sale). The order cost C_r is composed of: a) the cost of handling issue and receipt transactions, b) the cost of making and sending the purchase order, c) the cost of expediting and updating, and d) the administrative and overhead costs including the cost involved in making a check on the book inventory. It is assumed that the order cost is independent of order quantity. The order cost is the fixed cost attributed to placing an order with outside supplier. The second class of cost, the carrying cost includes the cost of money tied up in the inventory, storage cost, deterioration cost and insurance cost. It is expressed as the holding cost per dollar of goods per unit of time, which is the average cost incurred in carrying a dollar value of inventories. The unit cost of each of the three products is assumed to be constant and independent of the order quantity. The cost of not carrying inventory is called the out-of-stock cost. The cost resulting from experiencing a demand where there is insufficient stock available is the cost of a stockout.

This cost may be assumed the same as the unit cost. So the total cost of inventory for each item is composed of the three costs, the cost of order, the cost of carrying inventory and the cost of a stockout.

Simulation is used to evaluate the performance of the two inventory management systems. In evaluating the two inventory systems the multi product inventory simulation is carried out for the elapsed time and the performance of the two inventory policies is measured by the total cost.

3. QUEUEING MODEL

The queueing model for the multi-product inventory system is shown in Fig. 1. This is a single-channel, multi-station waiting line problem. This system is encountered generally in a big company, there is one posting department responsible for the replenishment of many warehouses in the vicinity. Since the warehouse and posting department are seldom located at the same place, there is delay in notification of the transactions that have taken place, resulting in a posting delay.

There is a delay between the actual sale and the posting of the transaction. Hence, we encounter two types of record keeping:

1. Physical Stock Record Keeping: A physical inventory is defined as a count of the items of stock for verification of the balances shown in the stock records. This is done at the warehouse, the physical stock is depleted by the sale amount of each item and updated on receipt of an order.
2. Book Stock Record Keeping: This is done at the posting department, the book stock is depleted by the amount of sale, when notified to do so by the warehouse. The book stock is updated on receipt of information from the warehouse. The time a transaction waits to be posted is called posting delay.

The relationship between the actual amount of physical stock maintained in a warehouse and the on-hand balance indicated in the stock records is a matter of serious concern for the efficient operation of the warehouse,

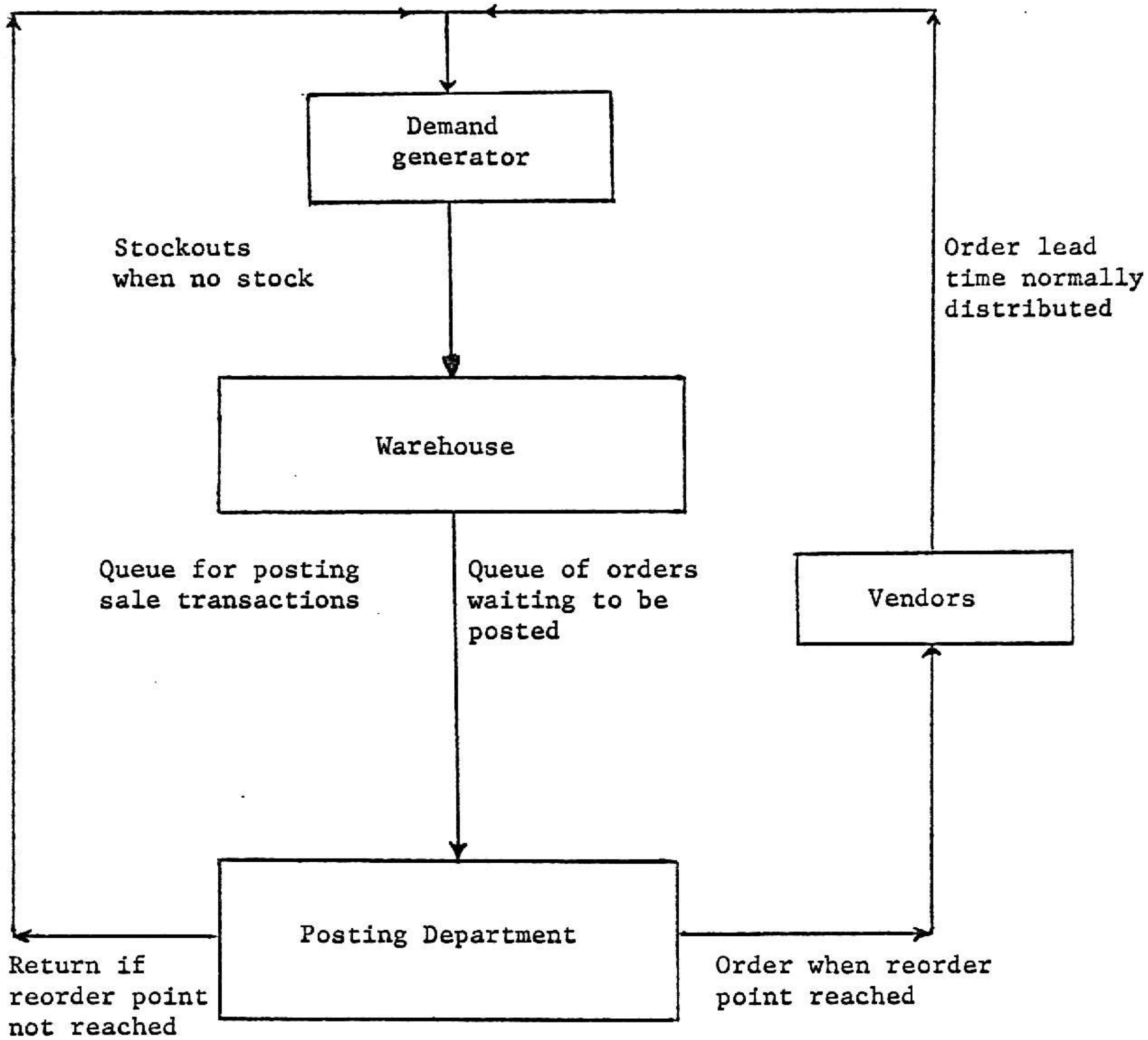


Fig. 1. Queueing model for multi-product simulation.

since the stock records are the basis for all supply and procurement decisions.

The customers are assumed to arrive at the warehouse to buy products available in an exponential fashion, that is the customer's interarrival time follows an exponential distribution with a known mean number of arrivals per unit time, the demand for each product by each customer is assumed to have a normal probability distribution with a known mean and standard deviation. It is assumed that there is no queue of customers waiting for products. If stock is available for the product the customer's demand is met. But if demand could not be filled from stock on hand, the sale is lost resulting in a stockout for the item. The receipt of a sale transaction is sent to the posting department, where it is posted. The posting delay is assumed to follow a normal distribution with a known mean and standard deviation. There is a queue of sale transactions before the posting department. The service time is negligible and included in the posting delay. The posting department as a service channel and the transactions for each item in different waiting lines constitute a single-channel, multi-station queueing system. Book stock is depleted as and when the notice of sale reaches the posting department. The reorder point is checked after each sale posting. When the reorder point is reached, the calculated order quantity is placed with the respective vendor for the product. The lead time, the period from the time the order is placed until the ordered material is available in inventory, is assumed to have a normal distribution, and the delay time of order receipt being posted in book is also assumed to follow a normal distribution.

4. GASP SIMULATION

GASP -- General Activity Simulation Program -- is a collection of subroutines and functions written in FORTRAN and expressly designed for use in simulation studies of industrial systems. The principle advantages offered by GASP are its machine-independence and its modular characteristics, which make it quite easy to expand and alter simulation programs to suit the needs of any given system. Perhaps the greatest benefit provided by GASP is that it is written in a source language and can therefore be recompiled using any compiling system available to a prospective user (12). The version used in this report was developed at the Arizona State University.

GASP has been designed to facilitate "next event" types of simulations. In such simulations, simulated time progresses from one event to another until an end of simulation event occurs, or a preplanned total simulation time is exceeded. An event is an occurrence, a taking place or possibility of taking place of a change in the state of the system. Events take place at specified points in time as determined by the system to be simulated. GASP views the world as consisting of seven basic components (17):

1. Elements
2. Attributes
3. Events
4. Decision Rules
5. Processes

6. States

7. Values

Items that exist in the real world such as people, machines, computers, etc. are called elements, which are described by attributes, and which are acted upon by other elements through events. The events and their effects upon system elements take place over time through the media of logical decision rules and physical processes. These rules and processes depend on the state of the system in time, that is, on the particular values of the element attributes and on various logical and physical parameters that characterize the system. As the model progresses through time certain events generate data that represent changes in the system resulting from the particular data characterization and logical structure of the model. After the period of simulated time the simulation is terminated, and the data examined to evaluate the performance of the model.

GASP provides the simulation model builder with GASP-FORTRAN macroinstructions to accomplish the most important tasks involved in simulation models. The procedures involved in GASP are as follows.

Time Movement and Control

1. Selection of next imminent event to occur in the future.
2. Scheduling of an event to occur sometime in the future.
3. Cancellation of an event that was previously scheduled.
4. Control of start and end of a simulation run.

File Maintenance

1. Filing of items (elements) in queues or waiting lines.

2. Removal of items from queues according to specified priority rules, FIFO for First-in First-out, LIFO for Last-in First-out.
3. Evaluation of attributes of queues (such as "number of units in the queue at any time").

Data Generation

1. Generation of random deviates from Uniform, Normal, Erlang, Poisson and Lognormal distributions.
2. Generation of random numbers as needed.

Input-Output

1. Standard initialization routine for setting parameter values, for setting certain simulation control values, and for sequential running of the simulation model with different data bases.
2. Computation of the mean, the variance, the maximum and the minimum of simulation-generated data.
3. Computation of frequency count (histogram) for simulation-generated data.
4. Automatic reporting at the end of simulation run including informative statistics on all queues used by the model.

Other Procedures

1. Automatic monitoring of program variables and conditions for error detection and debugging.
2. Selective tracing of program flow.
3. Programmed dumping of all system variables.

Timekeeping in GASP

GASP utilizes next event or the event step method which updates the occurrence of each event independent of the time elapsing between events. Essentially, it is a chronological list of events. Each event is considered in strict time order of occurrence. So contrasted to the incremental method of timekeeping, there is now only one list which represents the total length of simulated time. The simulation walks through time on events (15). It is completely time ordered since, as each event is generated, it is filled in order of occurrence through an indexing procedure which does not require rearrangement of the items. The program considers only when an event is to occur and the round-off error depends entirely on the programmer's desire for accuracy or the accuracy inherent to the machine and not upon the length of the increment.

GASP has an array NSET having six columns, for example, a file containing three events is shown in Fig. 2. The members of a file are identified by a pointer system maintained in rows '5' and '6'. The column number of successors of members of the file are given in row '5' with the value 7777 used for the last entry in the file to indicate that it has no successor. The column number of predecessors of members are given in row '6' with the 9999 used for the first entry in the file. By identifying both the first and last entries, processing through the file in either direction is possible. The columns are ranked in the file based on the priority specified. For the three events specified, the priority is FIFO (First-in First-out). The event one (row '2') with the time of occurrence '0' units (row '1') is identified as first entry to be served. Code 9999 (row '6') indicates that it is to be removed next. The successor to this is column two (row '5') having the time of occurrence of '30' units.

The predecessor to this is in column one (row '6'). The last entry in the file identified by 7777 (row '5') is column three. The predecessor to this is column two (row '6'). In this file, columns are ranked in row one.

The functional breakdown of GASP routines is as follows:

FUNCTION	SUBPROGRAM
Information Storage and Retrieval	GASP
	SET
	FILEM
	REMOVE
	FIND
Initialization	DATAIN
Data Collection	COLECT
	TMSTAT
	HISTOG
Monitoring and Error Reporting	MONTR
	ERROR
Statistical Computations and Reporting	PRINTQ
	SUMARY
	OUTPUT
Random Variable Generators	DRAND
	UNIFRM
	NPOISN
	ERLANG
	RNORML
	RLOGNM

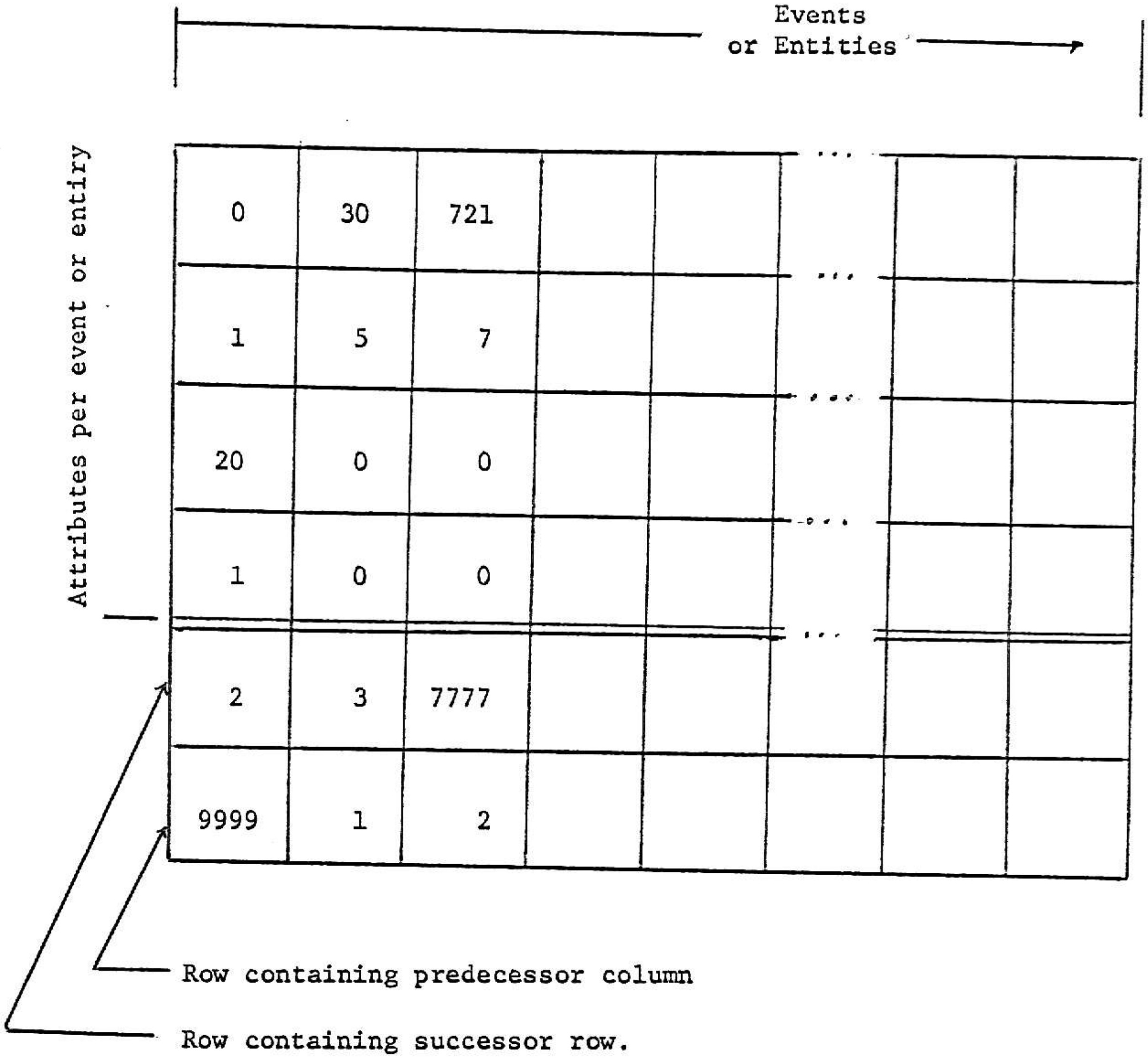


Fig. 2. Graphical representation of the array NSET of GASP.

5. MULTI-PRODUCT SIMULATION

Individual item optimization destroys the collective, system-wide optimization. This system concept is explained in terms of the many item, the many location and the many department effects (20). We have considered the many item effect and simulated three items having different usage rates. The many item effect is felt in a variety of ways. The inventory policy for each item interacts with that of every other item. So considering the large number of items in any typical company it is unrealistic to attempt to achieve an optimal policy for any of the individual items.

Approach

The mean interarrival time is known for the three products together. When an arrival occurs, the next arrival is scheduled by generating the time between arrivals. Each customer is permitted to buy one of the three products at any one time. The products are categorized into three types based on the frequency of usage. Item number '1' is assumed to have a 50 percent usage rate, item number '2', 30 percent and item number '3', 20 percent usage. The simulation was carried out on an IBM 360 using the GASP II simulation language. The specific events of the inventory simulation are:

1. The commodity is sold to the customer (SALE).
2. A sale transaction is posted in book inventory (POST).
3. An order is received by the warehouse from the wholesaler (RECEIV).
4. An order received at the warehouse is recorded in the book inventory (BOOK).

5. Monthly report is prepared for the management listing sales, lost sales, number of waiting units to be posted, etc. (MONTH).
6. Summary of lead time entities is made at the end of receipt of each order at the warehouse (TLEAD).
7. An end of simulation event (ENDSIM), which is used to discontinue the current run and begin another experiment.

Creation of a temporary entity is done by: 1) generation of the arrival time of the next customer, 2) creation of another transaction waiting to be posted in the queue at the posting department, 3) creation of the quantity and delivery time of an order placed, 4) creation of the time when an order received will be posted in the stock book.

Each of the events are described by four attributes:

Attribute '1' : Scheduled time of event

Attribute '2' : Event code

Attribute '3' : Sale quantity in case of a sale transaction and order quantity if an order transaction

Attribute '4' : Product number

A multiple occurrence of the basic events do not take time. For example it is possible for all the events (1,2,3,4,5,6) to occur simultaneously. The multiple events are recorded as taken place at the same simulated time.

The Simulation Program

The simulation consisted of generating customer arrivals for the items. The logical computer commands are used to simulate the inventory control system and to study the performance of detailed record-keeping tasks so as to produce the required information on the system behaviour. Figures 3-1 through 3-10 represent the flowcharts of the simulation program. The computer program for the simulation is shown in Appendix A. In the computer program in addition to 15 GASP routines one main program, one event selection subroutine, six event subroutines and an end of simulation subroutine are used. The function of each of the subroutines can be summarized as follows:

1. Main Program: The initial values of physical stock, book stock for each item and mean interarrival time are read. All the other non-GASP variables are initialized and subroutine GASP is called. The main program is used to read another initial random number in the next run. After the required number of runs, the simulation is terminated. Figure 3-1.
2. Subroutine Events: This calls the appropriate event. Event code '1' signifies an arrival event. Event code '2' signifies a posting event of a sale transaction, code '3' for an order receipt at the warehouse, code '4' for an order posting, code '5' for monthly report, code '6' for lead time summary and code '7' for the end of simulation event. Figure 3-2.
3. Subroutine SALE: When the sale event occurs, it checks for a stockout in case enough stock is not available to meet the demand and creates the posting delay for the current sale transaction and calls the GASP subroutine FILEM to insert the sale quantity with its attributes into the file in the proper

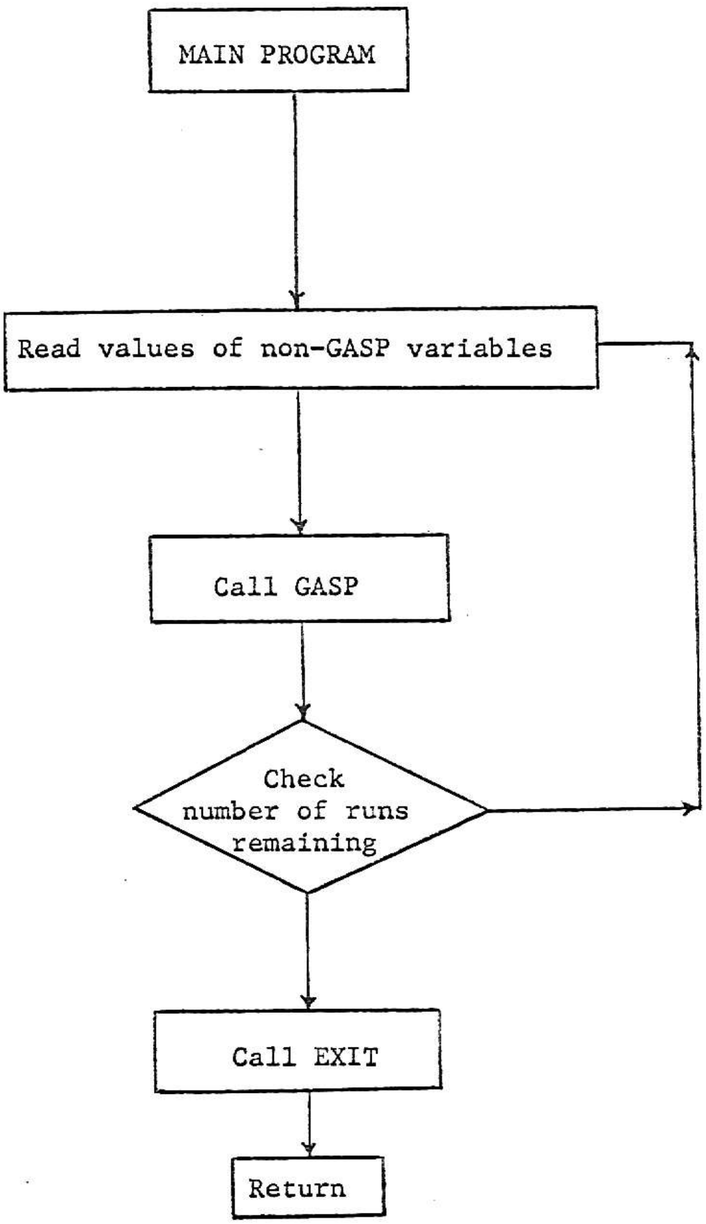
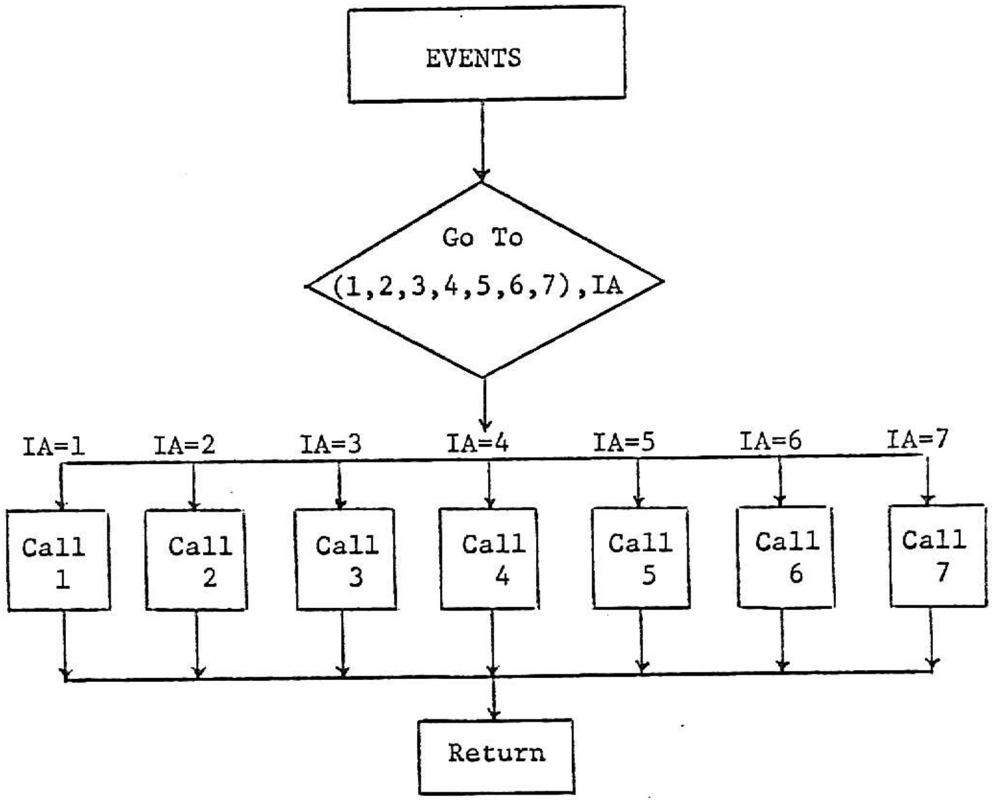


Fig. 3-1. Main Program for multi-product inventory simulation.



1 - SALE

2 - POST

3 - RECEIV

4 - BOOK

5 - MONTH

6 - LTIME

7 - ENDSIM

Fig. 3-2. Subroutine EVENTS.

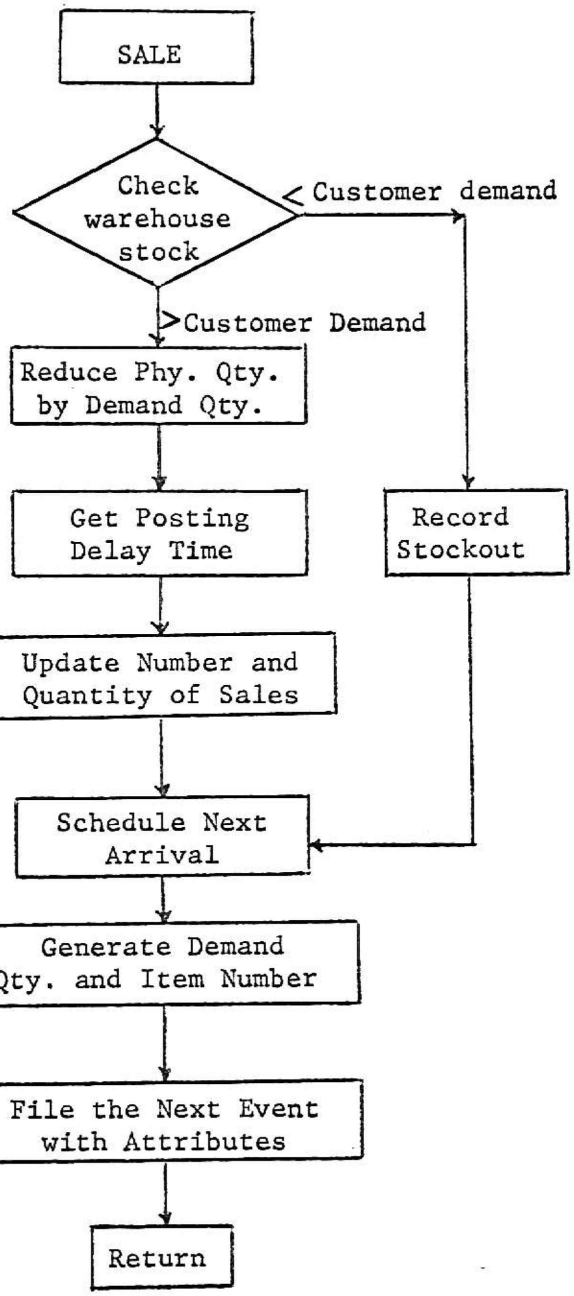


Fig. 3-3. Flow chart of the event SALE.

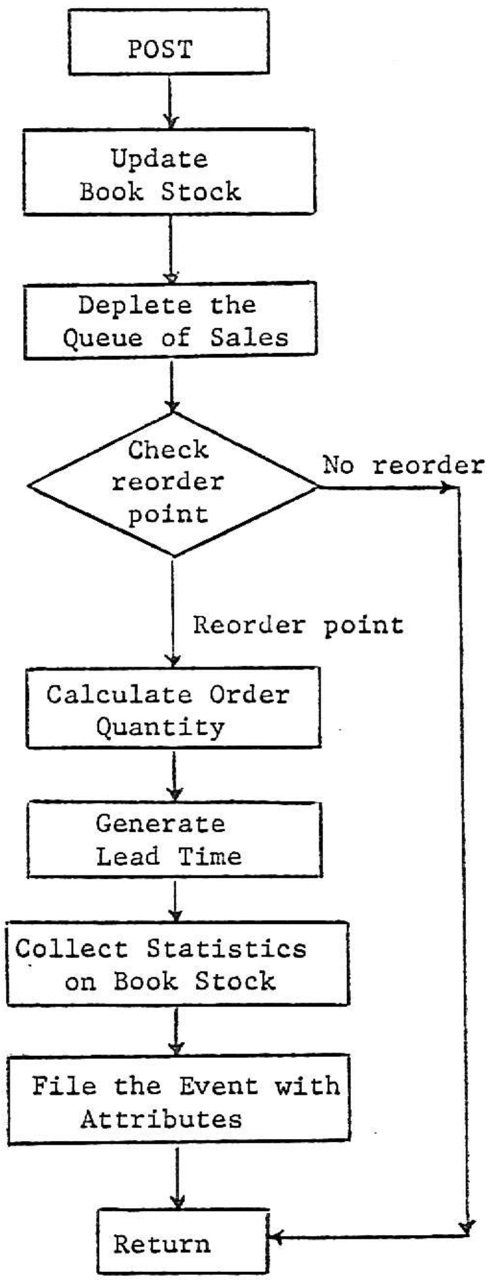


Fig. 3-4. Flow chart of the event POST.

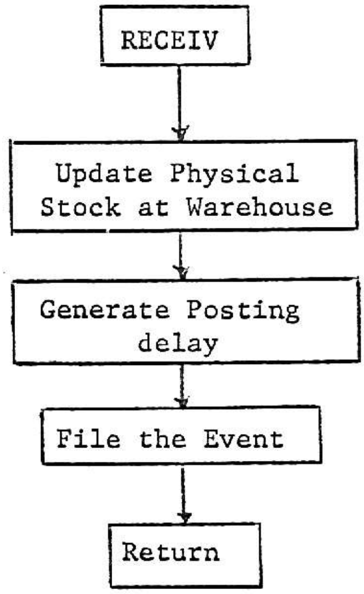


Fig. 3-5. Flow chart of the event RECEIV.

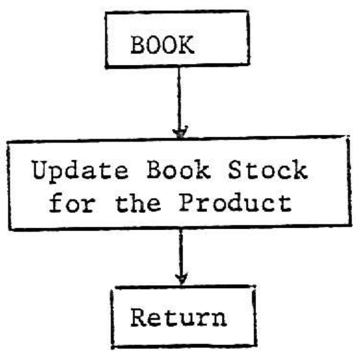


Fig. 3-6. Flow chart for the event BOOK.

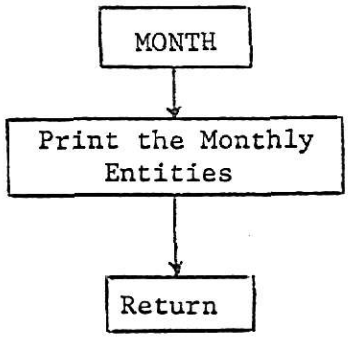


Fig. 3-7. Subroutine MONTH.

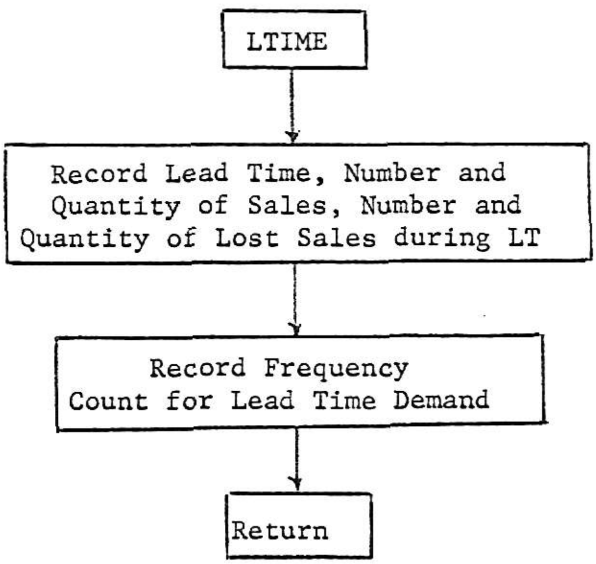


Fig. 3-8. Subroutine LTIME.

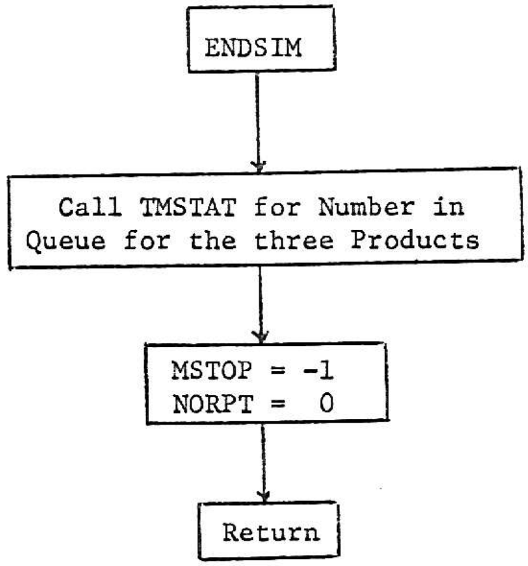


Fig. 3-9. Subroutine End of Simulation.

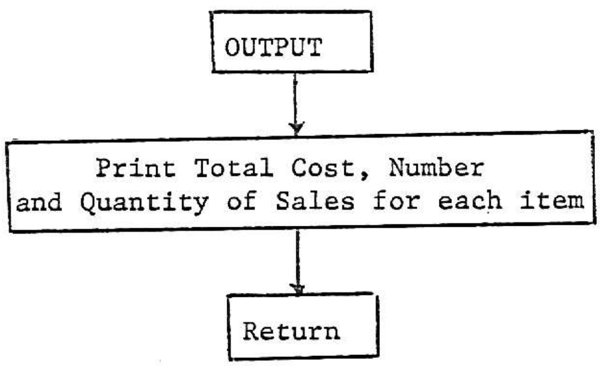


Fig. 3-10. Subroutine OUTPUT.

time order. The next arrival is scheduled by generating the interarrival time, demand quantity and the product number. Interarrival time is generated by using the expression [-ART*ALOG(DRAND(SEED))] where ART is the mean interarrival time. The expression generates random deviates from an exponential distribution with mean ART. The arriving customer for a product is identified randomly on the basis of the usage rate. This event with event time equal to current time plus interarrival time is filed in the array NSET of GASP. Figure 3-3.

4. Subroutine POST: When a sale is posted, the sale quantity is posted to book updating the book stock. After posting, the system stock is tested for reorder point, an order quantity is determined with its delivery time if necessary. Statistics are collected on book stock by calling the GASP subroutine COLECT. Figure 3-4.

5. Subroutine RECEIV: When an order is received at the warehouse, the physical stock is updated. It generates the posting delay when the order received at the warehouse will be posted to the stock book. Figure 3-5.

6. Subroutine BOOK: When an event book occurs, book stock for the product is updated. Figure 3-6.

7. Subroutine MONTH: At the end of each month, it lists the monthly records of sales, stockouts and queue lengths. Figure 3-7.

8. Subroutine Tlead: Records leadtime, sales during lead time and stockouts. Figure 3-8.

9. Subroutine Endsim: Terminates the run and gives the code to GASP to print all GASP output. Figure 3-9.

10. Subroutine OUTPUT: This calculates and prints total cost for each item and prints the total cost in dollars for the three products. Figure 3-10.

Fifteen GASP subroutines are used in the inventory simulation. The function of each of the subroutines used is briefly as follows:

1. GASP: Determines the next event and controls the simulation from start to end. It controls the monitoring of intermediate simulation results. Only one file is used in the simulation. It is used to store scheduled events.

2. FILEM: Stores the event vector of attributes in file NSET. It files the events in the first available column in the array NSET.

3. REMOVE: It removes the event vector from the file at the command of GASP.

4. SET: This accomplishes the storage and retrieval function of the file. SET provides a filing system.

5. DATAIN: Initializes GASP variables and reads all input data.

6. PRINTQ: Prints average number and maximum number of entires in the file.

7. COLECT: Collects the statistical data for the specified variables identified by codes. Statistics are collected on physical stock, book stock and system stock for each item.

8. TMSTAT: Collects time statistics between changes in the value of the variables. Time statistics are collected for the number in the queues at the posting department.

9. HISTOG: Provides histograms. Histograms are prepared for lead time demand distribution for the three products.

10. MONTR: Monitors the program by tracing the events during simulation.

11. ERROR: Errors are specified by codes. The program is terminated by an error condition.

12. SUMMARY: Prints the final report.

13. OUTPUT: Dummy routine. It is used to print total cost for the simulated time.

14. RNORML: Generates a random deviate from a normal distribution with specified parameters. It is used to generate normal deviates for demand, posting delay and lead time.

15. DRAND: Generates a random number. We have incorporated a random number generator into DRAND.

6. RESULTS

For the use of management, simulation reports are required to observe the behaviour of the inventory system through time. Monthly reports, lead time reports, and GASP reports were prepared. For evaluating the two inventory systems the total costs were first calculated for the period of one year (360 time units). High variability in the total cost was found. So the total simulation time was set at 720 units to overcome the initial transient conditions. In this system, the transient conditions were overcome after the completion of approximately four reorder cycles for each item.

The initial values of the simulation are shown in Appendix B. At the end of each month (30 time units) summary totals are printed, listing: 1) the time at which the summary was made; 2) the actual demand for each item during the period; 3) the number and quantity of actual sales during the period; and 4) the number of units waiting in each waiting line at that time. Also whenever an order is received by the warehouse summary totals during a lead time are printed listing the quantity of sales, quantity of lost sales, book sales and the number of units waiting in the queue at that time. The results are shown for a period of one year (360 units). GASP reports with codes are prepared. Statistics are collected for physical stock, book stock and system stock. Time statistics are collected for the number of units in the queues of the three items during the simulated time. Histograms are prepared for lead time demand distribution. Queue contents are printed after the termination of simulation. In the end total costs are written from subroutine OUTPUT.

A series of simulation runs were made using the two inventory management systems. For exponential smoothing, alpha values 0.05, 0.10, 0.15, 0.20 and 0.25 were tried. The review period was one month. Ten runs were made with different initial random numbers at each alpha value. The total costs for all the runs with means and standard deviations are listed in Table 1. The inventory system was simulated through a period of two years in each run.

The inventory system was simulated using the (s,S) system with review periods of 20, 30, and 40 days. The total costs for all the runs are listed in Table 2.

The total cost in each condition is shown graphically in Fig. 4. The total cost was highest at the review period of 20 days and lowest at 40 days.

Higher values of variances were obtained in total cost. An L-test (10) indicated that the within sample variances are homogeneous, Table 3.

The total cost was higher at alpha values of 0.05 and 0.10. in exponential smoothing. The total cost was found to be statistically equal under the other conditions (alpha values 0.15, 0.20 and 0.25 and review periods 30 days and 40 days in the (s,S) system). Table 4.

To investigate the variability in the experimental errors, two sets of simulations were carried out with alpha value 0.25. The first run was for a period of two years and the second run was only for one year with the same set of random numbers. The total cost for the

Table 1

Total Cost Using Exponential Smoothing Technique

Run No.	Alpha values 0.05	0.10	0.15	0.20	0.25
1	5033.27	5571.82	3770.71	4349.36	4206.45
2	4385.68	5743.19	4397.95	3439.65	4450.14
3	5133.39	4231.78	3622.22	4241.43	4454.84
4	6267.46	4367.52	3885.40	3599.98	3279.27
5	5966.35	4451.04	3023.53	3521.11	2441.54
6	3463.80	3239.53	5005.50	4898.49	3624.12
7	4755.72	5246.70	4149.25	5375.86	4206.50
8	5112.30	4384.61	6097.98	4861.93	3737.74
9	4302.19	4454.61	5138.73	6132.27	4027.89
10	4950.28	6185.09	5405.45	4806.36	3901.45
Means	4937.04	4787.58	4449.67	4522.64	3832.99
Std. Devs.	804.05	879.41	943.10	868.72	613.24

Table 2

Total Cost Using (s,S) System

Run No.	Review Period 20	30	40
1	5091.25	4770.54	2796.34
2	4519.72	2843.03	3598.88
3	5422.16	3100.22	4973.41
4	4393.04	4458.61	4518.20
5	7009.33	3166.00	2320.46
6	3835.42	3838.05	4230.89
7	7493.48	5356.94	3605.10
8	3927.29	4933.00	2875.27
9	5332.59	5928.67	3925.37
10	3771.25	3589.80	4121.55
Means	5079.51	4981.48	3696.55
Std. Devs.	1295.52	1047.74	831.34

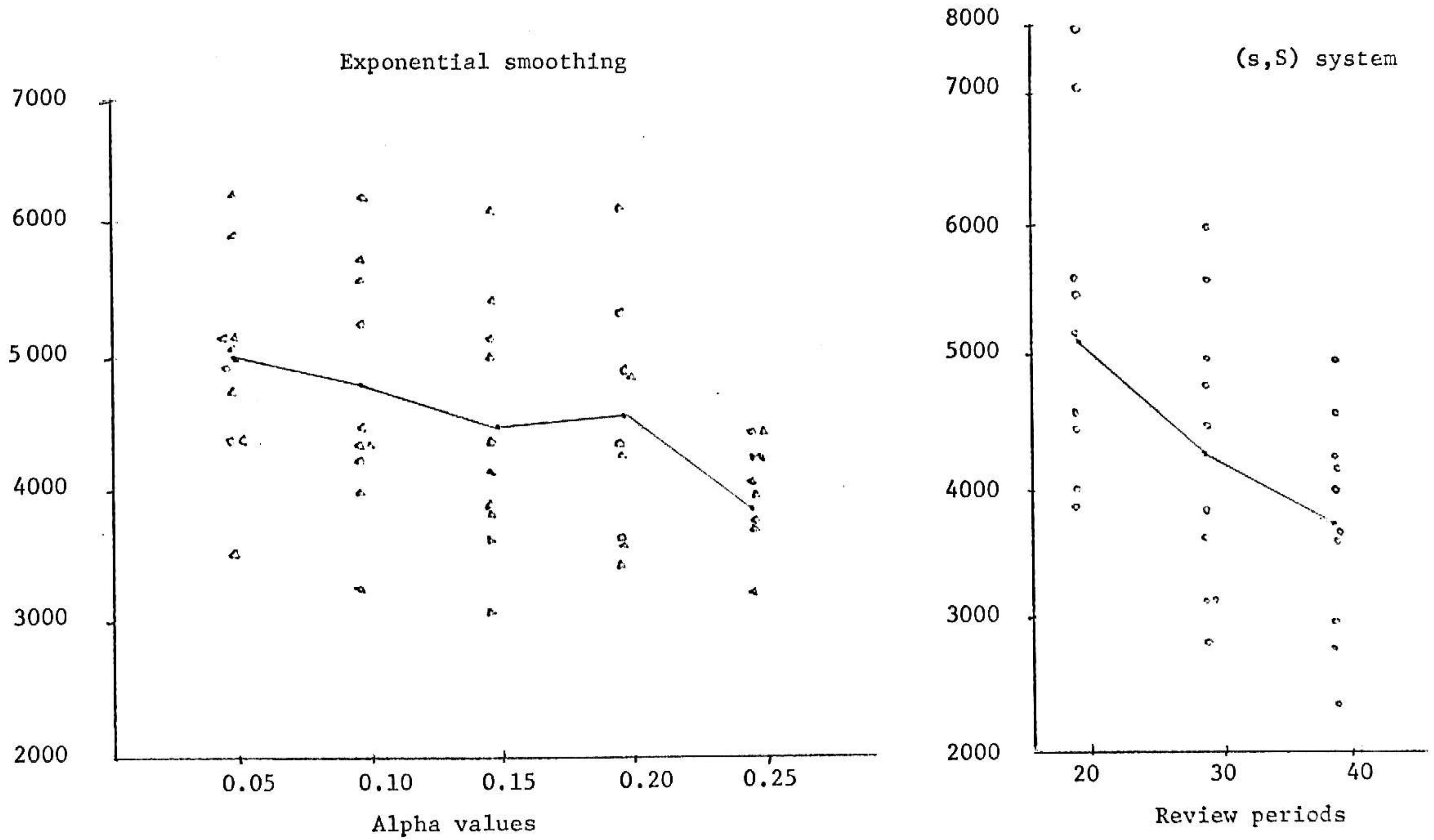


Fig. 4. Total Cost Using Two Inventory Management Systems.

Table 3

L-Test for Homegeneity of Variances

k = number of samples (8)

n = number of observations in each sample (10)

S_g^2 = geometric mean of the within-sample variances

S_a^2 = arithmetic mean of within-sample variances

$$L_1 = \frac{S_1^2 \cdot S_2^2 \ldots S_k^2}{S_a^2 \cdot S_a^2 \ldots S_a^2} \quad 1/k$$

$$\text{Log } L_1 = \log S_g^2 - \log S_a^2$$

$$\text{Log } S_g^2 = (4.81 + 4.89 + 4.95 + 4.58 + 4.22 + 5.03 + 4.22 + 5.83)/8$$

$$= \frac{37.54}{8} = 4.8175$$

$$S_a^2 = (64{,}650.5 + 77{,}337.5 \pm 88{,}945.0 + 75{,}467.8 + 37{,}607.3 + 167{,}833.8 + 109{,}776.2 + 69{,}113.5)/8$$

$$= 86342.0$$

$$\text{Log } S_a^2 = 4.9362$$

$$\text{Log } L_1 = \log S_g^2 - \log S_a^2$$

$$= 4.6925 - 4.9362$$

$$= -0.2437$$

$$L_1 = 0.8785$$

Tabled value of L_1(k=8, n=10) at 5 percent level = 0.8115

Decision: All the within-sample variances are equal.

Table 4

LAD Procedure for Multiple Comparison of Total Costs

$$LSD = 2.00 * \ 2 * \frac{863416.00}{10}$$

$$= 831.10$$

Mean Total Cost

3696.55	(40 days review period in (s,S) system)
3832.99	(alpha 0.25 in exponential smoothing)
4198.48	(30 days review period in (s,S) system)
4449.69	(alpha 0.15 in exponential smoothing)
4522.64	(alpha 0.20 in exponential smoothing)
4787.58	(alpha 0.10 in exponential smoothing
4937.04	(alpha 0.05 in exponential smoothing)
5079.51	(20 days review period in (s,S) system)

Table 5

Experimental Error

Alpha: .25			
	Two Years Total Cost	First Year Total Cost	Second Year Total Cost
	4206.45	2249.40	1956.46
	4450.14	2769.99	1680.15
	4454.84	2880.02	1574.82
	3279.27	1415.16	1864.11
	2441.54	1224.75	1216.79
	3624.12	2083.63	1540.49
	4206.50	2175.78	2030.72
	3737.74	2436.12	1301.62
Means	3833.99	2154.35	1643.39
Std. dev.	613.24	547.02	274.90

F-test for the two variances
(two year total cost and the second year total cost)

The two year total variance (S_1^2) = 376,073.81

The convoluted variance for two years of second year total cost (S_2^2): 111,122.73

$$F(7,7) = \frac{S_1^2}{S_2^2} = \frac{376{,}073.81}{111{,}122.73}$$

$$= 3.38$$

Tabled value of F .90(7,7): 2.78

Decision: The two variances are unequal.

second year (after the system has overcome the transient conditions) was obtained by deducting first year total cost from the two year total cost. The variance of the second year total cost when convoluted to two years was found to possess less variability and found significantly lower than the two year variance obtained from simulation. The variance was convoluted so that the variances could be compared for the same period. Much of the simulation variability in the two year simulation total cost can be attributed to the existing transient conditions, Table 5.

7. CONCLUSIONS

GASP is flexible and the multi-product inventory system was simulated without a great amount of computer programming. Even though GASP II was written for GE 225, it was used on the IBM 360 with a few changes in some subroutines. There was no scaling problem when the attributes were stored as a fixed point array. To handle larger size problem GASP can be extended by changing the dimensions of the variables.

The statistics collected on physical stock and book stock would be useful to find the correlation between physical stock and book stock. The time statistics on queue lengths can be used as a good measure of system effectiveness. The histograms provide the means of determining the lead time demand distribution which is used in setting reorder points for the products. However, in the simulations the few observations available could not provide lead time demand distribution. As a whole the multi-product inventory simulation was run satisfactorily with GASP.

Even though the experimental variation was high, the variances were found to be homogeneous. Investigation of the variability was carried out by eliminating the effect of transient conditions, which exist at the initial stages of the simulation. Total cost was found to possess reduced variability after the effect of transient conditons is removed. Some of the remaining variability might be due to the compounded variability in demand, posting delay and lead time. From the numerical results both the inventory policies were found to be equally effective for the inventory problem. But alpha value of 0.25

was found to have lower total cost (total cost of $3832.99 in the exponential smoothing when compared to $4198.48 in the (s,S system)). Keeping the review period at 30 days both of the inventory management systems, the exponential smoothing technique and the (s,S) system, have appreciably the same total cost. As far as the implementation cost is concerned, both the inventory systems would involve the same cost. The exact model for the (s,S) model with variable review period is available which requires a computer to set the operating inventory levels. But as the implementation cost may be more than all the inventory connected cost which appear in the sophisticated model, we have used the approximate formula in the (s,S) system to set the maximum inventory operating levels. It can be said that in a system having thousands of items, it would be desirable to use an adaptive technique like the exponential smoothing technique.

Further research should be directed toward eliminating much of the variability of experimental errors.

ACKNOWLEDGEMENT

The author wishes to express his sincere gratitude to his Major Professor, Dr. L. E. Grosh, for his guidance and helpful assistance during the preparation of this report and for the time and energy devoted for counsel.

REFERENCES

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

2. Brown, R.G., "Statistical Forecasting for Inventory Control", McGraw-Hill, 1959.

3. _____, "Decision Rules for Inventory Management", Holt, Rinehart and Winston, N.Y., 1967, pp. 237-239.

4. _____, "Smoothing, Forecasting and Prediction of Discrete Time Series", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

5. Buffa, E.S., "Modern Production Management", John Wiley & Sons, Inc., N.Y., 2nd Ed., 1961.

6. Chu K. and Naylor, T.H., "Two Alternate Methods for Simulating Waiting Line Models", The Journal of Industrial Engineering, Vol. XVI., No. 6, Nov. - Dec., 1965, pp. 390-394.

7. Dvoretzky, J.K. and Witz, J.W., "On the Optimal Character of the (s,S) Policy in Inventory Theory", Econometrica, Vol. 21, No. 4, Oct., 1953, pp. 586-596.

8. Dzielinski, B.P. and Manne, A.S., "Simulation of a Hypothetical Multi-Item Production and Inventory System", The Journal of Industrial Engineering", Vol. XVI, No. 6, Nov. - Dec., 1961, pp. 118-125.

9. Evans, G.W. Wallace, G.F. and Sutherland, G.L., "Simulation Using Digital Computers", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967, pp. 118-125.

10. Freeman, H.A., "Industrial Statistics", John Wiley & Sons, N.Y., 1947, pp. 85-87.

11. Hadley, G. and Whitin, T.M., "Analysis of Inventory Systems", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

12. Kiviat, P.J. and Colker, A., "GASP - A General Activity Simulation Program", The Rand Corporation, P-2684, February, 1964.

13. Kiviat, P.J., "Development of New Digital Simulation Languages", The Journal of Industrial Engineering, Vol. XVII, No. 11, November, 1966.

14. Krasnow, H.S. and Merikallio, R.A., IBM Corp., "The Past, Present and Future of General Simulation Languages", Management Science, Vol. 11, No. 2, November 1964, pp. 236-261.

15. Lave, R.E., "Timekeeping for Simulation", The Journal of Industrial Engineering, Vol. XVIII, No. 7, July, 1967, pp. 391-394.

16. McMillan, C. and Gonzalez, R.F., "Systems Analysis", A Computer Approach to Decision Models, Richard D. Irwin, Inc., Homewood, Illinois, 1965.

17. Naylor, T.H., Balintfy, J.L., Burdick, D.S. and Chu, K., "Computer Simulation Techniques", John Wiley & Sons, Inc., N.Y., 1966

18. Packer, A.H., "Simulation and Adaptive Forecasting as Applied to Inventory Control", Operations Research, Vol. 15, No. 4, July - August, 1967, pp. 660-679.

19. Pritsker, A.A.B. and Kiviat, P.J., "GASP II - A Fortran Based Simulation Language", Department of Industrial Engineering, Arizona State University, September, 1967.

20. Starr, M.K. and Millar, D.W., "Inventory Control: Theory and Practice", Prentice-Hall International, Inc., Cliffs, N.J., 1962.

APPENDIX A

```
C     THIS PROGRAM SIMULATES A MULTI-PRODUCT INVENTORY SYSTEM USING GASP
C     THIS MAIN PROGRAM INITILIZES ALL NON-GASP VARIABLES
      COMMCN ID,IM,INIT,JEVENT,JMCNIT,MFA,MSTCP,MX,MXC,NCCLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNCW,TSTART,TSTO
     2P,MXX
      COMMCN ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMCN IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMCN JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,50)
   50 FORMAT(9I5,F5.1,6I5/6I5,F5.1)
   51 FORMAT(14I4,2F4.1,I10)
   52 FORMAT(6I4)
  111 FORMAT(4F10.2)
      NKP=C
  105 ALPHA=0.05
  103 CONTINUE
      READ 50,IST(1),IST(2),IST(3),IBS(1),IBS(2),IBS(3),ISS(1),ISS(2),
     1ISS(3),ART,IR(1),IR(2),IR(3),IMAX(1),IMAX(2),IMAX(3),JQOL(1),
     2JQOL(2),JQOL(3),NOL(1),NOL(2),NOL(3),TIP
      READ 51,LBR,LBQR,NOS(1),NOS(2),NOS(3),JQS(1),JQS(2),JQS(3),
     1JBS(1),JBS(2),JBS(3),NOBS(1),NOBS(2),NOBS(3),TIP1,TIP2,IX
      READ 52,LBQ(1),LBQ(2),LBQ(3),LBQQ(1),LBQQ(2),LBQQ(3)
      DO 110 I=1,3
  110 READ 111,CR(I),C(I),CC(I),CS(I)
      OLT1=0
      INCRMT=30
      JM=1
      NOR(1,1)=680
      NOR(1,2)=330
      NOR(1,3)=185
      DO 54 I=1,12
```

```
      DO 54 JI=1,50
      OLTLT(JI,I)=0
      JQSLT(JI,I)=0
      JASLT(JI,I)=0
      NOSLT(JI,I)=0
      NOBSLT(JI,I)=0
      JBSLT(JI,I)=0
      JQSLT1(JI,I)=0
      JADLT1(JI,I)=0
      NOSLT1(JI,I)=0
      NOBLT1(JI,I)=0
   54 JBSLT1(JI,I)=0
      DO 3 I=1,3
      NOS1(I)=0
      JQS1(I)=0
      JAD(I)=0
      JAD1(I)=0
      JSO(I)=0
      NOL1(I)=0
      JQOL1(I)=0
      NOBS1(I)=0
      JBS1(I)=0
      LBQ1(I)=0
      LBQQ1(I)=0
      NOS2(I)=0
      JQS2(I)=0
      JAD2(I)=0
      NOL2(I)=0
      JQOL2(I)=0
      NOBS2(I)=0
      JBS2(I)=0
      LBQ2(I)=0
      JOR(I)=0
    3 LBQQ2(I)=0
      CALL GASP(NSET)
      ALPHA=ALPHA+0.05
      IF(ALPHA.LE.0.20) GO TO 103
  102 CONTINUE
      CALL EXIT
      END
```

```
      SUBRCUTINE EVENTS(IA,NSET)
C     THIS CALLS THE NEXT EARLIEST EVENT
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNOW,TSTART,TSTO
     2P,MXX
      COMMCN ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMCN IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMCN JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      GO TC (1,2,3,4,5,6,7),IA
    1 CALL SALE(NSET)
      RETURN
    2 CALL POST(NSET)
      RETURN
    3 CALL RECEIV(NSET)
      RETURN
    4 CALL BOOK(NSET)
      RETURN
    5 CALL MONTH(NSET)
      RETURN
    6 CALL TLEAD(NSET)
      RETURN
    7 CALL ENDSIM(NSET)
      RETURN
      END
```

```
      SUBROUTINE SALE(NSET)
C     THIS GENERATES POSTING DELAY AND SCHEDULES NEXT ARRIVAL
C     IT COLLECTS STATISTICS ON PHYSICAL STOCK AND TIME STATISCS ON THE
C     QUEUE OF TRANSACTIONS WAITING BEFORE POSTING DEPARTMENT
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNOW,TSTART,TSTO
     2P,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMON IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMON JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      PRODNC=0
      D=0
      ISL=0
      T=(-ART*ALOG(DRAND(SEED)))
      NPROD=ATTRIB(4)
      I=NPROD
      IF(IST(I)-ATTRIB(3))101,101,102
  101 NOL(I)=NOL(I)+1
      JQOL(I)=JQOL(I)+ATTRIB(3)
      JSL=0
      D=0
      GO TO 58
  102 JSL=ATTRIB(3)
      IST(I)=IST(I)-JSL
      JQS(I)=JQS(I)+JSL
      JAD(I)=JQS(I)+JQOL(I)
      NOS(I)=NOS(I)+1
      OLBQ=LBQ(I)
      IF(I.EQ.1) CALL TMSTAT(OLBQ,TNOW,1,NSET)
      IF(I.EQ.2) CALL TMSTAT(OLBQ,TNOW,2,NSET)
```

```
      IF(I.EQ.3) CALL TMSTAT(OLBQ,TNOW,3,NSET)
      LBQ(I)=LBQ(I)+1
      LBQQ(I)=LBQQ(I)+JSL
      IF(I.EQ.1) D=RNORML(2)
      IF(I.EQ.2) D=RNORML(6)
      IF(I.EQ.3) D=RNORML(10)
      ATTRIB(1)=TNOW+D
      ATTRIB(2)=2
      ATTRIB(3)=JSL
      CALL FILEM(1,NSET)
   58 CONTINUE
      IF(I.EQ.1) ISL=RNORML(4)+0.5
      IF(I.EQ.2) ISL=RNORML(8)+0.5
      IF(I.EQ.3) ISL=RNORML(12)+0.5
      ATTRIB(1)=TNOW+T
      ATTRIB(2)=1
      ATTRIB(3)=ISL
      PRODNO=DRAND(SEED)
      IF(0.50-PRODNO)11,10,10
   10 ATTRIB(4)=1
      GO TO 19
   11 IF(0.80-PRODNO)13,12,12
   12 ATTRIB(4)=2
      GO TO 19
   13 ATTRIB(4)=3
   19 CALL FILEM(1,NSET)
      OIST=IST(I)
      IF(I.EQ.1) CALL COLECT(OIST,7,NSET)
      IF(I.EQ.2) CALL COLECT(OIST,8,NSET)
      IF(I.EQ.3) CALL COLECT(OIST,9,NSET)
      RETURN
      END
```

```
      SUBRCUTINE POST(NSET)
C     IT POSTS THE SALE TRANSACTIONS AND CHECKS FOR REORDER POINT
C     IF REORDER POINT IS REACHED, A CALCULATED ORDER QUANTITY IS
C     PLACEC WITH THE VENDOR.SYSTEM STOCK AND BOOK STOCK STATISTICS
C     ARE COLLECTED.
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNOW,TSTART,TSTO
     2P,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNC(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMCN IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NCBLT1(50,12)
      COMMCN JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      JQB=ATTRIB(3)
      NPROC=ATTRIB(4)
      I=NPRCD
      IBS(I)=IBS(I)-JQB
      ISS(I)=ISS(I)-JQB
      JBS(I)=JBS(I)+JQB
      NOBS(I)=NOBS(I)+1
      OLBQ=LBQ(I)
      IF(I.EQ.1) CALL TMSTAT(OLBQ,TNOW,1,NSET)
      IF(I.EQ.2) CALL TMSTAT(OLBQ,TNOW,2,NSET)
      IF(I.EQ.3) CALL TMSTAT(OLBQ,TNOW,3,NSET)
      LBQ(I)=LBQ(I)-1
      LBQQ(I)=LBQQ(I)-JQB
      IF(ISS(I)-IR(I))110,110,111
  110 CONTINUE
      IF(JM-1)112,112,113
  112 JSO(I)=NOR(JM,I)
      GO TC 114
  113 JSO(I)=ALPHA*JADLT1(JM,I)+(1-ALPHA)*JSO(I)
```

```
114 IF(I.EQ.1) OLT1=RNORML(1)
    IF(I.EQ.2) OLT1=RNORML(5)
    IF(I.EQ.3) OLT1=RNORML(9)
    ISS(I)=ISS(I)+JSO(I)
    ATTRIB(1)=TNOW+OLT1
    ATTRIB(2)=3
    ATTRIB(3)=JSO(I)
    CALL FILEM(1,NSET)
    JOR(I)=JOR(I)+1
    TIP=0
    CALL TLEAD(NSET)
    TIP=1
111 CONTINUE
    OISS=ISS(I)
    IF(I.EQ.1) CALL COLECT(OISS,1,NSET)
    IF(I.EQ.2) CALL COLECT(OISS,2,NSET)
    IF(I.EQ.3) CALL COLECT(OISS,3,NSET)
    OIBS=IBS(I)
    IF(I.EQ.1) CALL COLECT(OIBS,4,NSET)
    IF(I.EQ.2) CALL COLECT(OIBS,5,NSET)
    IF(I.EQ.3) CALL COLECT(OIBS,6,NSET)
    RETURN
    END
```

```
      SUBROUTINE RECEIV(NSET)
C     UPDATES PHYSICAL STOCK AND COLLECTS STATISTICS ON PHYSICAL STOCK
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNOW,TSTART,TSTO
     2P,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMON IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMON JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      NPROD=ATTRIB(4)
      I=NPROD
      JSTOCK=ATTRIB(3)
      IST(I)=IST(I)+JSTOCK
      IF(I.EQ.1) OLT2=RNORML(3)
      IF(I.EQ.2) OLT2=RNORML(7)
      IF(I.EQ.3) OLT2=RNORML(11)
      ATTRIB(1)=TNOW+OLT2
      ATTRIB(2)=4
      CALL FILEM(1,NSET)
      OIST=IST(I)
      IF(I.EQ.1) CALL COLECT(OIST,7,NSET)
      IF(I.EQ.2) CALL COLECT(OIST,8,NSET)
      IF(I.EQ.3) CALL COLECT(OIST,9,NSET)
      RETURN
      END
```

```
      SUBRCUTINE BOOK(NSET)
C     AN ORDER RECEIVED AT WAREHOUSE IS POSTED IN BOOK.STATISTICS ON
C     BOOK STOCK ARE COLLECTED
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NCQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNOW,TSTART,TSTO
     2P,MXX
      COMMCN ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMCN IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMCN JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      NPROC=ATTRIB(4)
      I=NPRCD
      JSTOCK=ATTRIB(3)
      IBS(I)=IBS(I)+JSTOCK
      OIBS=IBS(I)
      IF(I.EQ.1) CALL COLECT(OIBS,4,NSET)
      IF(I.EQ.2) CALL COLECT(OIBS,5,NSET)
      IF(I.EQ.3) CALL COLECT(OIBS,6,NSET)
      RETURN
      END
```

```
      SUBRCUTINE MONTH(NSET)
      THIS PRINTS MONTHLY SALES, LCST SALES AND BOOK SALES
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTCP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNOW,TSTART,TSTO
     2P,MXX
      COMMCN ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMCN IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMCN JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      JM=JM+1
      DO 1C I=1,3
      NOS2(I)=NOS(I)-NOS1(I)
      JQS2(I)=JQS(I)-JQS1(I)
      JAD2(I)=JAD(I)-JAD1(I)
      NOR(JM,I)=JAD2(I)
      NOL2(I)=NOL(I)-NOL1(I)
      JQOL2(I)=JQOL(I)-JQOL1(I)
      NOBS2(I)=NOBS(I)-NOBS1(I)
      JBS2(I)=JBS(I)-JBS1(I)
      LBQ2(I)=LBQ(I)
   10 LBQQ2(I)=LBQQ(I)
      DO 8 I=1,3
    8 PRINT 21,TNOW,NOS2(I),JQS2(I),JAD2(I),NOL2(I),JQOL2(I),NOBS2(I),
     1JBS2(I),LBQ2(I),LBQQ2(I),I
   21 FORMAT(1H ,F8.2,I7,2I8,I7,I8,I7,I8,I7,I8,48X,I8)
      ATTRIB(1)=TNOW+INCRMT
      ATTRIB(2)=5
      ATTRIB(3)=0
      ATTRIB(4)=0
      CALL FILEM(1,NSET)
      DO 9 I=1,3
```

```
      NOS1(I)=NOS(I)
      JQS1(I)=JQS(I)
      JAD1(I)=JAD(I)
      NOL1(I)=NOL(I)
      JQOL1(I)=JQOL(I)
      NOBS1(I)=NOBS(I)
      JBS1(I)=JBS(I)
      LBQ1(I)=LBQ(I)
    9 LBQQ1(I)=LBQQ(I)
      RETURN
      END
```

```
      SUBRCUTINE TLEAD(NSET)
C     THIS COLLECTS SUMMARY OF LEAD TIME DEMAND AND MAKES HISTOGRAMS
C     OF LEAD TIME DEMAND
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNCW,TSTART,TSTO
     2P,MXX
      COMMCN ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMCN IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMCN JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      I=ATTRIB(4)
      IF(TIP)26,26,27
   26 JI=JCR(I)
      OLTLT(JI,I)=OLT1
      JQSLT(JI,I)=JQS(I)
      JASLT(JI,I)=JAD(I)
      NOSLT(JI,I)=NOS(I)
      NOBSLT(JI,I)=NOBS(I)
      JBSLT(JI,I)=JBS(I)
      ATTRIB(2)=6
      ATTRIB(3)=JI
      CALL FILEM(1,NSET)
      RETURN
   27 JI=ATTRIB(3)
      OLTLT(JI,I)=OLTLT(JI,I)
      JQSLT1(JI,I)=JQS(I)-JQSLT(JI,I)
      JADLT1(JI,I)=JAD(I)-JASLT(JI,I)
      ADLT1=JADLT1(JI,I)
      IF(I.EQ.1) CALL HISTOG(ADLT1,40.,5.5,1)
      IF(I.EQ.2) CALL HISTOG(ADLT1,30.,6.,2)
      IF(I.EQ.3) CALL HISTOG(ADLT1,15.,2.,3)
```

```
      NOSLT1(JI,I)=NOS(I)-NOSLT(JI,I)
      NOBLT1(JI,I)=NOBS(I)-NOBSLT(JI,I)
      JBSLT1(JI,I)=JBS(I)-JBSLT(JI,I)
      PRINT 25,OLTLT(JI,I),JQSLT1(JI,I),JADLT1(JI,I),NOSLT1(JI,I),
     1NOBLT1(JI,I),JBSLT1(JI,I),I
   25 FORMAT(1H ,76X,F8.2,6I8)
      RETURN
      END
```

```
      SUBRCUTINE ENDSIM(NSET)
C     A MEAN OF TERMINATING THE SIMULATION
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNOW,TSTART,TSTO
     2P,MXX
      COMMCN ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMCN IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMCN JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      MSTOP=-1
      NORPT=0
      RETURN
      END
```

```
      SUBRCUTINE OUTPUT(NSET)
C     PRINTS THE TOTAL COST FOR THE PERICD SIMULATED
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,SEED,TNOW,TSTART,TSTO
     2P,MXX
      COMMCN ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),MAXNQ(4),MFE(
     14),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),QTIME(4),SSUMA(10,5)
     2,SUMA(10,5)
      COMMCN IST(12),IBS(12),ART,IR(12),IMAX(12),JQS(12),JQOL(12),NOL(12
     1),NOS(12),ISS(12),JBS(12),NOBS(12),LBQ(12),LBQQ(12),TIP,TIP1,IX,
     2TIP2,INCRMT,JAD(12),NOS1(12),JQS1(12),JAD1(12),NOL1(12),JQOL1(12),
     3NOBS1(12),JBS1(12),LBQ1(12),LBQQ1(12),OLT1,NOR(50,12),OLTLT(50,12)
     4,JQSLT(50,12),JASLT(50,12),NOSLT(50,12),NOBSLT(50,12),JBSLT(50,12)
     5,JQSLT1(50,12),JADLT1(50,12),NOSLT1(50,12),NOBLT1(50,12)
      COMMCN JBSLT1(50,12),LBR,LBQR,NOS2(12),JQS2(12),JAD2(12),NOL2(12),
     1JQOL2(12),NOBS2(12),JBS2(12),LBQ2(12),LBQQ2(12),JOR(12),JSO(12),
     2JM,ALPHA,CR(12),C(12),CC(12),CS(12)
      DIMENSION NSET(6,1)
      DIMENSION TCOST(12),COOR(12),CCOST(12),CQOL(12)
   55 FORMAT(1H-,5F10.2,I8)
      DO 56 I=1,3
      COOR(I)=JOR(I)*CR(I)
      CCOST(I)=(SUMA((3+I),1)/SUMA((3+I),3))*C(I)*CC(I)
      CQOL(I)=CS(I)*JQOL(I)
      TCOST(I)=COOR(I)+CCOST(I)+CQOL(I)
   56 PRINT 55,ALPHA,COOR(I),CCOST(I),CQOL(I),TCOST(I),I
      DO 82 I=1,3
   82 PRINT 81,JQS(I),NOS(I),JQOL(I),NOL(I)
   81 FORMAT(1H-,20X,4I10)
      COST3=TCOST(1)+TCOST(2)+TCOST(3)
      PRINT 91,COST3
   91 FORMAT(1H-,'TOTAL COST FOR THREE ITEMS=',F10.2)
      RETURN
      END
```

APPENDIX B

65

PEERLESS
CLASP
FEDERAL ENVELOPE CO.
No. 63—6½ x 9½

THE
PRECEDING
ENVELOPE
WAS EMPTY
WHEN
RECEIVED
FROM THE
CUSTOMER

A GASP SIMULATION STUDY OF A MULTI-PRODUCT INVENTORY SYSTEM

by

SAMBAMURTY ARISETTY

B. E. (Mechanical), Andhra University, India, 1966

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1969

The objectives of this report are to use the GASP II simulation language to simulate a multi-product inventory system, the products having different demand rates and to compare two alternative inventory management systems, the (s,S) system and the one based on adaptive forecasting using an exponential smoothing technique.

The simulations were performed assuming an exponential arrival rate, with a normally distributed demand. A delay has been incorporated between the warehouse and the posting department. The posting delay is assumed to follow a normal distribution with known parameters. The order lead time is also assumed to follow a known normal distribution.

The inventory system was simulated for two years with the two inventory management systems. In exponential smoothing, different values of smoothing constant were tried to select the best. Also different review periods were tried in the (s,S) system. The effect of the two inventory management systems on the total cost was found to be insignificant. However, for the values of smoothing constant tested, the value 0.25 was found to have lower total cost. When using the (s,S) system the review period of 40 days was found to be better.

In the simulation high variability was observed in the total cost when the simulation runs were performed with different initial random numbers. Investigation of the high variability showed that the transient conditions which exist at the initial stages of the simulation contributed to much of the high variability. Simulation results in the form of monthly and lead time reports were prepared for management.

Further research should be directed on eliminating variability in the experimental errors.