INVENTORY CONTROL IN A MULTI-WAREHOUSE SYSTEM

BY

JOHN CLINTON

A Thesis presented for the degree of

Doctor of Philosophy

in the

University of London

September, 1968

ABSTRACT

The problem of inventory control in a stores complex consisting of a central store supplying a number of sub-stores upon which demands are made has received comparatively little attention in the literature.

This thesis looks into the problems of control from an overall point of view and proposes heuristic rules drawing from the experience of an author in this field. These rules are tested and refined on simple yet fairly representative models of a real-life complex and comparison of performance under the proposed control rules is made with the control proposed by J.A. Cran, who recently has developed ideas which he has shown to improve considerably on traditional methods.

It is shown that the rules proposed here are not only likely to lead to lower costs than the Cran control but are flexible enough to respond well to sudden changes in demand at any sub-store. The advantage of an anlytical procedure to obtain the instant the complex should order further stock benefits over the Cran control in which reorder level for the complex does not respond to change in demand rate at any sub-store.

ACKNOWLEDGMENTS

The author wishes to record here his thanks to Mr. William Lampkin and to Professor Samuel Eilon, under whose joint supervision in the Management Engineering Section of The Mechanical Engineering Department at Imperial College, London, this research was undertaken.

John Clinton,
Imperial College,
London University, S.W.7.

INDEX

CHAPTER NO.	TITLE	PAGE NO.
1	INTRODUCTION	5
2	LITERATURE REVIEW: DYNAMIC PROGRAMMING APPROACH	10
3	REVIEW OF THE LITERATURE PERTINENT TO THE CENTRAL STORE/SUB-STORE SYSTEM: CASE OF THE STEADY STATE APPROACH	24
4	ASSUMPTIONS OF THE INITIAL MODEL OF THE COMPLEX	94
5	FIRST ALTERNATIVE METHOD OF CONTROL	102
6	THE COMPARATIVE SIMULATIONS	118
7	COMPARISON OF CRAN'S CONTROL AND AUTHOR'S CONTROL WHEN MODEL DATA FAVOUR CRAN	139
8	EXTENSIONS TO NON-ZERO SUB-STORE LEAD TIME	145
9	INTRODUCTION TO FIRST DYNAMIC PROGRAMMING MODEL	156
10	AN ALTERNATIVE TO 'SHARE' RATION RULE	172
11	REORDER LEVEL OF SUB-STORES WHEN A PROCUREMENT IS NOT OUTSTANDING	177
12	TESTING THE IDEAS OF CHAPTERS 10 AND 11	184
13	A NEW RATIONING RULE "SHARE MK IV"	189
14	EXTENSION OF THE NEW RATIONING RULE	200
15	INTRODUCTION OF NEW MODELS TO TEST SUGGESTED CONTROL OVER A MORE GENERAL FIELD	208
16	CASE OF CONTROL WHEN DEMANDS COME FROM AN UNKNOWN DISTRIBUTION AND ARE SIGNIFICANTLY DIFFERENT FOR SUB-STORES	231
17	AN ANALYTICAL APPROACH TO OBTAINING THE TRIGGER FOR PROCUREMENT ORDER AND COMPARISON OF PROPOSED CONTROL IDEAS WITH THE CONTROL OF CRAN	265
18	SUMMARY	279
APPENDICES		284
EXHIBITS		327
PROGRAMS		345

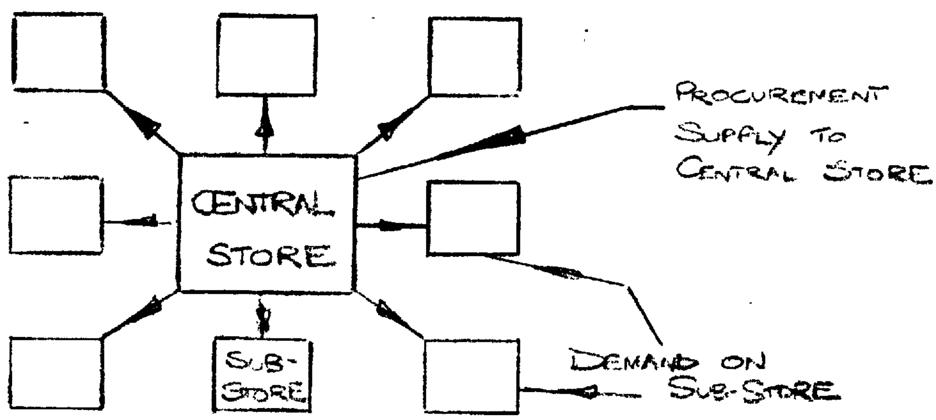
CHAPTER ONE

INTRODUCTION

This chapter discusses where the central store/sub-store complex is encountered in business and the types of policy decisions which have to be made in controlling it.

1.1 Introduction to the Central Store/Sub-store System

The stores system or complex under consideration consists of a central store and a number of sub-stores, the latter being dependent on their supply from the former, and where demands on the sub-stores are stochastic. The situation is depicted in the diagram below.



There are many real situations in business where such a complex occurs. For example, the 'central store' may be a warehouse at a convenient location supplying a very large number of stores retailing to the public. In this case, the retail stores are the 'sub-stores' of the complex. Alternatively, the 'central store' may be a warehouse attached to the factory's production unit supplying to district warehouses (the 'sub-stores') over the country, on which requisitions from retailers are made. It should be noted here that if these retailers are part of the same company operating the central store and the district warehouses, then the problem becomes more complicated in that the demands on the sub-stores from the retailers are not completely random, since control can, and is, exercised over them. However, the complex depicted in the diagram still has application in this case because the retailing store may be considered as the 'sub-store' and the district warehouse as the 'central store'.

1.2 Introduction to the Problems of Control

The basic control problem is to match supply and demand at the sub-stores in the most economical way. An excess of supply over demand is costly because of the inventory investment, and an excess of demand over supply is costly either because of the loss of a sale or (where backorders are backlogged) because of the extra cost incurred in fulfilling back-orders. In order to circumvent the problem, stock is usually held at some or all nodes of the complex, i.e., at the sub-stores and/or the central store.

Some of the basic problems will be introduced here so the reader can gain some insight into the overall difficulties.

When stock ordered for the system arrives at the central store, a decision has to be made as to whether or not it should be immediately distributed out to the sub-stores, and if so, whether all or only a fraction of it should be distributed. Also the allocation decision for each sub-store has to be made. In searching for a solution to this problem, it will be seen that a distribution to the sub-stores of all the available stock in the central store will be unwise since in the event of a new order for the system not being available at the sub-stores for a long period, a sub-store low in stock will be in either of two situations

(i) It must suffer a stock-out;

or (ii) It can receive a redistribution from some other sub-store.

Generally either of these alternatives are excessively costly, and it pays to retain stock in the central store to be available for replenishing needy sub-stores. It may be seen that the problem can in many instances be overcome by keeping a high level of average stock in the complex (i.e. by following on one stock order by another in less time than otherwise) so that sub-stores are replenished when their stocks are generally fairly high, but not only is this likely to be costly in terms of holding inventory costs, if the variance of sub-store demand is high, shortages may still occur.

When deciding on the quantity to be issued to a sub-store due to be replenished, it is clear that this issue quantity must relate to the inventory positions of all the other stores in the complex. Clearly when there is liberal stock in the complex, the sensible procedure is to try to aim that the quantity allocated will suffice the sub-store until stock from the next system delivery can get to this sub-store. In this way we attempt to minimise the overall cost of deliveries. On the other hand, when an issue is triggered by the sub-store reaching a reorder level whilst the complex is low in stock (the latter point may be made more specific by relating it to those instances when a stock order for the complex is outstanding) the more important problem now is to try to ensure that the issue is not so high that shortages result in the other sub-stores. Thus a policy of 'rationing' should exist at such times. The problem may be seen to resolve into balancing the expected shortages of all sub-stores and the costs of replenishment.

Further, it is clear that sub-store reorder levels should decrease as the time that the order for the complex is due to arrive gets nearer. The reason for this is that it may well be cheaper to wait for a large delivery rather than trigger a small one.

The order for the system should itself be triggered by some function which encompasses the inventory position of the whole complex. Central store stock may be seen not to be a sensible trigger; it is costly in inventory holding to trigger an order when central store stock is low when sub-stores have plenty of stock. The total inventory existing in the complex is better but not really satisfactory as a trigger because this would make a system with uneven distributions of stock over the sub-stores appear falsely safe.

The variables which will be determined by the control method are as follows:-

(i) number of replenishments to sub-stores

(ii) number of shortages and time of shortages at sub-stores

(iii) number of stock orders for the complex

(iv) inventory holding costs related to the average level of inventory maintained in the complex.

The ~~variables~~ parameters which essentially define the problem are:

(i) lead time for replenishments to sub-stores

(ii) lead time for the stock order to the complex

(iii) number of sub-stores

(iv) nature of demand at each sub-store

(v) cost of stock holding

(vi) penalties for shortages

(vii) cost of item itself

(viii) cost of replenishing sub-stores

The controlling decisions, of which some may be established heuristically, are as follows:

(i) when to trigger the order for the complex

(ii) quantity of stock ordered for the complex

(iii) amounts delivered to sub-stores

(iv) when to replenish sub-stores.

1.3 Chapter Summary and Introduction to Chapter Two

It is hoped that the reader will have some idea of the problems this thesis attempts to solve. In the relevant literature on this subject two distinct approaches are made. The first, the dynamic programming approach, considered by Clark[8,9,10] and his colleagues, is presented in Chapter Two.

CHAPTER TWO

REVIEW OF THE LITERATURE PERTINENT TO THE CENTRAL STORE/SUB-STORE PROBLEM:-

CASE OF THE DYNAMIC PROGRAMMING APPROACH

2.1 The Work of Clark and Scarf

2.1.1 Summary of work

This work develops a dynamic programming model to produce decision rules for a stores complex consisting of a central store, a factory for supply of a procurement, a repair facility, and a number of sub-stores.

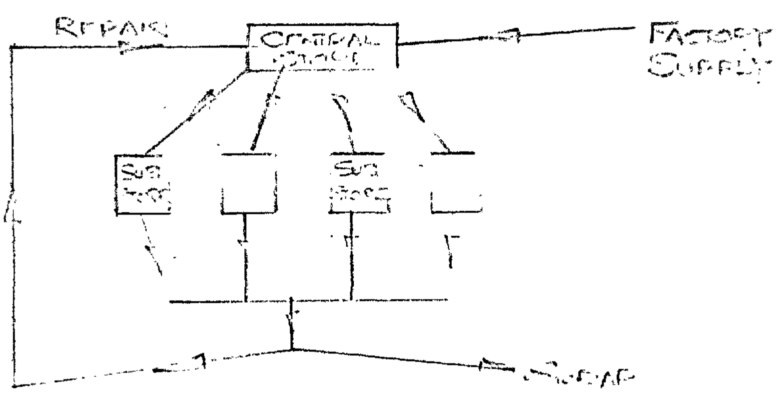
Demands occur at the sub-stores and are assumed stochastic and independent from sub-store to sub-store and from one period to another. Demands can be time dependent and distributions may be different between sub-stores. Stock review takes place at discrete instances in time, and lead times for replenishment and procurement and repair lead time are all taken to be fixed integers times the interval between stock reviews.

Costs of stockholding for any period are taken as proportional to the end-of-period stock at the particular sub-store but the unit cost may differ between sub-stores. Shortages cost a fixed amount times the total number of shortages experienced, and the cost of any shipment in stock is considered proportional to the quantity shipped. There is a restriction on movement of stock - only downwards in the stores hierarchy (not from sub-store to sub-store). Clark only considers costs up to a finite time horizon.

The technique of solution is to set up a sequence of one-dimensional dynamic programming equations. In these the expected discounted future cost for period n for an echelon K is a function of the total stock of echelon K and depends on the function for period (n+1), echelon K and the function for period n, echelon (K-1). This representation of costs is admitted to be approximate though it is claimed to produce good results.

2.1.2 Introduction

In 1958, A.J. Clark published his report[8] in which a dynamic programming model was proposed for the solution of the control problems of the complex depicted below.



The arguments employed by Clark in establishing his model were heuristic, and no claim of optimality was made. In 1959, Clark and Scarf together showed[9] that Clark's model was optimal for the case of an unbranching series of stores, and that it was probably good for the central store/sub-store problem. Later still, in 1961, Clark and Scarf went on to show[10] how one of their more restrictive assumptions could be relaxed.

2.1.3 The single echelon model

The case of a single store is introduced first. Stock is considered at discrete review instants. The criterion is the minimization of discounted costs up to a finite planning horizon. The discount rate is w. Holding costs are assumed proportional to the stock after each decision is made.

A vector $\underline{H}_j$ is defined thus:

$$(\underline{H}_j)_x = \begin{cases} h_j x & \text{for } x > 0 \\ 0 & \text{for } x \leqslant 0 \end{cases}$$

$(H_j)_x$ is the holding cost associated with a virtual stock x at time j; h_j is the unit holding cost at time j. Depletion costs are assumed proportional to the number of shortages experienced, and thus a depletion vector D_j is similarly defined:

$$(D_j)_x = \begin{cases} 0 & \text{for } x \geqslant 0 \\ -d_j x & \text{for } x < 0 \end{cases}$$

Demands are assumed captive.

A demand matrix P_j is defined thus:

$$(P)_{xy} = \begin{cases} g_j(x-y) & x \geqslant y \\ 0 & x < y \end{cases}$$

where $g_j(i)$ is the probability of a demand of i units between time j and (j+1).

Then, if $\underline{C}_j$ denotes the cost of the best policy from time j onwards, and assuming zero lead time, the following equation is given:

$$\underline{C}_j = \underline{f}(\underline{H}_j + P_j\underline{D}_j + w\, P_j\underline{C}_{j+1}) \quad , \qquad (1)$$

$\underline{f}$ being the ordering function or ordering policy.

We put $\underline{T}_j = \underline{H}_j + P_j\underline{D}_j + w\, P_j\underline{C}_{j+1}$, and so:-

$\underline{T}_j$ is a vector whose x^{th} element is the expected future cost if the stock at time j is x, and if no ordering action is taken at time j, and if an optimum policy is thereafter followed.

It has been shown[11] that if the ordering cost is proportional to the quantity ordered then the least cost ordering policy, $\underline{f}$, is as follows:-

$$\underline{f}(t_j,x) = \begin{cases} t_{j,x} & \text{for } x \geqslant S_j \\ t_{j,S_j} + v_j(S_j - x) & \text{for } x < S_j \end{cases} \tag{2}$$

where
$$S_j = \underset{j}{\text{Max}} \left(x : (t_{j,x} - t_{j,x-1}) \geqslant v_j\right) \tag{3}$$

and v_j is the unit ordering cost at time j.

Thus the ordering policy is:

Do not order if $x \geqslant S_j$

Order $(S_j - x)$ if $x < S_j$

If the ordering cost for q units is $C_{R_j} + qv_j$, then

$$\underline{f}(t_j,x) = \begin{cases} t_{j,x} & \text{for } x > s_j \\ t_{j,S_j} + v_j(S_j - x) + C_{R_j} & \text{for } x \leqslant s_j \end{cases} \tag{4}$$

where $s_j = \text{Max } \{x: \left(t_{j,x} - t_{j,S_j} - v_j(S_j - x)\right) \geqslant C_{R_j}\}$ (5)

Equation (1) leads to $\underline{C}_j = \underline{H}_j + \underline{P}_j \underline{D}_j + \underline{f}(wP_j \underline{C}_{j+1})$ (6)

2.1.4 The Multi-echelon problem

The model proposed here considers the complex depicted in 2.1.2 although it is said to have application to many different stores complexes.

Suppose the time horizon to be n intervals. Let t_P, t_R, be the number of intervals in the production lead time and repair lead time respectively. Then, for this complex,

$$t_P > t_R \geqslant 1$$

2.1.4.1 Echelon definition

An echelon is defined as a store and all stores fed directly or indirectly from it.

If S_s = Total Sub-store Stock,

S_c = Central Store Stock,

q_{Ra} = Repair stock with "a" intervals before availability at the central store,

q_{Pb} = Production stock with "b" intervals before availability at the central store,

we have:

$$t_R \geqslant a \geqslant 1$$

$$t_P > b \geqslant 1$$

Echelon "K" is defined as containing E_K as follows:-

$$E_{-1} = S_s$$

$$E_o = S_s + S_c$$

$$E_K = S_s + S_c + \sum_{a=1}^{K} q_{Ra} + \sum_{b=1}^{K} q_{Pb}; \quad 1 \leqslant K \leqslant t_R \qquad (7)$$

$$E_K = S_s + S_c + \sum_{a=1}^{t_R} q_{Ra} + \sum_{b=1}^{K} q_{Pb}; \quad t_R < K < t_P$$

$$E_{t_P} = \infty$$

2.1.4.2 Assumptions

(a) For Echelon -1

It is assumed that the costs of supply to a sub-store are identical, whether the supply be from a sub-store or from the central store.

(b) For Echelon 0

It is assumed that the supply costs are identical, whether from production or repair.

(c) There is no fixed ordering cost for any echelon except the highest one.

2.1.4.3 Heuristic formulation of the model

Consider the K^{th} echelon. It will be clear from the definition that the stock present in this echelon is contained in all echelons of higher order.

The holding cost charged to the K^{th} echelon is thus not the complete cost of holding its stock but just the increment in holding cost as a result of having stock at echelon K rather than at echelon (K+1). The summation of such stock-holding costs then is correct for the whole complex. Similarly, the unit shortage cost charged to the K^{th} echelon is the difference between the cost to the system of a demand not being met, until the $(K+1)^{th}$ echelon instead of the K^{th}.

There is, further, the extra cost involved if the stock at the K^{th} echelon is insufficient to supply the required stock at the $(K-1)^{th}$ echelon when the latter makes its order. The cost assigned to echelon K is the difference in the expected costs of working echelon (K-1) with the optimum stock level and with the stock available.

2.1.4.4 <u>The analysis for echelon -1</u>

Consider echelon -1 and suppose there are N sub-stores. $\underline{H}_{j,i}$ is the holding cost vector for period j at sub-store i. $\underline{D}_{j,i}$ is the corresponding depletion cost vector.

$C_{R_{j,i}}$ is the unit ordering cost

$g_{j,i}$ is the demand probability distribution

ℓ_i is the lead time from central store to sub-store - which is to be either zero or unity.

Then $S_{j,i}$, which is the maximum stock level for sub-store i, can be obtained by the use of one-dimensional dynamic programming on equation (6).

Now if all or part of the order of the sub-store cannot be satisfied, then the sub-store incurs more costs than assumed by the ordering policy. These extra costs, $\underline{B}_{j,i}$ are given by:-

$$\underline{B}_{j,i} = \begin{cases} \underline{t}_{j,i} - \underline{f}(\underline{t}_{j,i}) & \text{if } \ell_i = 0 \\ wP_{j,i}\underline{C}_{j+1,i} - \underline{f}(P_{j,i}\underline{C}_{j+1,i}) & \text{if } \ell_i = 1 \end{cases} \qquad (8)$$

This cost vector simply represents the expected cost of sub-store i being forced to operate below its optimum inventory level, i.e. if x is the notional stock of sub-store i plus the quantity at the central store free for allocation to sub-store i at the time j, then the penalty charged to echelon 0 as an extra depletion cost is $\{\underline{B}_{j,i}\}_x$. If there exist $\sigma_j = \sum_{i=1}^{N} S_{j,i}$ serviceable items in echelon 0 at time j, then no such penalty is to be charged to echelon 0.

It is to be noted that the extra depletion cost charged to echelon 0 will be a total over all sub-stores. It is important to note that Clark is only able to obtain a unique answer on the assumption of optimal deployment of stock among the sub-stores (even if this means redistributing stock amongst them).

2.1.4.5 Clark's treatment of the problem of optimal deployment of stock amongst sub-stores

A vector $\underline{\Delta}_{j,-1}$ is given such that the elements of the vector, which are denoted by $\delta_{j,-1,x}$ are given by

$$\delta_{j,-1,x} = \begin{cases} 0 & \text{for } x \geq \sigma_j \\ b_{j,i,y} & \text{for } x < \sigma_j \end{cases} \tag{9}$$

where $b_{j,i,y}$ is the y^{th} element of $\underline{B}_{j,i}$ and i,j are chosen so that

$$\delta_{j,-1,x-1} \geq \delta_{j,-1,x} > 0 \qquad \text{for } x < \sigma_j$$

The elements of $\underline{\Delta}_{j,-1}$ increase in value as the amount of serviceable stock, x, decreases. For stock positions below σ_j, these constitute the depletion cost chargeable to echelon 0.

2.1.4.6 The analysis for higher echelons

It is assumed that the following are available.

$\underline{H}_{j,k}$ = the holding cost vector at echelon K.

$\underline{D}_{j,k}$ = the depletion cost vector.

$v_{j,k}$ = the unit ordering cost

C_{P_j} = the fixed production set-up cost.
(applies to echelon (t_p-1) only).

$G_{j,k}$ = the demand probability matrix.

It is assumed that the holding cost is proportional to the cash value of the stock balance. The additional cost of holding x units of inventory at echelon K rather than at echelon (K+1) is

$$h_{j,k,x} = \begin{cases} w_k v_{j,k} x & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

where w_k is the unit holding cost per unit of increased value, and $v_{j,k}$ is the value added to the item by ordering it into echelon K.

$G_{j,k}(x)$ represents demands (or "losses" in the context of Clark's work) to the echelon during the period j. For $0 \leq K \leq t_R$ it includes all failures, and for $t_R \leq K \leq t_P$, it includes just the condemned failures.

Each echelon is treated as a single echelon problem using the following dynamic programming recurrence relations:

$$\underline{C}_{j,k} = \underline{H}_{j,k} + \underline{\Delta}_{j,k-1} + P_{j,k}\,\underline{D}_{j,k} + \underline{f}(wP_{j,k}\underline{C}_{j+1,k}) \tag{10}$$

where $0 \leq K < t_P$ and $\underline{f}$ is defined just as in the single echelon problem (equation (2) or (4)).

For $0 \leq K \leq t_P - 1$

$$\underline{\Delta}_{j,k} = wP_{j,k}\underline{C}_{j+1,k} - \underline{f}(wP_{j,k}\underline{C}_{j+1,k}) \tag{11}$$

From these recurrence relations, $\underline{C}_{j,k}$ and the echelon ordering parameters $S_{j,k}$ for $0 \leq j < n$, $0 \leq K < t_P$, can be obtained.

2.1.5 The central store/sub-store problem

With the assumptions of the Clark model holding, the sub-store stocks together comprise the lower echelon, and the higher echelon comprises all the stock in the complex (i.e. sub-store stocks plus central store stock). It is necessary to the model to obtain a unique penalty function (which depends only on the stock of the higher echelon), to penalise the higher echelon for inability to supply the lower to optimal levels. To do this, Clark was compelled to make the assumption that inventory was easily exchangeable amongst sub-stores, and the value of the extension of the Clark model to this problem is limited by the extent to which this assumption is invalid.

2.1.6 Comments on the work of Clark and Scarf

The work of Clark and his echelon concepts is a major contribution to the literature for inventory control in a hierarchy of stores. Lampkin, in his thesis,[3] specifically disagrees with Clark on his construction of the vector $\underline{\Delta}_{j,-1}$ in equation (9). This author agrees with Lampkin in noting that the costs defining $\delta_{j,-1,x}$ should contain the costs for not allocating the optimal stock from the quantity available summed for all sub-stores (not just for one sub-store).

In considering the application of Clark's model to the central store/sub-store problem it has been stated that Clark was forced to assume easy exchangeability of stock between sub-stores to get over the problem of assigning a unique penalty for a higher echelon stock less than σ_j. However, in doing this, the future stocks at any sub-store become dependent on the stock level at all other sub-stores, since any of them might be short of stock and require shipments to them. The lower echelon model is thus incorrect, since it treats each sub-store separately and independently of other sub-stores. The usefulness of Clark's model applied to the central store/sub-store problem is thus limited by the extent to which transshipments would be a good idea in practice.

In many instances in practice, inter sub-store distribution will often not occur, and so the model due to Clark will be nearly correct and should produce good results. There is the ~~downfall~~ difficulty that this dynamic programming model may not be acceptable to some managements for running a central store/sub-store complex because of the considerable computation necessary.

2.2 The Work of A. Gradwohl

2.2.1 Introduction

In 1959, Gradwohl published a report[16] in which the nature of solutions to the multi-echelon inventory problem was investigated through case studies. This particular work is a straightforward exposition of the paper due to Clark and Scarf[9], in which the latter authors considered the mathematical aspects of the formulation of the multi-echelon inventory problem and established conditions under which the policies determined were truly least-cost (viz. in the case of an unbranching series of stores).

After Clark and Scarf introduced a modification to their first formulation[10] relaxing one of their more restrictive assumptions (in which fixed order costs at lower echelons were allowed yielding policies approximating the least-cost policies for the case of zero fixed order cost at lower echelons), Gradwohl (also in 1961) published a report[24] dealing with several aspects of this formulation.

* This latter report describes the technique for computing the approximate** policies in non-mathematical terms. The sensitivity of these policies (and their costs) to parameter changes is then investigated. Finally, the effect of the implementation of these approximate policies in a central store/sub-store complex is studied in an attempt to determine the cost increase over the theoretically optimum.

2.2.2 Technique for computing approximate policies

2.2.2.1 A simple example

A simple example demonstrates the method. The case of a sub-store ordering from a central store is considered. It is now assumed the sub-store incurs zero fixed ordering cost.

A purely arbitrary cost curve is considered. If the cost associated with a level s at the sub-store exceeds that with one unit less by more than the cost P of a unit, then at least this level of inventory should be maintained. Thus it is required to find

$\left(\text{Max } s : C(s) - C(s-1) > v\right)$, where v is the per unit ordering cost.

Thus if we denote this level of s by $\hat{s}$, if it is found that the stock level is less than $\hat{s}$, stock is ordered up to $\hat{s}$.

---

* In this review, the present author feels an indication of the type of consideration in hand by general terms (rather than the particular case given by Gradwohl) to be more appropriate.

** Approximations are used becausetrue least-cost solutions require computation which cannot economically be allowed.

If, however, an order for this stock difference cannot be met by the central store, the sub-store will be expected to experience greater costs than otherwise. If it is at a level $s_o (s_o < \hat{s})$ then its expected cost is $C(s_o)$. If the order had been filled, the cost would be $C(\hat{s}) + (\hat{s} - s_o)v$. It is suggested that a cost of $\{C(\hat{s})+(\hat{s}-s_o)v-C(s_o)\}$ be charged to the supply store because it has failed to supply. This is termed the 'implied shortage cost'. This charge occurs at any time the total number of units at sub-store and central store is less than $\hat{s}$.

2.2.2.2 Case of non-zero fixed order cost at sub-store (c_R)

The analysis here is analogous to that above with the exception that we now have an (s,S) policy where S equals $\hat{s}$ and s is

$$s_{ROL} = \left(\text{Max } s: \ C(s) - \{C(\hat{s} + (\hat{s}-s)v + c_R\} > 0\right) .$$

The implied shortage cost must now be computed. If total stock at central store and sub-store S_T is such that $S_T \leqslant s_{ROL}$, then the implied shortage cost would be $C(S_T) - \{C(\hat{s}) + (\hat{s}-S_T)v + c_R\}$. This cost will be incurred because the sub-store is sure to be at its reorder point.

2.2.2.3 Case where S_T exceeds s_{ROL}

The implied shortage cost is more difficult to compute for this case because it depends on the distribution of stock between central store and sub-store. It is argued that the implied shortage cost should be charged if the sub-store is at or below s_{ROL}, its reorder point, and not if it is above reorder point. This distribution of stock can be taken into account but means excessively long computation, and so to decide whether or not to charge an implied shortage cost to the central store, the following procedure is adopted.

Whenever S_T exceeds s_{ROL}, sub-store stock s is considered to always be described by:

either (i) $s \leqslant s_{ROL}$

or (ii) $s > s_{ROL}$.

Case (i) results in a "maximum implied shortage cost" and (ii) a "minimum implied shortage cost".

2.2.2.4 Control at the central store

It is assumed that control at the central store is an (s,S) policy. For this store, s and S are computed twice with independent computations. The first considers the maximum implied shortage costs added to the central store future costs and then the minimum implied shortage costs added. From each new cost curve the (s,S) values are obtained. Neither (s,S) policy, of course, represents the optimal policy because neither the implied shortage costs of Cases (i) and (ii) are correct.

The unknown optimum cost is shown to lie somewhere between the two possibilities, and the feasibility of using the central store "maximum policy" (Case (i)) with the assurance that the increase in costs over the optimum policy does not exceed that over the "minimum policy" (Case (ii)) is argued. It is shown that similar assurance is not possible if the "minimum policy" is used.

It is clear then that the central store "maximum policy" must be used since the margin by which the cost of this policy exceeds that of the least-cost policy can be calculated.

2.2.2.5 Exceptions to use of a "Maximum Policy" at the central store

It is stated that for very cheap items where it is economical to order a supply to last for a long time, the "minimum policy" is the more correct policy.

2.2.3 Sensitivity of Policies

The sensitivity of policies and costs to parameter values is investigated to identify regions in the parameter space where the maximum and minimum central store policies lead to expected costs which are not significantly different. In these regions both these policies approximate the unknown optimum policy.

2.2.3.1 Method of investigation

A nominal case with nominal parameter values is fixed and each are varied systematically about this nominal one at a time. The parameters considered are

(i) Mean demand per review period
(ii) Sub-store fixed order cost
(iii) Central store fixed order cost
(iv) Sub-store shortage cost
(v) Central store shortage cost
(vi) Replenishment cost
(vii) Number of sub-stores
(viii) Distribution of demand among sub-stores
(ix) Lead time to central store

2.2.3.2 Result of investigation

The following parameter values lead to good approximations to the optimum policy:

(a) High Mean Demand per period

(b) Low Sub-store Fixed Order Cost

(c) High Central Store Fixed Order Costs

(d) High Replenishment Cost

(e) Few Sub-stores

(f) Imbalance of Demand among Sub-stores

(g) Long Central Store Lead Time

2.2.4 Application of Policies

The evaluation of the practicality of the application of the suggested approximate policies is made by computing both the costs of controlling the complex with the central store maximum and minimum levels and comparing. In this way (with the assumption that the central store maximum levels are used for the entire stock) the cost increase over the optimum is estimated (assuming optimum cost and central store minimum level cost are equal). Then since the cost increase over optimum attributable to traditional control policies is roughly known through experience, savings attributable to the approximate policies may be estimated. A simple hypothetical multi-echelon complex (a central store/sub-store complex was used) was chosen for this evaluation. It was found that the cost associated with the maximum policies was 2.5% greater than with the minimum policy. Thus use of the maximum policy results in a maximum increase of 2.5% over the optimum cost.

Notably, the factor keeping this difference so low is the high-value nature of the stock. Gradwohl goes on to suggest traditional inventory policy costs range from 20-30% above optimal, and can be expected almost always to exceed 2.5%. It is concluded that implementation of the approximate policies is likely to result in significant savings.

2.2.5 Comments on the work of Gradwohl

This work is valuable in that it shows up (in very general terms) the regions in which the control model due to Clark and Scarf will produce fairly substantial savings over traditional controls. The same criticisms apply as in the review of the work of Clark (2.1); the ideas established are purely heuristic and (s,S) controls at sub-stores and central store may be improved upon.

Gradwohl has indicated that for very cheap items a "minimum policy" at central store is the preferred policy. There are likely to be other regions over the parameter space where this policy is to be preferred to a "maximum policy". It may be a good idea if these could be explicitly defined.

The results of the investigation into the sensitivity of policies over the parameter space again are not very explicit, and for the converse of those parameter values cited in 2.2.3.2, Gradwohl states that the "maximum policy" poorly approximates the optimum policy. This thus tends to restrict the usefulness of the ideas in Clark's model when applied to practical cases where these paramters occur.

As Gradwohl himself suggests, an "approximation" to the approximate policies to allow direct computation of policies for non-dynamic cases is a suitable area for further research, in order that the lengthy iterations of the suggested dynamic programming procedure may be eliminated for these cases.

2.3 Chapter Summary and Introduction to Chapter Three

The very general model due to Clark is applicable in the case of a policy decision for stock to be reviewed periodically and simultaneously at all stores. It is important to note that it will, necessarily, entail considerable computation for each item considered, and can really only sensibly be used for expensive and important items.

Most of the literature appertaining to controlling the central store/ sub-store complex under consideration takes a steady state approach and these works are dealt with in Chapter Three.

CHAPTER THREE

REVIEW OF THE LITERATURE PERTINENT TO THE CENTRAL STORE/SUB-STORE SYSTEM:-

CASE OF THE STEADY STATE APPROACH

3.1 The Work of F. Hanssmann

3.1.1 Introduction

The work of F. Hanssmann as described here is found in a paper to 'Operations Research' and in his book[2]. This work considers the problem of inventories of partially finished goods but this is exactly analogous to the central store/sub-store problem, and the ideas are written here in the language of the latter.

Hanssmann assumes that when shortages occur, no direct cost is involved, but that demand is related to the average waiting time of customers, and this relation is assumed known, being estimated empirically. It is assumed that demands are captive.

Stock is reviewed simultaneously at all sub-stores after equal time intervals. Demands on the sub-stores are continuous, and demands in separate weeks are independently normally distributed. Stock is ordered each week from the sub-stores, and also from the central store, the former orders being independent of each other, the latter order assuming full knowledge of the sub-store orders.

As with a few authors in this field, Hanssmann develops his ideas by considering firstly a single store, secondly two stores in series, and then thirdly the central store/sub-store problem. For comprehensiveness, and convenience for the reader, these ideas will be taken in order.

Initially two functions are introduced:

$$A_n(I_n) = \frac{1}{\sqrt{2\pi}} \int_0^{\infty} I \exp\{-(\frac{I-I_n}{2})^2\}\, dI \qquad (1)$$

$$B_n(I_n) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{0} I \exp\{-(\frac{I-I_n}{2})^2\}\, dI \qquad (2)$$

where $I_n = I_0/\sigma$ and where I_0, σ are, respectively, the mean inventory level and the standard deviation of sub-store demand.

These are, respectively, the normalised overage and normalised shortage.

3.1.2 The single store

Let V be the starting stock, the lead time ℓ weeks, d_j is the demand in the j^{th} week, and q_j is the amount ordered by the store in week $(j-\ell)$, i.e. q_j arrives at week j. Then the virtual inventory level at week i is I_i where:-

$$I_i = V + \sum_{j=1}^{i} (q_j - d_j) \qquad (3)$$

Action by the store at time zero affects the stock level at time ℓ, but not before. Thus q_ℓ (the stock ordered at time zero for arrival in week ℓ) is chosen so that at time ℓ, the store holds a target level of inventory I_o.

Thus we have: $\varepsilon(I_\ell) = I_o$ *

Using (3) then, $I_o = V + \varepsilon\{\sum_{j=1}^{\ell} (q_j - d_j)\}$

$$\text{i.e.} \quad I_o = V + \sum_{j=1}^{\ell} \{q_j - \varepsilon(d_j)\} \qquad (4)$$

From (4) q_ℓ may be found:

$$I_o = V + \sum_{j=1}^{\ell-1} q_j + q_\ell - \sum_{j=1}^{\ell} \varepsilon(d_j)$$

$$\text{and} \quad q_\ell = I_o + \sum_{j=1}^{\ell} \varepsilon(d_j) - \{V + \sum_{j=1}^{\ell-1} q_j\} \qquad (5)$$

Now (3) shows that $I_\ell = V + \sum_{j=1}^{\ell} (q_j - d_j)$

$$\text{and so, using (4)} \quad I_\ell = I_o + \sum_{j=1}^{\ell}\{\varepsilon(d_j) - d_j\} \qquad (6)$$

If d_j is $N(\mu,\sigma^2)$, then I_ℓ is $N(I_o, \ell\sigma^2)$.

3.1.2.1 A simpler ordering rule than (5)

From equation (5) we obtain the stock to be ordered at week i, viz. $q_{i+\ell}$:

$$q_{i+\ell} = I_o + \sum_{j=i+1}^{i+\ell} \varepsilon(d_j) - \{I_i + \sum_{j=i+1}^{i+\ell-1} q_j\}$$

$$\text{i.e.} \quad q_{i+\ell} = I_o + \ell\mu - I_i - \sum_{j=i+1}^{i+\ell-1} q_j \qquad (7)$$

$$\text{Also,} \quad q_{i+\ell-1} = I_o + \ell\mu - I_{i-1} - \sum_{j=i}^{i+\ell-2} q_j \qquad (8)$$

by replacing i by (i-1) in equation (7) and so:-

$$q_{i+\ell} - q_{i+\ell-1} = I_{i-1} - I_i + q_i - q_{i+\ell-1}$$

* ε indicates "Expected Value of"

and, $$q_{i+\ell} = I_{i-1} - I_i + q_i \tag{9}$$

But from (3)

$$I_i = I_{i-1} + q_i - d_i$$

and so (9) yields:-

$$q_{i+\ell} = d_i \tag{10}$$

Thus, the rule is, "originally order as indicated by (5), and subsequently order the quantity demanded in the previous week".

3.1.2.2 <u>Analysis to establish optimal I_o</u>

Hanssmann computes holding cost from inventory left at the store at the end of a week.

Mean stock left over at the end of a week, the "Mean Overage" is given by A:-

$$A = \frac{1}{\sigma\sqrt{2\pi\ell}} \int_o^\infty I \exp - (I-I_o)^2/2\ell\sigma^2.dI$$

$$= \sqrt{\ell}\ \sigma\ A_n \tag{11}$$

Similarly "mean shortage" $\overline{B}$ is $\sqrt{\ell}\ \sigma\ B_n$ (12)

If the mean potential sales rate is s, and the empirical function of decay is u(t) such that a real sales rate su(t) is experienced, where an average customer delay time t is maintained ($0 \leq u(t) < 1$) then the standard deviation of weekly demands s is assumed to be $\sigma u(t)$ and I_ℓ will thus be

$$N\{I_o,\ \ell(u(t)\sigma)^2\} \tag{13}$$

The average waiting time is defined as the total amount of waiting time divided by the total number of demands

i.e. $t = -\overline{B}/su(t) = -B_n\sqrt{\ell}\ \sigma/s$

and so: $B_n = -ts/\sqrt{\ell}\ \sigma.$

If p is the profit per unit (excluding consideration of holding or shortage):-

We have expected weekly profit as a function of (t) equalling P(t):-

$$P(t) = p\ s\ u(t) - h\ A\ u(t)$$

$$= p\ s\ u(t) - h\ \sqrt{\ell}\ \sigma u(t)\ A_n \tag{14}$$

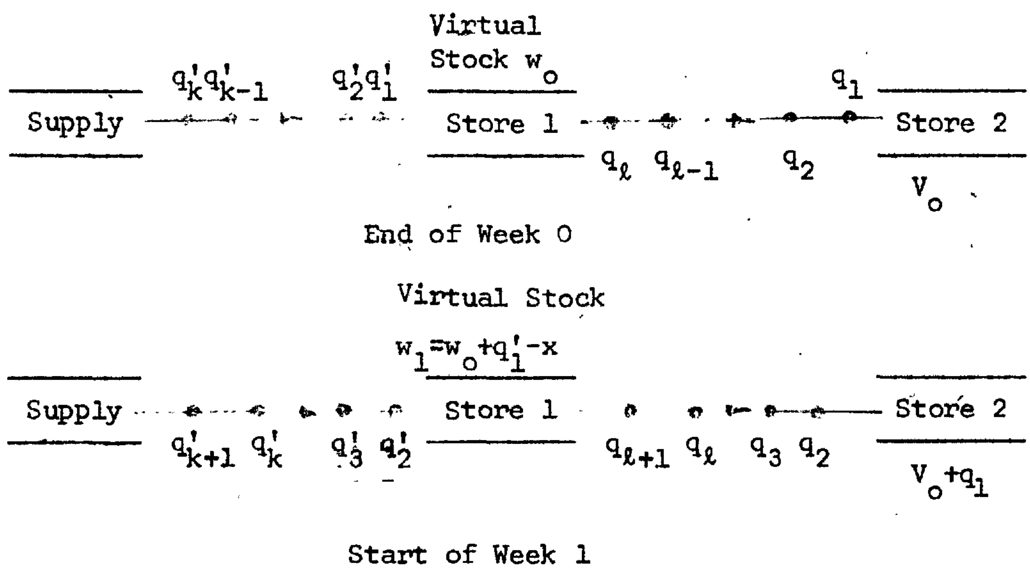
Now A_n is a monotonic function of I_n and so $A_n(B_n^{-1}(x))$ is a well-defined function. Hence (14) may be written thus:-

$$P(t) = psu(t) - h\sqrt{\ell}\sigma u(t)\, A_n(B_n^{-1}(-ts/\sqrt{\ell}\sigma))$$

The optimum value of t, $\hat{t}$, can be found numerically. Optimal I_o is given by

$$I_o = \ell\sigma u(\hat{t})\, B_n^{-1}(-\hat{t}s/\sqrt{\ell}\sigma).$$

3.1.3 Two stores in series



~~Figure B~~

Using the notation as in the previous section, we have in addition target inventory levels for stores 1 and 2, J_o, I_o respectively.

At the beginning of week 1, as shown in ~~Figure B~~ the above figure, $q_{\ell+1}$ is ordered by store 2 and q_{k+1}' by store 1 on the supply. The lead times to stores 2 and 1 are ℓ and k respectively. It should be noted that q_1' is available for issue in store 2 against the order for $q_{\ell+1}$ on it.

The ordering process is as follows:-

Given the quantities in stock and in the pipelines, a quantity x is computed which is the order from store 2 on store 1. Immediately x is decided upon, store 1 orders from the supply a quantity based on its new virtual stock (which will be $w_1 = w_o + q_1' - x$) and on the pipeline stock to it, viz. $q_2' \ldots q_{k+1}'$. The assumption is made that the supply always holds sufficient stock to meet orders in full.

Store 1 may experience shortage of course, and at the beginning of week 1, the quantity placed in the pipeline to store 2, viz. $q_{\ell+1}$

can differ from x.

If the virtual stock before the arrival of q'_1 was negative (i.e., store 1 was depleted) then we would like the shipment quantity to be $-w_o + x$, and, since the quantity available in the store is q'_1 we will have:

$$q_{\ell+1} = \text{Min } \{q'_1, x-w_o\} \qquad \text{for } w_o < 0$$

On the other hand, if $w_o \geq 0$, then no demands are outstanding on store 1, and there is a physical quantity of $q'_1 + w_o$ available with a desired amount x ordered; so

$$q_{\ell+1} = \text{Min } \{q'_1 + w_o, x\} \qquad \text{for } w_o \geq 0.$$

3.1.3.1 The effect of a request for x on future inventory levels at store 2

The week in which the effect is felt depends on the shortage at store 1. This shortage is a consequence of the target J_o and can be characterised by the average time k' taken to fill an order x. It is assumed that k' is an integer.

Then the first inventory level at store 2 over which control can be exercised is $V_{\ell+k'+1}$ and

$$V_{\ell+k'+1} = V_o + \sum_{i=1}^{\ell} q_i + \sum_{i=1}^{k} q'_i + x - \sum_{i=1}^{\ell+k'+1} d_i \qquad (15)$$

Hence an order rule is chosen to make

$$\varepsilon(V_{\ell+k'+1}) = I_o$$

i.e. $$I_o = V_o + \sum_{i=1}^{\ell} q_i + \sum_{i=1}^{k'} q'_i + x - (\ell+k'+1)\, su(t) \qquad (16)$$

where t is the average delivery time maintained at store 2.

After x has been found from equation (15), $q_{\ell+1}$ can be computed and then q'_{k+1} is determined by the order rule:-

$$w_1 + \sum_{i=2}^{k+1} q'_i - (k+1)\, su(t) = J_o \qquad (17)$$

The inventory in store 2 will be normally distributed thus $\{I_o;\ \sqrt{\ell+k'+1}\,\sigma u(t)\}$.

Since the weekly output from the system at store 2 is a normally distributed random variable with parameters $\{su(t), \sigma u(t)\}$, it is assumed further that the output $q_{\ell+1}$ from store 1 can be approximated reasonably well by the same normal random variable.

It then follows from (17) that the virtual stock at store 1 is normally distributed with parameters $\{J_o; \sqrt{k+1}\,\sigma u(t)\}$. (18)

The expected shortages at stores 1,2 are given respectively by $-k'su(t)$ and $-tsu(t)$.

As with the reasoning for the single store case, the expected profit per week is given as:-

$$\varepsilon(P) = psu(t) - h_2\sqrt{\ell+k'+1}\,\sigma u(t)\; A_n\{B_n^{-1}(\frac{-ts}{\sigma\sqrt{\ell+k'+1}})\}$$

$$-h_1\sqrt{k+1}\,\sigma u(t)\; A_n\{B_n^{-1}(\frac{k's}{\sqrt{k+1}\sigma})\} \quad (19)$$

where h_1, h_2 are the holding costs at stores 1,2 respectively.

The problem is now reduced to that of maximizing $\varepsilon(P)$ with respect to the variables t and k' subject to the following restrictions:-

$$\begin{cases} 0 \leq t \leq k'+\ell \\ 0 \leq k' \leq k \end{cases}$$

3.1.4 The central store/sub-store problem

The method of control and general details of the model are as for the stores in series model, the difference lying in store 2 being replaced by N independent sub-stores. The lead time for each sub-store is assumed to be ℓ weeks, and policies are restricted to those giving the same expected delivery time t to customers at each of the N sub-stores. The same decay function u(t) is assumed for all sub-stores. The holding cost for the i^{th} sub-store is h_2^i. The mean potential sales rate on sub-store i is s^i.

If I_o^i is the target value at the i^{th} sub-store the inventory level is normally distributed with parameters $\{I_o^i, \sqrt{\ell+k'+1}\,\sigma^i u(t)\}$

$i = 1,2 \ldots n$

The expected shortage is

$$B^i = -ts^i u(t)$$

Taking σ as $\sqrt{\{\sum_{i=1}^{N}(\sigma^i)^2\}}$ and S as $\sum_{i=1}^{N} s^i$, analogy with earlier considerations yields $\varepsilon(P)$, the expected weekly profit:-

$$\varepsilon(P) = p\,S\,u(t) - \sum_{i=1}^{N} h_2^{\,i} \sqrt{\ell+k'+1}\;\sigma^i u(t)$$

$$\times\, A_n\,\{B_n^{-1}\,(\frac{-ts^i}{\sqrt{\ell+k'+1}\;\sigma^i})\}$$

$$-\, h_1 \sqrt{k+1}\;\sigma u(t)\, A_n\,\{B_n^{-1}\,(\frac{-k'S}{\sqrt{k+1}\,\sigma})\}$$

where $0 \leq t \leq \ell+k'$, $\quad 0 \leq k' \leq k$.

3.2 The Work of W. Lampkin Specifically Related to the Work of F. Hanssmann

Lampkin, in his unpublished university thesis[3] and in a paper to Operations Research[4] has pointed out weaknesses in Hanssmann's work and corrected some mistakes. This work is given below.

3.2.1 Corrections to Hanssmann's work

Lampkin feels that Hanssmann's reasoning becomes invalid at the derivation of equation (15). Suppose that at the end of week zero, store 1 is empty and $q_1', q_2' \ldots q_{k^*}'$ are all needed to meet demands on store 2 which have already been placed. Then it is true that demands at the start of week one cannot affect any stock level at store 2 before $V_{\ell+k^*+1}$ but what is not true is that this order determines $V_{\ell+k^*+1}$. If the order does not allocate all of q_{k^*+1}', then a later order may allocate more of q_{k^*+1}' and this will affect $V_{\ell+k^*+1}$.

It is pointed out that the ordering rule for the quantity x (equation (16)) includes the expected number of outstanding orders by store 2, i.e. $(\sum_{i=1}^{\ell} q_i + \sum_{i=1}^{k'} q_i')$ rather than the true number. This, Lampkin says, throws away nearly all feedback and produces violent oscillation in stock levels.

It is concluded that the intended ordering rule is

$$V_o + \text{stock outstanding on order} + x - (\ell+k'+1)\,su(t) = I_o$$

i.e. $x = I_o + (\ell+k'+1)\,su(t) - V_o$ - stock oustanding on order.

This is said to be equivalent to the simpler equation:-

$x = S_o$; where S_o is the demand in week zero.

Lampkin goes on to correct expressions (17), (18), (19) by replacing k+1 by k in each of them, giving (17'), (18'), (19').

The equation $w_1 = w_o + q_1' - x$ and equation (17') together imply $q'_{k+1} = x$ and since $x = S_o$, we have:-

$$q'_{k+1} = S_o.$$

These considerations are said to lead to the following very simple control of the system:-

"Once the notional stocks have been set to the required levels, subsequent control proceeds as follows: if S_i denotes demand in the i^{th} week, orders for an amount S_i are placed by store 2 on store 1 and by store 1 on the supplier."

At this stage, two new constants are introduced.

$$J_1 = J_o + ksu(t) \qquad (20')$$

$$I_1 = I_o + (\ell+k'+1)\, su(t) \qquad (21')$$

For the moment the dependence of the demand distribution on t is suppressed. Let the demand distribution in a week be $N(s,\sigma^2)$.

Then the initial ordering rules are:-

$$x = I_1 - V_o - \text{orders by store 2 outstanding} \qquad (22')$$

$$q'_{k+1} = J_1 - w_o - \sum_{i=1}^{k} q_i' + x \qquad (23')$$

Then for $T > \ell+k$ we have:-

$$w_T = \underset{\substack{\text{Stock +}\\ \text{Outstanding}\\ \text{Orders}}}{J_1} - \underset{\substack{\text{Outstanding}\\ \text{Orders}}}{\sum_{i=T-k}^{T-1} d_i} \qquad (24')$$

and

$$V_T = \underset{\substack{\text{Stock +}\\ \text{Outstanding}\\ \text{Orders}}}{I_1} - \underset{\substack{\text{Outstanding}\\ \text{demands from}\\ \text{start of week}\\ T-\ell+1 \text{ to start}\\ \text{of week T}}}{\sum_{i=T-\ell}^{T} d_i} + \underset{\substack{\text{Outstanding}\\ \text{demands from}\\ \text{before start of}\\ \text{week } T-\ell+1}}{\text{Min}\left(w_{T-\ell}, 0\right)} \qquad (25')$$

$$\therefore\ V_T = I_1 - \sum_{i=T-\ell}^{T} d_i + \text{Min}\,\{J_1 - \sum_{i=T-k-\ell}^{T-\ell-1} d_i,\ 0\} \qquad (26')$$

Let y denote $\sum_{i=T-\ell}^{T} d_i$; y is a normal deviate with parameters

$\{(\ell+1)s, \sqrt{\ell+1}\,\sigma\}$

Let z denote $\sum_{i=T-\ell-k}^{T-\ell-1} d_i$; z is a normal deviate with parameters $\{ks; \sqrt{k}\,\sigma\}$

and y, z are independent.

It is clear that w_T is a normal deviate with parameters $\{J_o; \sqrt{k}\,\sigma\}$, which agrees with the corrected result of Hanssmann's equation (18).

3.2.2 <u>The distribution of V_T</u>

The distribution of V_T, called v for convenience, is to be found.

$$v = \begin{cases} I_1 - y & \text{for } z \leqslant J_1 \\ I_1 + J_1 - y - z & \text{for } z > J_1 \end{cases}$$

Thus the distribution of v is

$$\psi(v)\,dv = \left\{\frac{1}{\sigma\sqrt{\ell+1}}\, P\left(\frac{J_1-ks}{\sigma\sqrt{k}}\right) Z\left(\frac{I_1-v-(\ell+1)s}{\sigma\sqrt{\ell+1}}\right)\right.$$

$$\left. + \frac{1}{\sigma^2\sqrt{k(\ell+1)}} \int_{J_1}^{\infty} Z\left(\frac{z-ks}{\sigma\sqrt{k}}\right) Z\left(\frac{I_1+J_1-v-z-(\ell+1)s}{\sigma\sqrt{\ell+1}}\right) dz\right\} dv$$

where $P(X) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{X} e^{-\frac{1}{2}x^2}\,dx$; $Z(X) = \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2}X^2}$

Using this equation, it can be shown that

$$\varepsilon(v) = \int_{-\infty}^{\infty} v\psi(v)\,dv = \bar{v}$$

$$= J_o(1-P_1) + I_1 - (\ell+1)s - \sigma\sqrt{k}\, Z_1 \qquad (27')$$

$$\text{Also } \mathrm{Var}(v) = (\ell+1)\sigma^2 + k\sigma^2(1-P_1)$$

$$+ (J_oP_1 + \sigma\sqrt{k}\, Z_1)(J_o(1-P_1) - \sigma\sqrt{k}\, Z_1) \qquad (28')$$

where $P_1 = P\left(\frac{J_o}{\sigma\sqrt{k}}\right)$ and $Z_1 = Z\left(\frac{J_o}{\sigma\sqrt{k}}\right)$

The integration in the expression for $\psi(v)\,dv$ may be achieved and yields:-

$$\psi(v)\,dv = \{\frac{1}{\sigma\sqrt{\ell+1}} P(\frac{J_1-ks}{\sigma\sqrt{k}})\, Z\,(\frac{I_1-v-(\ell+1)s}{\sigma\sqrt{\ell+1}})$$

$$+ \frac{1}{\sigma\sqrt{\ell+k+1}} \left(1-P(\frac{\ell J_1+kv-kI_1}{\sigma\sqrt{k(\ell+1)(\ell+k+1)}})\right) Z(\frac{J_1+I_1-v-(\ell+k+1)s}{\sigma\sqrt{\ell+k+1}})\}\, dv$$

3.2.3 Comparison of the V_T distributions of both Hanssmann and Lampkin

The question posed here is how does this distribution of v compare with Hannsmann's solution that v is a normal deviate with parameters $\{I_o;\ \sqrt{\ell+k'+1}\sigma\}$ where

$$k' = -\frac{\sigma\sqrt{k}}{s}\ B_n\ (\frac{J_o}{\sigma\sqrt{k}})$$

Evaluation of k' in terms of the P, Z functions yields:-

$$sk' = \sigma\sqrt{k}\ Z_1 - J_o(1-P_1)$$

and so equation (27') reduces to:-

$$\varepsilon(v) = I_1 - (\ell+1)s - k's = I_o$$

and so agreement is established with Hanssmann on the mean of the distribution, viz. I_o.

The variances differ, however. Hanssmann's result is:-

$$\sigma_H^{\ 2} = (\ell+k'+1)\sigma^2$$

$$= (\ell+1)\sigma^2 + \frac{\sigma^2}{s}\{\sigma\sqrt{k}\ Z_1 - J_o(1-P_1)\}$$

which is not the same as (28').

Lampkin comments that Hanssmann's result is a function not only of ℓ, k, σ^2, J_o, as his own, but also of s. He then gives the following argument to indicate that the variance should not depend on s:

Suppose each weekly demand was augmented by a further large invariant weekly demand. If the target levels I_o and J_o were retained, it is clear that the stock levels in both stores would not be affected. The distribution of v and hence the variance would be unaltered.

The next step Lampkin considers is that of investigation as to whether the shape of the v distribution is roughly normal, and if a normal distribution could reasonably be used as an approximation to $\psi(v)$.

Firstly, it is stated, the region of doubt is where $(J_o/\sigma\sqrt{k})$ is low since $J_o/\sigma\sqrt{k} \to \infty$ leads to $\psi(v)\,dv$ approaching a normal distribution with parameters $\{I_o;\ \sqrt{\ell+1}\ \sigma\}$. A value of $J_o/\sigma\sqrt{k}$ "as low as would be met in practice" is investigated and the corresponding $\psi(v)$ compared to a normal distribution with the same mean and variance. It is shown by graphing the two distributions that the two distributions show a marked resemblance.

From the expressions for $\varepsilon(v)$, $\mathrm{Var}(v)$, and from $w_T = N(J_o;\ \sqrt{k}\ \sigma)$, the following cost function, suggested as an improvement to that postulated by Hannsmann, is derived.

$$\mathcal{E}(P) = psu(t) - h_2\xi(J_o,t)\ A_n\{B_n^{-1}\ (\frac{-tsu(t)}{\xi(J_o,t)})\} - h_1\sqrt{k}\sigma u(t)\ A_n\{\frac{J_o}{\sigma\sqrt{k}\,u(t)}\}$$

where $\xi(J_o,t)$ is given by:-

$$\xi^2(J_o,t) = (\ell+1)\sigma^2u^2(t) + k\sigma^2u^2(t)\ \left(1-P(\frac{J_o}{\sqrt{k}\sigma u(t)})\right)$$

$$+ \{J_oP(\frac{J_o}{\sqrt{k}\sigma u(t)}) + \sigma u(t)\sqrt{k}\ Z\ (\frac{J_o}{\sqrt{k}\sigma u(t)})\}$$

$$\times\ \{J_o\ \left(1-P(\frac{J_o}{\sqrt{k}\sigma u(t)})\right) - \sigma u(t)\ k\ Z(\frac{J_o}{\sqrt{k}\sigma u(t)})\}$$

where $0 \leqslant t \leqslant k+\ell$ and the minimum value of $\mathcal{E}(P)$ is to be found over the (J_o,t) plane.

3.2.4 The central store/sub-store problem

Lampkin's rule for sharing to sub-stores follows:-

"The quantity available shall be issued in such a manner that for each sub-store, the amount by which the total in the pipeline to the sub-store falls short of the quantity outstanding on order shall be proportional to the average demand from the sub-store."

It is said that this is usually, but not always, possible. In practice, this condition would be met as closely as possible. The following analysis is presented on the assumption that sharing according to the given rule is always possible, and thus it is said to be in error to this extent.

3.2.4.1 The distribution of V_T

As in the case of two stores in series, the dependence of the demand distribution parameters in t is suppressed and

w_T is the stock at store 1 at the end of week T

V_T^i is the stock at sub-store i at the end of week T

J_o is the target stock for store 1

I_o^i is the target stock for sub-store i

S_t^i is the demand at sub-store i in week t.

$$S_t = \sum_{i=1}^{N} S_t^i$$

Then if $\quad J_1 = J_o + kS$

and $\quad I_1^i = I_o^i + (\ell+k'+1)\, S^i$

Then $$w_T = J_1 - \sum_{t=T-k}^{T-1} S_t$$

and $$V_T^i = I_1^i - \sum_{t=T-\ell}^{T} S_t^i + \frac{S^i}{S}\left(\min\{w_{T-\ell},\ 0\}\right)$$

$$= I_1^i - \sum_{t=T-\ell}^{T} S_t^i + \frac{S^i}{S}\left(\min\{(J_1 - \sum_{t=T-k-\ell}^{T-\ell-1} S_t),\ 0\}\right)$$

Let y denote $\sum_{t=T-\ell}^{T} S_t^i$; y is a normal deviate with parameters $\{(\ell+1)S^i ;\ \sqrt{\ell+1}\ \sigma^i\}$

Let z denote $\sum_{t=T-\ell-k}^{T-\ell-1} S_t$; z is a normal deviate with parameters

$\{kS;\ \sqrt{k}\ \sigma\}$ and y, z are independent. Replacing V_T by v we have:-

$$v = \begin{cases} I_i^i - y & \text{for } z \leqslant J_1 \\ I_i - y + \frac{S^i}{S}(J_1 - z) & \text{for } z > J_1 \end{cases}$$

The distribution of v is given by

$$\psi(v) = \frac{1}{\sigma^i\sqrt{\ell+1}} P\{\frac{J_1 - ks}{\sigma\sqrt{k}}\}\ Z\ \{\frac{I_1^i - v - (\ell+1)S^i}{\sigma\sqrt{\ell+1}}\}$$

$$+ \frac{1}{\sigma^i \sigma\sqrt{k(\ell+1)}} \int_{J_1}^{\infty} Z\{\frac{z-ks}{\sigma\sqrt{k}}\}\ Z\{\frac{I_1^i - v - (\ell+1)S^i + S^i(J_1 - z)/S}{\sigma^i\sqrt{\ell+1}}\}\ dz$$

which leads to:-

$$\varepsilon(v) = I_1^i - (\ell+1)\ S^i + \frac{S^i}{S}\ \left(J_o(1-P_1) - \sigma\sqrt{k}\ Z_1\right)$$

$$= I_o$$

$$\mathrm{Var}(v) = (\sigma^i)^2 (\ell+1) + \sigma^2 k \left(\frac{S^i}{S}\right)^2 (1-P_1)$$

$$+ (J_o P_1 + \frac{S^i}{S}\sigma\sqrt{k}\, Z_1)(J_o(1-P_1) + \frac{S^i}{S}\sigma\sqrt{k}\, Z_1)$$

where $P_1 = P(J_o/\sigma\sqrt{k})$; $Z_1 = Z(J_o/\sigma\sqrt{k})$.

3.2.4.2 The profit function

The following expected weekly profit is given as a suggested improvement over that presented by Hanssmann:

$$\varepsilon(P) = pSu(t) - h_1\sqrt{k}\sigma u(t)\, A_n \left\{\frac{J_o}{\sigma\sqrt{k}u(t)}\right\}$$

$$- \sum_{i=1}^{n} h_2^{\,i}\, \xi^i(J_o,t)\, A_n\{B_n^{-1}\left(\frac{-tS^i u(t)}{\xi^i(J_o,t)}\right)\}$$

where $\xi^i(J_o,t)$ is given by:-

$$\{\xi^i(J_o,t)\}^2 = (\sigma^i)^2(\ell+1)u^2(t) + \sigma^2 u^2(t)\, k\, \frac{(S^i)^2}{S}\left(1-P\left(\frac{J_o}{\sqrt{k}\sigma u(t)}\right)\right)$$

$$+ \{J_o P\left(\frac{J_o}{\sigma\sqrt{k}u(t)}\right) + \frac{S^i}{S}\sigma\sqrt{k}\, Z\left(\frac{J_o}{\sqrt{k}\sigma u(t)}\right) \times$$

$$\times \{J_o\left(1-P\left(\frac{J_o}{\sigma\sqrt{k}u(t)}\right)\right) + \frac{S^i}{S}\sigma\sqrt{k}\, Z\left(\frac{J_o}{\sigma\sqrt{k u(t)}}\right)\}$$

Thus $\varepsilon(P)$ is now a function of J_o,t, and the maximum profit results from the use of those J_o,t values which maximize $\varepsilon(P)$.

3.2.5 Discussion of the work of F. Hanssmann and the work of W. Lampkin applied to Hanssmann's paper

The combined works of the two authors are open firstly to criticism on the assumptions made initially. Firstly the assumption that demand is related to the average waiting time of customers is a perfectly feasible alternative to the usual concept of a shortage cost resulting if a demand cannot be made ex stock at the store on which the demand is made. However this assumption does limit the field of applicability of Hanssmann's model. There are many instances where the penalty for a shortage is felt on the demand side in the form of (a) fixed cost because unit is not available when demanded or (b) fixed cost plus time dependent cost or (c) time dependent cost only (i.e. cost type (b) where the fixed cost is zero).

The demand does not necessarily have to drop off because of the non-immediate character of delivery. This is especially true when the demand is from a source which is under the same ownership as the central store/sub-store system itself. Even where the model is applicable, it is suggested that practical estimates of the function u(t) are likely to be inadequate. If the demand distribution is seen to vary in practice, it will be very difficult to assign just what proportion is due to the customer waiting time function u(t) and what is due to other changes of a dynamic nature, except for those distributions which are stationary. (These are rather limited.)

The assumption that demands are captive is necessary for the analysis, and although this assumption is restrictive, it does not unduly limit the applicability of the model. The assumption of normal distribution of the demands makes the analysis simpler, especially in that it allows employment of the normalised overage and shortage functions. However it is felt that requiring a normal distribution of demands makes the model unfortunately restrictive, from the overall contribution to the point of view of control of stock in a central store/sub-store complex.

The next point, namely that stock is reviewed simultaneously at all sub-stores after equal time intervals and is ordered on the central store without knowledge of the other sub-store orders, represents a very large restriction on policy considerations. Firstly, reviewing at equal time instants immediately precludes reorder level considerations, which are shown by Lampkin[3] to result in "overwhelming advantage ... in the costs of stockholding and shortages at the sub-stores". Further, (although of course this requires centralised knowledge of stocks at various sub-stores) the control can be made much more efficient at sub-stores by taking into account the stock positions of all other sub-stores.

Hanssmann develops his analysis with the assumption of attaining target inventory levels at all stores (central and sub-store), and then optimizing ordering decisions. It is not clear that the use of the criterion of target inventory is very useful; no support or evidence is offered to indicate the value of this policy. However the analysis proceeds to the development of an expected profit function in terms of t (the average customer waiting time) and k' (the average time to fill an order from a sub-store). Search over the (k',t) plane will provide the k',t to maximize this function. From these the target inventories are obtainable. The method of analysis development, namely considering firstly a single store, then two stores in series, and then the central store feeding sub-stores is a rather useful approach.

Some of the results of Hanssmann's work are clearly wrong and Lampkin corrects these. The present author feels that Lampkin does a useful job in strengthening the model presented by Hanssmann by showing up and correcting some false thinking. Lampkin's rule for sharing stock out to sub-stores in those cases when the total order on the central store is too much to be met completely is thought to be appropriate to the type of control being considered. The final expression for expected profit as presented by Lampkin is a function of the target stock for the central store and t, and thus searching over the (J_o,t) plane to yield the optimal J_o,t is required.

The assumption made by Hanssmann when extending his "Two Stores in Series" Model to "The Central Store/Sub-store Problem", (namely, that policy considerations would be restricted to those having the same t value at each sub-store) is, of course, very useful in permitting simplification of analysis. However, it places the final control policy further away from optimal.

Other assumptions, implicit in Hanssmann's model (e.g. lead times to sub-stores constant and identical, lead time from supply constant) may all be criticised from the point of view of being restrictive, but this does not mean that they are unnecessarily so. It is recognized that models often require recourse to such restrictions to be workable.

3.3 The Work of Lawrence, Stephenson, and Lampkin

3.3.1 Introduction

We deal here with the work of Lawrence and his associates at the National Coal Board. Inventory levels for spare parts for coal-face machinery were calculated and the work published in a paper to Operational Research Quarterly.[5]

The central store was operated on a reorder level system, triggered by central store stock. The sub-stores were controlled by a base stock method. The demands on the sub-stores were from independent Poisson processes and were captive. The lead times to N sub-stores were identical and constant, and the central store lead time was constant. A fixed shortage cost of c_{s_1} resulted every time a demand on a sub-store could not be met ex sub-store stock but could be met ex central store stock. A cost of c_{s_2} resulted if neither of these stores could meet the demand. As is usual in such models, the holding cost was proportional to average inventory. No transfers between sub-stores were allowed.

3.3.2 Symbols

The following symbols appear throughout the work:

Q = reorder quantity at central store

λ_i = annual demand at sub-store i

λ_T = $\sum_{i=1}^{N} \lambda_i$

P = price of item

L = central store lead time

ℓ = sub-store lead time

h = fraction of the price of an item which is the cost of holding it for one year

m_i = maximum stock level at store i

M = reorder level at central store

c_p = cost of an order at the central store

U = $\ell\lambda_i$

3.3.3 Analysis for the sub-stores

Attention is concentrated on one sub-store. The assumption is made for the present, that the central store never fails in its ability to supply.

The number of annual shortages* is:-

$$\lambda \sum_{n=m}^{\infty} \frac{U^n}{n!} e^{-U} \qquad (1), \quad \text{dropping the suffix "i"}$$

The sub-store average stock is:-

$$m \sum_{n=0}^{m} \frac{U^n}{n!} e^{-U} - U \sum_{n=0}^{m-1} \frac{U^n}{n!} e^{-U} \qquad (2)$$

If $\bar{r}$ denotes the average cost of a shortage at the sub-store, we may write:-

$$\bar{r} = r_1(1-\omega) + r_2\omega$$

where r_1, r_2 are the unit shortage costs if the shortage is respectively met or not met by the central store. ω is the proportion of shortages not met by the central store.

---

* Here this author thinks that the word "expected" is omitted.

The average annual cost at the sub-store is then

$$C_m = \lambda \bar{r} - \lambda(\bar{r} + Ph\ell) \sum_{n=0}^{m-1} \frac{U^n}{n!} e^{-U} + Phm \sum_{n=0}^{m} \frac{U^n}{n!} e^{-U}$$

whence:

$$C_{m+1} - C_m = Ph \sum_{n=0}^{m} \frac{U^n}{n!} e^{-U} - \bar{r}\lambda \frac{U^m}{m!} e^{-U} \qquad (3)$$

The boundary between the decision to hold stock m and stock (m+1) will be when:-

$$C_{m+1} = C_m$$

This is easily obtainable from (3):-

$$Ph\ell/\bar{r} = \frac{U \left(\frac{U^m}{m!} e^{-U}\right)}{\sum_{n=0}^{m} \frac{U^n}{n!} e^{-U}} = f\,\{U, m\}$$

A $(Ph\ell/\bar{r})$ vs. U chart can be drawn up to give the optimum base level of stock m to be held.

3.3.4 Analysis for the central store

It has been shown[6] that in a reorder level system the average stock is given by:-

$$F(M,Q) = \sum_{s=0}^{M} \left(M + \frac{Q+1}{2} - s\right) A(s) + \frac{1}{2Q} \sum_{s=M+1}^{M+Q} (M+Q+1-s)(M+Q-s)\, A(s)$$

and the average number of demands not met immediately per annum is given by:-

$$G(M,Q) = \lambda_T \left(1 - \sum_{s=0}^{M} A(s) - \frac{1}{Q} \sum_{s=M+1}^{M+Q} (M+Q-s)\, A(s)\right)$$

where A(s) is the probability of s demands in time L.

For a Poisson distribution of demands, these functions become:

$$F(M,Q) = \frac{1}{2Q} \Big((M+Q)(M+Q+1-\lambda_T L) P(\lambda_T L, M+Q+1) - \lambda_T L (M+Q-\lambda_T L) P(\lambda_T L, M+Q) - M(M+1-\lambda_T L) P(\lambda_T L, M+1) + \lambda_T L (M - \lambda_T L) P(\lambda_T L, M) \Big)$$

and $$G(M,Q) = \lambda_T \left(1 - \frac{1}{Q}\{(M+Q)P(\lambda_T L, M+Q+1) - \lambda_T L P(\lambda_T L, M+Q) - MP(\lambda_T L, M+1) + \lambda_T L P(\lambda_T L, M)\}\right)$$

where $$P(m,c) = \sum_{s=0}^{c-1} \frac{m^s}{s!} e^{-m} .$$

The system cost function is thus:-

$$C(M,Q) = PhF(M,Q) + RG(M,Q) + c_p\lambda_T/Q$$

where R is a notional run-out cost, not known at the moment.

3.3.5 The calculation of R

Let e^* be the average time period from the instant the central store is depleted until it receives stock. Then the average time sub-stores have to manage without being replenished is $e' = e^* + \ell$.

e^* is the expected part of the lead time that the virtual stock at the central store is zero or less (given that this state is attained).

Hence

$$e^* = L(1 - a/b)$$

where $$a = \frac{M}{L\lambda_T} \sum_{s=M+1}^{\infty} \frac{(L\lambda_T)^s}{s!} e^{-\lambda_T L} \quad \text{and} \quad b = \sum_{s=M}^{\infty} \frac{(L\lambda_T)^s}{s!} e^{-\lambda_T L}$$

Each sub-store i has a notional stock of m_i at the beginning of the period. The cost of shortages in the period is thus approximately:

$$C_a = r_2 \sum_{i=1}^{N} \sum_{s=m_i+1}^{\infty} \frac{(e'\lambda_i)^s}{s!} (s-m_i) e^{-e'\lambda_i}$$

In normal circumstances the shortage costs would have been:

$$C_b = \sum_{i=1}^{N} r_1 \lambda_i e' \sum_{s=m_i}^{\infty} \frac{(U)^s}{s!} e^{-U}$$

The shortage at the central store forces an average saving in stock holding costs at the sub-stores of approximately:

$$C_h = Ph(e'-\ell) \frac{\lambda_T}{2} (e'-\ell)$$

The average cost of the system being forced to last for e' without extra supplies is:-

$$C = C_a - C_b - C_h$$

The average number of items demanded from the central store in the period e' is $e'\lambda_T$ and so the average cost of an unmet demand is $R = C/e'\lambda_T$.

Note that the calculation of R entails knowledge of the sub-store stock levels m_i and the central store parameters (M,Q).

3.3.6 The calculation of $\bar{r}$

ω = the proportion of emergency demands (after sub-store run-out) which cannot be met by the central store $= G(M,Q)/\lambda_T$, and $\bar{r} = r_1(1-\omega) + r_2\,\omega$.

3.3.7 Reconciliation of the Calculations

Note that

(a) To calculate the m_i, we need $\bar{r}$.

(b) To calculate (M,Q) we need R.

(c) To calculate $\bar{r}$ we need (M,Q).

(d) To calculate R we need the m_i and (M,Q).

It is argued that the definitions are circular. An iterative process is suggested which has been found to converge. No theoretical proof for convergence is offered.

3.3.8 Comments on the Lawrence paper

By stipulating a base-stock system at sub-stores, Lawrence immediately establishes sub-optimal operation of the whole complex. Delivery costs to sub-stores are now no longer considered since delivery policy has already been decided.

Treating the analysis in terms of two models, one for sub-stores, the other for the central store, is a useful idea but for the fact that the two analyses are not really independent (hence the circular definitions resulting finally). Maintaining a reorder level policy at the central store and not allowing inter sub-store replenishment, leads to the usual criticism of overall non-optimality, but of course, Lawrence is not suggesting the latter anyway.

The analysis for the central store presupposes that shortage costs vary directly as the number of orders not met immediately ex central store stock. However, a few shortages are likely to be of little consequence cost-wise whereas a large number (thus making a penalty of r_2 more likely) would cost more then proportionately more, on average.

3.4 The Work of P. Winters

In a paper[21] published in 1960, Winters considers the control of stock in a stores complex without a central store. Procurement of further stock is obtained from a factory production unit. An extension of this work to cover the case of the central store/sub-store complex problem is considered in his unpublished thesis[7].

Winters divides control consideration into two individual considerations. Firstly, he considers the criterion for triggering a procurement, and secondly, the procurement quantity and how to split this up for delivery to the stores. This is done by establishing two cost equations, the first of which involves only the stock levels at the different stores for making the procurement order decision. It is admitted that this entails ignoring the interaction between procurement quantity and the procurement order trigger.

In order to build these cost equations the cost of holding buffer stock for the length of time in any procurement cycle other than the procurement lead time is ignored. Also, the individual store reorder quantities (breakdown of the procurement quantity) are chosen without reference to the expected costs of shortage.

The cost considered for establishing the trigger includes the expected sum of holding and shortage costs in the respective stores' procurement lead times. Couched for simplicity in terms of a complex with only two stores, Winters gives this cost sum as:-

$$\tfrac{1}{2}h_1 P \ell_1 \{2s_1 - \lambda_1 \ell_1\} + \tfrac{1}{2}(h_1 P + c_{s_1}) \int_{s_1/\ell_1}^{\infty} \frac{(s_1 - \lambda_1 \ell_1)^2}{s_1} f(\lambda_1)\, d\lambda_1$$

$$+ \tfrac{1}{2}h_2 P \ell_2 \{2s_2 - \lambda_2 \ell_2\} + \tfrac{1}{2}(h_2 P + c_{s_2}) \int_{s_2/\ell_2}^{\infty} \frac{(s_2 - \lambda_2 \ell_2)^2}{s_2} f(\lambda_2)\, d\lambda_2$$

where, for store i,

- h_i is the holding cost per unit cost per unit time
- ℓ_i is the lead time
- s_i is the inventory at time of triggering procurement
- λ_i is the demand rate
- P is the cost or price of a unit of stock
- c_{s_i} is the unit shortage cost per unit time*
- $f(\lambda_i)$ is the probability distribution of demand

* Note Winters' model is only applicable for shortage cost being a linear function of time short.

Winters goes on to show that the values of s_1, s_2 which minimize this expected cost during the lead time are given by the following equation:

$$\frac{h_1}{h_2}\{\ell_1 + (1+\frac{c_{s_1}}{h_1 P})\int_{s_1/\ell_1}^{\infty} \frac{s_1-\lambda_1 T_1}{\lambda_1} f_1(\lambda_1)\, d\lambda_1\}\lambda_1$$

$$+ \lambda_2\{\ell_2 + (1+\frac{c_{s_2}}{h_2 P})\int_{s_2/\ell_2}^{\infty} \frac{s_2-\lambda_2 T_2}{\lambda_2} f_2(\lambda_2)\, d\lambda_2\} = 0$$

This "trigger equation" is an implicit function of the stock levels and generates a curve on the s_1,s_2 plane which is hyperbolic in form. If the s_1,s_2 combination point moves below this curve, a procurement should be ordered. (This point is generalised in Winters' thesis to a function "f" of the stocks at the stores. Whenever this function changes sign from positive to negative, the procurement order is placed.)

To obtain the procurement quantities for the various stores, the following total cost function is built:

$$T.C. = c_P + \tfrac{1}{2}h_1 P\{2(\hat{I}_1-\lambda_1\ell_1) + Q_1\}L_1 + \tfrac{1}{2}h_2 P\{2(\hat{I}_2-\lambda_2\ell_2)+ Q_2\} L_2$$

where $\hat{I}_1$, $\hat{I}_2$ are the inventory levels at time of procurement trigger for stores 1,2 and where Q_1, L_1, Q_2, L_2 respectively refer to the procurement quantity and cycle time (time between arrivals of stock) for stores 1 and 2. c_P is the procurement cost. Clearly, it is argued, $L_1 = L_2$. The imposition of this restraint using the Lagrange multiplier technique in forming the function $\phi = \frac{T.C.}{L_1} + \lambda(L_1-L_2)$, taking partial derivatives, and setting them equal to zero yields optimal Q_1, Q_2:

$$\hat{Q}_1 = \sqrt{\{2c_P\lambda_1^{\,2}/P(h_1\lambda_1+h_2\lambda_2)\}} ,$$

$$\hat{Q}_2 = \hat{Q}_1(\frac{\lambda_2}{\lambda_1})$$

The extension of the ideas to cover the case when the stores complex is supplemented by a central store regards the procurement as being triggered when there is no way of allocating the central store stock to the sub-stores so that the trigger function calculated on the new levels remains positive. The procurement quantity and its distribution to various sub-stores is established as above.

3.4.1 Comments on the work of P. Winters

This work suffers from the distinct error that the two functions, the trigger and the procurement quantity are computed independently. The error that the trigger equation ignores the cost of holding the buffer stock outside of the procurement lead time is a very important error for those cases when the lead time /cycle time ratio is not large. Further, the cost of shortages should be taken into account in the choice of procurement quantity.

3.5 The Work of Hadley and Whitin

3.5.1 Summary

In 1961, G. Hadley and T.M. Whitin published[12] their model for a single echelon, multi-store supply problem for items of extremely low demand governed by a stationary Poisson process. Information on the overall stock position of this stores complex is held centrally, and both procurement (supply to the complex) and store redistribution* (either of two modes) lead times are assumed constant. The complex is operated as a base stock system and decision rules are developed for allocating the unit procurement, for redistribution of stocks amongst the stores, and for establishing how much stock should be retained at each store. These rules are established by the criterion of minimizing the expected overall costs of deliveries and shortage.**

Store lead time is supposed to comprise both an administration and delivery time, and when more than a single unit is on order, the allocation of a unit ready to be delivered is determined by the solution of a dynamic programming problem. (It does not automatically go to the store initiating the procurement order.) In the case of only a single unit on order by the stores complex, the optimal allocation procedure is reduced to allocating the unit to that store which has the greatest probability of demanding it in a time period equal to the procurement lead time L plus $(1/\lambda_T)$, where λ_T is the total demand rate in the stores complex.

3.5.2 Stock objectives

Shortage costs are denoted by c_{s_i} per unit time for store i, (i=1,2 ...N) at each of the N stores. Additionally, a cost c_{ss} is assumed to occur per unit time if there is no physical stock in the complex. The

---

* i.e., from another store.

** This author feels that holding costs should be included here.

maximum stock of the base-stock system is m. It is stated that the optimal value of m may be determined by balancing stock holding costs against the costs due to shortage and redistribution. Procurement costs are said to be irrelevant since for a base-stock system the procurement quantity will be fixed.

3.5.2.1 The case of zero store lead time and zero cost of redistribution

This trivial case is considered as an introduction to the general approach. Firstly, the only shortage costs are those due to shortage in the stores complex. Clearly, individual store maximum stocks are irrelevant. The maximum stock in the complex, m, is obtained by balancing holding costs against shortage costs for the complex.

The expected physical stock in the complex at any random instant of time will be:-

$$\sum_{x=0}^{m} (m-x) \times \{\text{Probability of demand } \sout{\text{in complex}} \text{ of } x \text{ in complex in time } L\}$$

$$= \sum_{x=0}^{m} (m-x)\, (\lambda_T L)^x \, \frac{e^{-\lambda_T L}}{x!}$$

The expected number of backorders will be:-

$$\sum_{x=m+1}^{\infty} (x-m)\, (\lambda_T L)^x \, \frac{e^{-\lambda_T L}}{x!}$$

The total cost per unit time will then be:-

$$TC_1(m) = hP \sum_{x=0}^{m} (m-x)\, p(x) + c_{ss} \sum_{x=m+1}^{\infty} (x-m)\, p(x) \qquad (1)$$

where hP is the holding cost per unit stock per unit time and

$$p(x) = (\lambda_T L)^x e^{-\lambda_T L}/x! \quad . \quad .$$

Equation (1) leads to optimal m being the minimum m satisfying:-

$$\sum_{x=m+1}^{\infty} p(x) < \frac{hP}{hP+c_{ss}} \qquad (2)$$

3.5.2.2 The general case of non-zero lead time of redistribution with cost attached

The purpose of redistribution is said to be the elimination of shortages in the time interval between allocations. The assumption is now made that a redistribution is never made to a store until it acquires a backorder. Attempt to justify this is given by the statement that since the complex is stocking low-demand items, stock levels at most stores will frequently be either 0 or 1. The assumption is also made that when shortage occurs, a redistribution will always occur, and that the shortage lasts for the full length of this redistribution lead time (and therefore that no allocation from a procurement will arrive during this time).

With these assumptions, the cost associated with a shortage at store i is:-

$$\hat{c}_{s_i} = c_x + c_{s_i} \ell_x \tag{3}$$

where c_x is the cost of redistribution by the appropriate mode of transportation* and ℓ_x is the redistribution lead time. Both c_x, ℓ_x are averaged values, since they depend on which store does the redistribution to store i.

3.5.2.2.1 Distribution of demand at a store between allocations

The time period between allocations will have the same distribution as the time period between demands in the complex (viz. $\lambda_T e^{-\lambda_T t}$ for a Poisson process) since the procurement lead time is assumed constant.

The probability of a demand d on store i between allocations is then:

$$p_{ba_i}(d) = \int_0^\infty \frac{(\lambda_i t)^d}{d!} e^{-\lambda_i t} (\lambda_T e^{-\lambda_T t}) dt$$

$$= \left\{\frac{\lambda_i}{\lambda_i+\lambda_T}\right\}^d \left\{\frac{\lambda_T}{\lambda_i+\lambda_T}\right\} \quad ; \quad d \geq 0 \tag{4}$$

from integration by parts,

where λ_i is the rate of demand per unit time for store i.

3.5.2.2.2 Expression for overall cost of shortages and redistribution

Suppose there exist ξ units at store i when an allocation is received; the expected shortages (further) occurring in the time period between allocations will be:-

$$\sum_{d=\xi+1}^{\infty} (d-\xi)\, p_{ba_i}(d) \tag{5}$$

---

* to be discussed later.

If the probability of existence of ξ units at store i when an allocation is received is $p_{e_i}(\xi)$ then the expected further shorages between allocations is:-

$$\sum_{\xi} p_{e_i}(\xi) \sum_{d=\xi+1}^{\infty} (d-\xi)\, p_{ba_i}(d) \tag{6}$$

The number of different allocations per unit time is λ_T and so, the overall expected cost of shortages and redistribution per unit time is:-

$$\lambda_T \sum_{i=1}^{N} c_{s_i} \sum_{\xi} p_{e_i}(\xi) \sum_{d=\xi+1}^{\infty} (d-\xi)\, p_{ba_i}(d) \tag{7}$$

No attempt is made to compute $p_{e_i}(\xi)$. The procedure adopted is to let s_i be the safety stock maintained at store i (i.e. maximum stock at store i is $s_i+\lambda_i L$). Since $\lambda_T L$ will be the average stock on order by the stores complex, the maximum stock for the complex is given by

$$m = \lambda_T L + \sum_{i=1}^{N} s_i \tag{8}$$

Now the assumption is made that since the item is low-demand by nature, s_i will be 0 or 1 for most stores; and so an approximation for the virtual stock at store i after an allocation is s_i — Assumption (a)

Expression (7) is thus approximated to:-

$$\lambda_T \sum_{i=1}^{N} c_{s_i} \sum_{d=s_i+1}^{\infty} (d-s_i)\, p_{ba_i}(d) \tag{9}$$

3.5.2.2.3 <u>The total cost expression</u>

The total costs per unit time are then

$$TC(m) = TC_1(m) + \lambda_T \sum_{i=1}^{N} c_{s_i} \sum_{d=s_i+1}^{\infty} (d-s_i)\, p_{ba_i}(d) \tag{10}$$

where $TC_1(m)$ is the expression given by equation (1).

In order for any given set of s_i to minimize (10) it is necessary that:-

$$\Delta TC(s_i) = TC(s_i+1) - TC(S_i) \geq 0$$

$$\Delta TC(s_i-1) < 0, \qquad i = 1, \ldots N$$

Thus the smallest s_i ($i = 1,2,\ldots, N$) is chosen so that:-

$$\Delta TC(s_i) = hP-(hP+c_{ss}) \sum_{x=m+1}^{\infty} p(x) - \lambda_T c_{s_i} \sum_{d=s_i+1}^{\infty} p_{ba}(d) \geq 0 \tag{11}$$

If the complex shortage cost, c_{ss}, is negligible, then the smallest s_i is chosen so that:-

$$\sum_{d=s_i+1}^{\infty} p_{ba}(d) \leq \frac{hP}{\lambda_T \hat{c}_{s_i}} \quad , \quad (i = 1, \ldots, N) \qquad (12)$$

and hence m, by use of equation (8).

3.5.2.3. Case where procurement lead time is less than redistribution lead time

Hadley follows on by pointing out that if the procurement lead were ever less than the redistribution lead time, redistribution would never be used (Hadley's model implicitly assumes that the supply to the complex never fails). In this case the optimal store maximum stock levels are m_i for store i such that m_i is the smallest m_i satisfying:-

$$\sum_{d=m_i+1}^{\infty} p_i(d) \leq \frac{hP}{hP+c_{s_i}}$$

where $p_i(d)$ is the probability of a demand for d items at store i in the procurement lead time. The given inequality is obtainable by minimizing the costs of holding and shortage for each store individually.

3.5.2.4 The decentralised model and the centralised model

3.5.2.3. gives a clear instance where the "centralised model" considered prior to 3.5.2.3 would not be sensible. Hadley points out that for cases where procurement lead time is greater than redistribution lead time, the decentralised model of 3.5.2.3 may possibly still lead to lower expected costs. In these cases, the decentralised model is clearly to be preferred.

3.5.2.5 Conclusion of Hadley's work

The work is concluded by the treatment of the problems of (i) selection of mode of transportation from the source, (ii) rules of allocating the unit stocks on order to the various stores, and (iii) whether to redistribute or not, and if so, by what mode of transportation.

For (i) Hadley's result is quoted. The optimal mode of transportation from the supplier or source is the one that minimizes:-

$$\sum_{i=1}^{N} c_{is}^{r} \lambda_i + TC_1(m)$$

where c_{is}^{r} is the cost of shipping one unit from the source to store i by mode r and m is that value of m appropriate to the mode of transportation considered.

For (iii) two cost expressions are compared, and their difference ΔC is considered. If $\Delta C \geqslant 0$, no redistribution should be made since this is more costly than awaiting the procurement arrival. If $\Delta C < 0$, redistribution should be made by that store and mode of transportation that minimize ΔC.

The result for (ii) is quoted in the summary at 3.5.1.

3.5.3 Discussion of Hadley' and Whitin's work

The assumption (a), page 50 in 3.5.2.2 may be criticised. It does not follow that since stock levels at most stores are 0 or 1, redistribution does not occur until there is a backorder. This surely depends on the redistribution rule itself. Further, stock levels are not necessarily usually 0 or 1. The cost case Hadley considers itself has safety stock levels of 2 and 1.

Clearly, it is more likely that redistribution will be required for a store experiencing a backorder than for the case where a backorder is not produced by a demand, but the assumption that a redistribution necessarily occurs is not supported, and does not appear generally reasonable.

The cost of such a redistribution and the corresponding shortage cost is:

$$c_x + c_{s_i} \ell_x.$$

The cost of awaiting the procurement is

$$c_{s_i} t';$$

where t' is the earliest time a procurement which has not been allocated can get to this store.

Clearly if $t' \leqslant \ell_x$, there is never any point in redistribution; neither is there if:-

$$c_{s_i} t' \leqslant c_x + c_{s_i} \ell_x$$

$$\text{i.e.} \quad t' \leqslant \frac{c_x}{c_{s_i}} + \ell_x$$

Redistribution then should occur if $t' > \dfrac{c_x}{c_{s_i}} + \ell_x$

and then the back order will clearly last the time ℓ_x; this is not an assumption, as Hadley states.

Thus the cost associated with every shortage at store i is:-

$$\begin{cases} c_x + c_{s_i} \ell_x & \text{if } t' \geqslant \dfrac{c_x}{c_{s_i}} + \ell_x \\ c_{s_i} t' & \text{if } t' < \dfrac{c_x}{c_{s_i}} + \ell_x \end{cases}$$

Thus the expression for $\hat{c}_{s_i}$ will be less than that given by (3).

3.5.3.1 Criticism of expressions (5), (6), (7)

The expression (5) would give the expected shortage occurring in the time period between allocations correctly only for $\xi \geqslant 0$.

In the case of $\xi < 0$, the expected extra shortages in this time period would be:-

$$\sum_{d=0}^{\infty} d p_{ba_i}(d)$$

However, both expressions ignore the dependence of $p_{ba_i}(d)$ on ξ. Clearly if ξ is very low (negative), then the time interval between allocations must be correspondingly low and so it follows that the summation of (6) cannot be used to compute a general expression for the expected (extra) shortages between allocations.

Further, this author feels that in expressions (7), (9), (10), (11), Hadley has mistakenly written c_{s_i} for $\hat{c}_{s_i}$. The inequality (12) seems to be in order.

3.5.3.2 Criticism of the expressions (9), (10), (11), (12)

Little support is given for taking s_i as an approximation for the physical stock at store i after an allocation. Clearly there will be many cases where this virtual stock is either negative or greater than s_i in which case equation (9) will not hold. This means (10), (11), (12) are not strictly correct.

3.5.3.3 Why not incorporate a central store?

Such a base stock system keeps down the overall inventory commitment; this is likely to be a good idea for those cases where inventory holding costs are high. There are numerous shipments in this model, arising from the very nature of a base-stock system and also because of inter-store redistributions. The latter arise because of the length of the procurement lead time.

Completely discounted, however, is the feasibility of holding some stock in a central store from which a shipment instead of a redistribution could take place. The cost of extra stock holding involved may be more than offset by the saving of redistributions.

If Q items were suitably shipped to the central store at a cost of c_P and then delivered in unit amounts to the stores, (at a cost of c_R) the cost of this policy per unit time is

(i) approximately hPQ in stock holding:

(ii) $\sum_{i=1}^{N} \lambda_i c_{R_i} + \frac{\lambda_T}{Q} c_P - \sum_{i=1}^{N} \lambda_i c_{Y_i}$, in delivery costs (may be a saving if $c_{Y_i} > c_{R_i}$):

(iii) extra shortage costs by shipment from central store instead of redistribution (may be negligible or even a saving).

Savings equal to the total expected cost of redistribution would result.

In the above, c_{R_i} is the cost of a delivery from central store to store i and c_{Y_i} is the cost of a delivery from source to store i.

3.6 The Work of M. Shakun and Comments

3.6.1 Summary

In a paper published in The Journal of Industrial Engineering[13] Shakun compares two types of inventory control for a multi-store complex. He classifies the two types as (i) "Independent", wherein each store orders according to its own economic order quantities and reorder levels, these being set independently of each other, and (ii) "System-Wide", wherein a "System Economic Order" is triggered by the total stock in the complex, and a decision is made as to how to split this order up into the allocations to the various stores. No central store is incorporated, and so rules for inter-store shipments to balance out the store stocks as stock-out approaches are developed. The paper reaches the conclusion that considerable savings are to be gained by use of a policy of "System-Wide" control.

3.6.2 Comments

The work of Shakun has not been covered other than superficially because it is felt that it is not a particularly valuable contribution to the literature on the central store/sub-store system inventory control.

This is not to say that the paper does not clearly show a saving in costs by "System-Wide" operation over independent store operation. It is such a result that has led on to the present author presenting this thesis for governing the central store/sub-store inventory problem other than by independent sub-store operation.

It is felt that by incorporating a central store within Shakun's complex, substantially better control is possible. Shakun already makes the point that "it is better to delay the allocation until near the end of the lead period". Why not delay the allocation until as late as possible, until a store reorder level is met? This will entail, of course, keeping part of the system procurement in a store (i.e. the central store) to be called upon when necessary. Further, it is to be recognised that inter-sub-store shipments are costly to the system if the cost per item of shipment is significant. This is because they are extra costs as opposed to shipment from a central store, where only the fixed cost is relevant (the per unit variable cost is irrelevant since this cost must necessarily be incurred, if not now, then in the future). No indication is given as to how the system reorder point or individual store reorder levels are established. Neither is it clear whether the system "Economic Order Quantity" takes into account the cost of shortages or not. Further, it is not clear at all from which store and with what quantity, a store dropping to its reorder level will be replenished.

It appears wrong to consider a redistribution to a store when this store will shortly be replenished from a procurement. ~~that redistribution to a store is always considered right up until the time that store will receive stock from the procurement which is on order~~. This might lead one to assume that the lead time for inter-store shipments is zero. However, although no figure for the lead time is given, it cannot be zero, otherwise there would be no point in considering redistribution until a backorder were experienced. If the redistribution lead time is t_r, then there is no point in considering redistribution for those store reorder points which occur when the time until the procurement will arrive at the store is less than t_r. The analysis appears in error in this respect.

3.7 The Work of W. Lampkin

3.7.1 Introduction

In 1963, W. Lampkin presented his thesis[3] for the M.Sc. degree in the University of London. ~~A general introduction to stock control theory is given, and in Chapter Two, the literature appertaining to inventory control in a hierarchy of stores is reviewed.~~ In the third chapter eight control procedures for the central store/sub-store problem are considered, and it is this work which is particularly pertinent to this thesis.

Lampkin commences his discussion for the problem by considering an (s,S) system at each sub-store and an (s,S) system at the central store. He suggests that this simple scheme might prove difficult to improve on in practice.

3.7.2 Suggestions for control policy

Some suggestions of ideas intuitively likely to cause improvements over the above control policy are ~~now~~ put forward.

3.7.2.1 Suggestion 1: Use of a cyclical review system at sub-stores.

An (s,S) control system at the sub-stores will cause bunching of orders, and hence a high variance of orders on the central store. Hence the central store would have to hold high levels of safety stock if it were to meet most orders. This problem would be got over if a cyclical review control system at sub-stores (with ~~the~~ equal review periods ~~equally spaced~~ at each sub-store) were used. The orders at the central store would be much smoothed with a corresponding reduction in safety stock for the same service level to sub-stores.

Specifically, the suggestion is to review store i at the time instants $t\times(i/N + r)$ where $r = (0,1,2 \ldots)$, N is the number of sub-stores, ~~i = (1,2, ..., N~~) and t is the review period for sub-stores. It is recognised that the cyclical review system is, in general, inferior to the reorder level system in that sub-stores would require to hold more stock to give the same protection against shortage.

3.7.2.2 Suggestion 2: Use of an (s,S,t) system at sub-stores.

Perhaps a reasonable compromise between the advantages of an (s,S) system and a cyclical review system can be made by using an (s,S,t) system at the sub-stores. ~~It is suggested that this suggestion might compromise the problem which Suggestion 1 faces.~~

3.7.2.3 Suggestion 3: Schedule sub-store reviews so that demand in the lead time is reduced.

It is supposed that the central store works on an (s,S) system and the sub-stores on a cyclical review system according to Suggestion 1. Suppose that the central store procurement lead time L is almost a multiplte of (t/N), say $L \approx m(t/N)$.

Now since a central store order must be triggered by a sub-store order, then if $LN/t > m$, we have m sub-store reviews in the lead time whereas if

$LN/t < m$, we have $(m-1)$ reviews. Thus it may be possible to reduce the number of reviews at sub-stores in the lead time by a small change in t.

3.7.2.4 Suggestion 4: Trigger the central store ordering by the total stock in the whole system.

If knowledge of the sub-store stocks is available, the above suggestion is a simple way of utilizing this information.

3.7.2.5 Suggestion 5: Keep a running account of the probability that the central store virtual stock will be below zero in time L, and let the central store order when this probability drops to a certain level.

This is said to be equivalent to the central store ordering at "comparable" stock positions for the sub-stores.

3.7.3 Model and operating rules

The above suggestions were investigated by simulating their application in the following model:

5 Sub-stores.
Sub-store Demands Poisson, Mean Rate 10.
Sub-store Lead Time zero.
Central Store Lead Time .4.
Captive demands at both levels.
Shortage costs at sub-stores proportional to backup.
Holding costs proportional to average stock.
Central store ordering costs proportional to number of orders.

The following operating rules were stipulated.

(1) No inter-sub-store deliveries.
(2) If the quantity in a central store is less than the sub-store order, the amount available is shipped, the balance remaining captive, and delivered as soon as stock becomes available. The latter delivery is termed 'extra' (as opposed to 'normal') and there are different associated costs.

3.7.4 Restrictions on policies

The point is made that, apart from Suggestion 3, the suggestions given do not involve mention of the order quantities, either of the central store, or of the sub-stores. It is reasoned that these quantities will be, in the main, governed by the costs of ordering and delivering, respectively, and so should be about the same for each control system based on the different suggestions.

It was then decided to restrict attention for the specified model to policies with a fixed average order quantity of 30 for the central store and 5 for the sub-stores.

A cyclical review system at sub-stores will generally result in overshoot at the central store, and so some additional arrangement was required in order to make the systems comparable, since now it was not always possible to ensure an average order of 30. It was argued that for the results of comparison to be meaningful, it would not be sensible for one control system to show up best for certain values of the cost parameters only because its average central store order quantity was nearer to optimum than that of the other system. It was suggested that the problem be overcome by fixing the central store order cost such that the optimum reorder quantity is about 30. This suggestion was supported by the fact that the cost curves are flat near the optimum, so small differences in the average order quantities for the central store have negligible consequences.

3.7.5 Approach to the problem

The following policies were simulated for the central store/sub-store complex.

Policy Type	Central Store Policy	Sub-store Policy
A	(s,S) system, S-s = 30	(s,S) system, S-s = 5
B	(s,S) system, S-s = 30	Cyclical Review System t = .4965
C	(s,S) system, S-s = 30	Cyclical Review System t = .5001
D	(s,S) system, S-s = 30	(s,S,t) System with S-s = 4, t = .2001
E	(s,S) system worked on total complex stock; S-s = 30	(s,S) System, S-s = 5
F	(s,S) system worked on total complex stock; S-s = 30	Cyclical Review System t = .4965
G	(s,S,t) system taking advantage of precise scheduling; orders triggered on total complex stock	Cyclical Review System t = .5001
H	Continuous review; "Order when probability of central store stockout time L hence reaches trigger level"	(s,S) system, S-s = 5

Each policy-type was simulated for a period of at least 300 years and the following were observed:-

(a) the distribution of stock in the central store;

(b) the distribution of stock in the sub-stores;

(c) the number of normal deliveries;

(d) the number of extra deliveries;

(e) the average size of a central store order;

(f) the average size of a sub-store order;

(g) the distribution of central store stock just prior to delivery.

(g) is designed to see if Suggestions 1 and 5 have their desired effect.

3.7.6 Average reorder quantities for a (s,S,t) system with Poisson demands

Some of the policy-types to be simulated have cyclical review or (s,S,t) systems controlling sub-stores. It is necessary to be able to predict the average reorder quantity from the s,S,t levels in order that it can be made equal to 5.

If sub-store demand rate is λ_i, then the expected size of the next order from a sub-store is found to be:-

$$\overline{Q} = \sum_{n=0}^{\infty} \lambda_i t \{1 - \sum_{i=q}^{\infty} \frac{(n\lambda_i t)^i}{i!} e^{-n\lambda_i t}\}$$

where q is the nominal reorder quantity (S-s).

The values of t for which $\overline{Q} = 5$ were obtained from this expression by a plot of $\overline{Q}$ versus $\lambda_i t$ and interpolation. The results are given below:-

S-s	t
1	.4965
2	.4758
3	.3982
4	.2001
5	0

3.7.7 Discussion of the policy types

3.7.7.1 Policy Type-A

This is the method of control described as an 'obvious first thought'.

3.7.7.2 Policy-Type B

This is an adoption of Suggestion 1 - a cyclical review system at sub-stores. The reviews at sub-stores are .4965 time units apart. Since there are 5 sub-stores, a central store receives an order from a sub-store every .0993 time units. During each central store lead time, there will be four orders on the central store.

3.7.7.3 Policy-Type C

This is an adoption of Suggestion 3. By increasing the spacing of reviews to .5001, the average order size of sub-stores is negligibly changed, yet now only three orders occur on the central store during its supply lead time.

3.7.7.4 Policy-Type D

This is a dual adoption of Suggestion 2 and of Suggestion 3. There are three possible (s,S,t) systems for the sub-stores (see the (S-s) vs. t table above) intermediate between policies of reorder level and cyclical review. The combination (S-s = 4, t = .2001) was chosen since:

(i) .2001 was the most nearly intermediate between zero and .4965:

(ii) the value, t=.2001, enabled the precise scheduling of Suggestion 3 to be incorporated.

3.7.7.5 Policy-Type E

This is the application of Suggestion 4.

3.7.7.6 Policy-Type F

This is an application of Suggestions 1 and 4.

3.7.7.7 Policy-Type G

This is an application of Suggestions 1,3,4. Central store orders are triggered by the total stock in the stores complex according to an (s,S,t) policy where the review instants are such that they occur just after each sub-store review. In this way stock will have just arrived in the central store when a sub-store orders.

3.7.7.8 Policy-Type H

This is an attempt at adopting Suggestion 5, with a reorder level system at sub-stores. Simulation time was deemed too much for calculations after each demand to obtain the probability of the central store stock dropping below zero in the following time L (= .4 time units). The following alternative suggestion was adopted.

Each sub-store has five possible notional stock positions 0, 1, 2, 3, 4. Each of these positions has associated with it an expected number of orders by that sub-store in the following time L and also a variance about that mean.

After each sub-store demand, the mean and variance of the total orders on the central store in the following central store lead time was calculable by use of the means and variances of the individual sub-stores' orders. The ordering rule adopted was:-

"Order if 5 x Mean Number of Orders + 5 x k x Standard Deviation of Number of Orders is greater than the Notional Stock at Central Store".

The parameter k was varied between -0.7 and 2.80 in steps of 0.35.

3.7.8 Simulation results

3.7.8.1 Variance of central store stock just prior to delivery

It is shown that, in respect of variance of virtual stock at the central store just before a supply of stock to this store, those policy-types incorporating cyclical review at sub-stores are best. Additionally, of these, those using the total stock to trigger the central store order (viz., policies F,G) are best. The same is true of the policy-types using a reorder level policy at sub-stores. The more complicated policy-type H is not better than E.

It thus appears that Suggestion 1 has some merits. ~~whilst Suggestion 2 does not work at all~~.

3.7.8.2 Service given by central store

Three criteria of this measure are considered:-

(i) average central store backup*

(ii) proportion of sub-store orders not met completely

(iii) proportion of sub-store orders not met at all.

Each criterion of measure is plotted against the total cost of stock holding and ordering at the central store. Policy-types F,G are almost identical in performance (i) whilst policy-type D is worst, corroborating opinion on Suggestion 2.

By use of criterion (ii) policy-type G is best, with C runner-up. F performs badly. Criterion (iii) has G best with C runner-up.

* time-averaged number of units ordered on central store and not shipped at once.

3.7.8.3 Costs of the best policies

Lampkin tabulates the annual costs of operation for the optimal combination of parameters (the latter being reorder level at sub-store and the central store control parameter). The cost of ten deliveries (since all policy-types are committed to at least ten deliveries per annum) is deducted throughout.

The results clearly show that all the policy-types incorporating cyclical review at the sub-stores are inferior to the worst of the reorder level at sub-store policy-types.

The policy-types with a reorder level policy at sub-stores have two advantages. Firstly, better sub-store performance results and, secondly, stocks can be arranged so that the central store always contains a multiple of 5. In this way it is said that the delivery totals can be minimized; either there is a normal delivery of five units and no extra delivery is necessary, or no stock is available, whence the cost of a normal delivery is saved.

Lampkin expects, then, that the reorder level based policy-type to show up best when:-

(a) cost of shortages at sub-stores is high:

(b) extra deliveries cost no more than normal.

Two examples are chosen, one when (a), (b) hold true, the other where the converse is true. Even in this second example, it is shown that the reorder level-based policy-types work better. The advantage is seen to be in the costs of stockholding and shortages. The conclusion is drawn that Suggestion 1 is of no value.

Also noted is the fact that for many cost combination performances, the second advantage of the reorder level based policy-type (viz. the ability to arrange stocks so that the central store always contains a multiple of 5) is not utilized in many cases. The reason given is that holding one unit extra over the multiple of 5 enables the central store to supply a needy sub-store. When the central store runs out, the sub-stores have to manage until the next supply arrives at the central store. The latter time is generally short and often just one unit would be enough to save a shortage. It is shown that even where delivery costs are high, it is often worthwhile having extra deliveries to prevent shortage.

It is suggested that a rule to ration stock when the central store is low might prove a good idea. In concluding the discussion of the results, Lampkin puts forward his view that, whilst not being able to provide information regarding the reliability of the figures in his table of results, policy-type H is best, and probably could further be improved by allowing central store stocks to be other than multiples of five.

3.7.9 Conclusions

Lampkin concludes from his work the following points related to his specific suggestions:-

1. The use of evenly spaced cyclical review policies at sub-stores, although reducing the variance of demands on sub-stores, is inferior due to the basic inferiority of cyclical review compared with reorder level based policy.

2. The 'compromising' use of an (s,S,t) system at sub-stores with evenly spaced reviews does not reduce the central store orders' variance much. It is said, however, to be better than cyclical review.

3. The use of precise scheduling seems valuable at first sight, but for practical applications, seems difficult to utilize.

4. The use of total stock for central store trigger seems valuable.

5. The use of a policy-type triggering central store orders by a target level for the probability of the central store stock being depleted time L in the future seems slightly better than triggering the complex order on total system stock. However, this conclusion is not stated with complete confidence.

6. It is not usually a good idea to arrange central store stock to be a multiple of sub-store order; one or two units above the multiple is a more sensible idea.

3.7.10 Comments on the work of Lampkin

This author feels that the work of Lampkin is a ~~useful~~ valuable contribution to the literature on the problem of controlling a central store/sub-store complex. ~~It is very useful for~~ Operational Researchers need to know the effect of different policies to control such a complex, and with Lampkin having done the basic research work for a wide range of policy-types and cost combinations here is a good guide for the choice of a policy-type to suit a particular need. However, the practical applications do appear to be somewhat restricted by the assumptions, especially the one specifying zero lead time. It is felt that a justification of the assumptions and an attempt to indicate their non-engendering of speciality for overall practical application would have been worthwhile. It is, of course, recognised that the assumption of no inter-sub-store replenishments was necessary - otherwise much more work would be entailed to establish a more realstic model.

All the suggestions presented initially by Lampkin appear feasible, and it is only simulation that can show up the performance of the complex under control of the different policy-types. It might have been useful to apply some significance tests to the results.

Most important of all, this author feels, is that the simulation results show the overwhelming superiority of policy-types E, H. Both incorporate a reorder level policy at sub-stores and E has the order for the complex (procurement) trigger on total complex stock whilst H tries to utilize more information regarding sub-store stocks. Thus this points the way to more ~~useful~~ sensible criteria for triggering the complex order.

Lampkin in concluding his thesis, suggests that the order trigger should take into account the sub-store individual stocks. Clearly this is the more sensible thing to do, when the information regarding the latter is available centrally. When it is not, one can fall back on one of the other policy-types. When this information is held centrally, even greater improvement is available over E and H. The size of the order to the sub-store can now not necessarily be constant, as in E, and H; rather, it can vary to take into account the stock positions of all stores in the system. This is just what Lampkin advocates - rationing stocks to stores in difficult times (when not much stock is available over the whole stores complex).

3.8 The Work of K.F. Simpson, Jr.

3.8.1 Summary

In a paper to Operations Research, Simpson[15] considers the problem of allocating a procurement to a series of stores. A further procurement is expected to arrive at a known time in the future. The demand statistics at the stores are known. Simpson proposes a model for two management objectives and gives, with a proof, a theorem of allocation for each case. No central store is assumed, but the decisions are made by some central agency.

3.8.2 Simpson's model

The allocation to be made from the procurement Q must last a known time T for the case of a cyclical reorder scheme. For reorder level systems, the time T will not be known, but Simpson assumes it can be estimated sufficiently accurately from demand forecasts. The quantity to be allocated is assumed given.

3.8.3 The first management objective considered. The case of "Emergency Replenishment"

This objective is the case of replenishing a sub-store when it reaches an emergency reorder level, set high enough so that shortage at a store is seldom experienced.

The sources of this "emergency replenishment" considered are:-

(i) original source:

(ii) another store with surplus stock.

Whatever the source, the cost of the emergency replenishment procedure is said to be approximately proportional to the number of emergency replenishments. Simpson's model assumes exact proportionality.

Further, Simpson assumes that the initial allocation is always sufficiently high that a particular store will not require to be "emergency replenished" more than once in any given "replenishment period".* Then if the cost of emergency replenishment is c_{R_i} and the initial allocation to store i is A_i, then the probability of an emergency replenishment is denoted by $P(d_i \geq A_i - r_i)$ where r_i is the reorder level for store i and d_i is the demand on store i before stock from the new procurement can arrive there.

The costs then per replenishment period are in total

$$\sum_{i=1}^{N} c_{R_i} \; P(d_i \geq A_i - r_i) \qquad (1)$$

for the N stores.

3.8.3.1 The optimal allocation policy for the "Emergency Replenishment" case

The above will be found from the minimization of (1) subject to $A_i \geq 0$,

$$\sum_{i=1}^{N} A_i = Q.$$

The following theorem, Theorem 1, proved in an appendix, is given: "The optimal allocation requires that $c_{R_i} \; P(d_i = A_i - r_i)$ is equal for each store."

---

* i.e. before it gets its next "normal replenishment" or allocation.

3.8.4 The second management objective. The case of "No Emergency Replenishments"

This case deals with a policy of having no reorder level and allowing stores to incur shortages without any action being taken until the next procurement is available. The shortage penalty c_{S_i} is assumed time-independent. If the density function of demand at the i^{th} store is $g_i(d')$ then the expected cost of shortages for store i is

$$\int_{A_i}^{\infty} c_{S_i}(d' - A_i)\, g_i(d')\, dd'$$

whence the total cost of shortages is:-

$$\sum_{i=1}^{N} \int_{A_i}^{\infty} c_{S_i}(d' - A_i)\, g_i(d')\, dd' \qquad (2)$$

3.8.4.1 The optimal allocation policy for the "No Emergency Replenishment" case

The problem of establishing the A_i such that expression (2) is minimized is said to be analogous to the previous problem.

The following theorem, Theorem 2, is given:

"The optimal allocation requires that the weighted probabilities $c_{S_i} P(d_i \geq A_i)$ be equal for all stores."

3.8.4.2 The case of equal c_{S_i}

If all the c_{S_i} are equal, then the optimal allocation policy is such that the probability of a demand on any store being greater or equal to the amount allocated is the same for all stores. Distinction is made between this result of equalising the distribution function $P(d_i \geq A_i)$ and the result of equalising the probability density function for the case of emergency replenishment (for the case when the reorder levels r_i are all zero and the c_{R_i} are identical).

3.8.5 Applications of given theorems

Equalisation of probability density functions is admitted to be impractical for the case of large numbers of items. It is said that forecast errors at different stores, normalized by dividing by the standard deviation of forecast errors, frequently have the same distribution for all stores. When three conditions are met, the theorems have greater applicability. These conditions relate to the density function f(x) of the common distribution referred to above. They are:-

(1) $f(x)$ decreases in $(0,\infty)$:

(2) $f(x)$ increases in $(-\infty,0)$:

(3) $f(0) \leqslant \frac{1}{2}$.

The normal distribution satisfies these conditions.

3.8.6 The "Emergency Replenish" case

The case where the common distribution $f(x)$ of forecast errors meets conditions (1) to (3) inclusive is considered.

The probability of a demand between A_i and $(A+dA_i)$ at store i is:

$$(1/\sigma_i)\ f\{(A_i-F_i)/\sigma_i\}\ \ dA_i$$

where F_i is the forecast of demand at store i in the period T and σ_i is the standard deviation of past forecast errors.

The function $f(x)$:

$$g(x) = \begin{cases} f(x) & \text{for} \quad x \geqslant 0 \\ 1-f(x) & \text{for} \quad x < 0 \end{cases},$$

is defined and the conditions (1) to (3) imply $f(x)$ to be decreasing over $(-\infty,\infty)$.

Theorem 1 is said to be extendable to show that equating $(c_{R_i}/\sigma_i)g\{(A_i-F_i)/\sigma_i\}$ for all stores is both a necessary and sufficient condition for the allocation resulting to be optimal. The computation of the allocation quantities by an iterative method is said now to lie within the reaches of electronic computation.

3.8.7 No-Emergency-Replenishment Case

By similar means to the foregoing case, where the c_{S_i} are equal, a common forecast error distribution is all that is needed to reduce Theorem 2 to the solution:

$$A_i = \left((Q - \Sigma F_i)/\Sigma\sigma_i\right) \sigma_i + F_i$$

(which is the same result for the Emergency Replenishment Case for the special circumstances where c_{R_i}/σ_i are nearly equal for all i).

3.8.8 Comments on the work of K.F. Simpson, Jr.

The above-described work is really only peripheral to the problem in hand, in as much as it only deals with the very specialised problem of how to allocate an amount of stock, once very important decisions have already been made. These decisions are that either the reorder level of stores will be r_i or that stores will be allowed to experience shortage without replenishment to them taking place. Even with these pre-established policies it is not made clear what quantity to give to a store (in the case of emergency-replenishment). This is of no little significance if holding cost is itself not unimportant in the case that the stock comes from the original supplier.* In the case of supply from another store, there is again a substantial problem of deciding on the quantity, since the store from which the stock comes may be jeopardised by too high a quantity shipped out of it. Too little stock shipped may mean that further distribution is necessary. To state that only one (at most) emergency shipment per replenishment period for any given store will occur does seem rather a restrictive assumption since actual demand may well exceed forecast demand.

Where the cost of stock holding is minimal and where stock for the emergency distribution comes from the supplier at short notice, Simpson's work has some applicability. It is clear, however, that it may be expedient to hold some stock back instead of distribute out all the procurement initially. This stock may be held either at a central store or at some other store.

To summarise, this author feels that Simpson is establishing control of a too-well specified system. The fact that the arrival time for the next procurement is assumed to be known means that the buffer stock for the system is already specified. This may well be too high. The r_i may be either too high (such that a replenishment which might not really be needed could be saved) or too low, so shortage occurs in the lead time.

The present author tackles essentially the same problem of how to allocate a procurement amongst sub-stores and although the control necessarily prohibits inter-sub-store replenishment and requires a central store, and the allocation formula is in no sense optimal, it does try to establish reorder levels, quantities distributed, when to order a procurement, etc., not from management objectives, but from an overall view of attempting to optimise the combined procedures in control of the stores complex. All these considerations are interrelated and should surely be considered as such.

* Or perhaps the quantity supplied is assumed a fixed quantity, and the cost associated with the replenishment takes into account an average extra stock holding cost. This will not strictly be correct since extra holding cost would necessarily, for a particular instance of a delivery, have to relate to the expected extra holding time, and thus the holding cost would differ appreciably for different delivery decisions.

3.9 The Work of S.G. Allen

3.9.1 General

This work was published[17] in 1958. The complex that is considered consists of N stores, each of which holds stock, and no more stock is available to replenish this complex until time T.

At time zero stock may be redistributed over the stores as wished, with cost proportional to total number of units of stock moved. The objective function is the total cost of redistribution and shortages during the period T. This function is required to be minimized. What occurs beyond T is ignored for the purposes of this minimization.

3.9.2 Notation

Cost of moving 1 unit from store i to store j is c_{ij}.

Probability that demand does not exceed Γ in time T for store i is $P_i(\Gamma)$.

Each shortage costs c'_S.

3.9.3 Analysis

Total redistribution cost is:-

$$C_{RT} = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij};$$

where x_{ij} is the redistribution quantity from store i to store j.

Total expected shortage cost is:-

$$C_{ST} = c'_S \sum_{i=1}^{N} \int_{s_i}^{\infty} (\Gamma - s_i)\, dP_i(\Gamma)$$

where s_i is the stock at store i after the redistribution.

The objective function is thus $C_{RT}+C_{ST}$, which is to be minimized subject to:-

$$s_i = s_i^{o} - \sum_{j=1}^{N} x_{ij} + \sum_{j=1}^{N} x_{ji}, \quad (i = 1,2\ldots N) \qquad (1)$$

where s_i^{o} is the stock at store i before redistribution.

The further restriction, $x_{ij} \geqslant 0$, (2)
is given.

It is stated that the assumption,

$$c_{ij} < c_{ik} + c_{kj} \qquad (3),$$

(which imples that it is always cheaper to ship directly from a store i to j than by any other indirect route) is sufficient to prevent the need for additional restrictions like:-

$$\sum_{j=1}^{N} x_{ij} \leq s_k^{o}$$

since this is implied by equations (1) through (8).

If the value of x_{ij}, which minimises $TC = C_{RT}+C_{ST}$ is $\hat{x}_{ij}$, either

$$\left.\frac{\partial(TC)}{\partial x_{ij}}\right|_{\hat{x}_{ij}} = 0$$

or $$\left.\frac{\partial(TC)}{\partial x_{ij}}\right|_{\hat{x}_{ij}} > 0 \text{ and } \hat{x}_{ij} = 0$$

where $\partial(TC)/\partial x_{ij} = c_s'\{P_j(s_j) - P_i(s_i)\} + c_{ij}$

is satisfied for each pair (i,j).

It follows that if:-

$$1-P_i(s_i^{o}) \geq 1-P_j(s_j^{o})$$

there cannot be any transfer from i to j. This means that redistribution will only occur when the receiving store has a higher probability of shortage than the shipping store.

3.9.4 Computational procedure

The suggested method is to vary just one of the x_{ij} (with the others held constant) until a conditional minimum is attained. This procedure is then repeated for the remaining x_{ij} in cycles until a convergence criterion is satisfied.

3.9.5 Author's comments on the work of S.G. Allen

As this is essentially a publication peripheral to the work in this thesis, comments will be brief.

There are two basic criticisms. Firstly, there is the non-provision of a further chance for redistribution within the period T should a store's stocks get to a level where the shortage cost can clearly be reduced by such a redistribution.

Secondly, there is the possibility that the total cost of operation of the stores complex can be reduced by incorporating a central store which receives the procurement and ships some stock to stores out of each newly arrived procurement. Further distribution from the central store or redistribution could then occur at strategic times in the time period T. As a result of the implementation of this central store, the total cost of shortages and deliveries might well be reduced to a figure below the optimum corresponding to the specific control considered by Allen. There is the further point that the value of T is itself a restriction. Relaxation of this restriction could itself lead to a cheaper cost of operation overall.

3.10 The Work of S.A. Bessler

3.10.1 Summary

Bessler[19] considers the case of a "Polaris type" multi-echelon supply complex consisting of a central store and N sub-stores. Replenishment of the central store and the sub-stores is periodic. Emergency replenishments also occur to a sub-store whenever a backorder occurs at that sub-store. No inter-sub-store shipments are allowed.

The optimal multi-echelon inventory procedure is a vector $\bar{y}$, such that at the order time, the i^{th} sub-store orders that amount of stock to bring its notional stock to $\bar{y}_i$ (the latter being the i^{th} component of $\bar{y}$). Bessler constructs an algorithm with which to obtain $\bar{y}$ for a complex with a central store and three sub-stores. He goes on to compute optimal inventory procedures for various data combinations (unit cost, shortage cost, etc.).

3.10.2 Introduction

An item has a unit cost P and the cost of an emergency replenishment is c_{R_e}. If the central store is unable to supply the sub-store (because it is depleted) when an emergency replenishment is ordered, a cost C_1 results. The case considered is:-

$$P/c_{R_e} = 10; \quad C_1/c_{R_e} = 15;$$

where the cash discount factor is β per period and where the demand comes from nine sources for sub-store 1, six sources for sub-store 2, and three sources for sub-store 3. The source demand pattern follows the binomial distribution where the probability of demand for a unit in the interval between replenishments is .01. The result of the application of Bessler's algorithm gives $\bar{y} = (1,1,0,0)$, i.e. $\bar{y}_o = 1$ (i.e. order to a level 1 for central store), and $\bar{y}_1 = 1$, $\bar{y}_2 = \bar{y}_3 = 0$.

3.10.3 The model

Let notional stocks for the central store and the N sub-stores be denoted by $x_t = (x_{ot}, x_{1t}, \ldots, x_{Nt})$ at the general instant when ordering is taking place, say the beginning of the t^{th} period (i.e. the suffix zero refers to the central store). Notional stocks after the order will be $y_t = (y_{ot}, y_{1t}, \ldots, y_{Nt})$.

The actual demand in the t^{th} period over the N sub-stores will be some vector $d_t = (d_{1t}, \ldots, d_{Nt})$. This demand is assumed to be identically distributed between periods with a joint probability function of $p(d_1, d_2, \ldots d_N)$.

Now let the vector of stock levels at the end of period t be $s(y_t, d_t)$.

Let $s_{it} = \text{Max}\left[(y_{it} - d_{it}), 0\right]$

and $s'_{it} = \text{Max}\left[(d_{it} - y_{it}), 0\right]$, $i = 1, 2, \ldots, N$.

Then if the i^{th} component of $s(y_t, d_t)$ is $s_i(y_t, d_t)$, we have:-

$$s_o(y_t, d_t) = y_o - \sum_{i=1}^{N} s'_{it}$$

$$s_i(y_t, d_t) = s_{it}, \qquad i = 1, 2, \ldots, N.$$

This assumes all demands to be captive.

3.10.4 The cost function

Costs incurred during the t^{th} period consist of a "regular" replenishment cost, "emergency" replenishment cost and shortage costs.

3.10.4.1 "Regular" replenishment costs

The costs associated with the regular replenishment at the beginning of the t^{th} period, quantity $(y_t - x_t)$, are

$$P \times \sum_{i=0}^{N} (y_{it} - x_{it}).$$

Thus, it is seen that P is strictly the delivered cost of the item. (Production set-up costs are proportional to the number of units produced and are included in P.)

3.10.4.2 "Emergency" replenishment costs

These are $c_{R_e} \times \text{Min}\left(\sum_{i=1}^{N} s'_{it}, y_{ot} \right)$.

3.10.4.3 Shortage costs

These are $c_1 \times \text{Max}\left(\sum_{i=1}^{N} s'_{it} - y_{ot}, 0 \right)$

The sum of the emergency replenishment and shortage costs for the t^{th} period will be denoted by $g(y_t, d_t)$.

3.10.5 Ordering policies

The information available at the beginning of the t^{th} period is $H_t = (x_i, x_2, \ldots, x_t;\ y_1, y_2, \ldots y_{t-1};\ d_1, d_2, \ldots d_{t-1})$. The ordering policy $Y = (Y_1, Y_2, \ldots)$ is a sequence of vectors. Y_t is defined on H_t, i.e. $Y_t = Y_t(H_t)$ and $(Y_t - X_t)$ will be the order vector for the central store and sub-stores at the start of period t.

Clearly, we are concerned with those policies such that $Y_t \geq X_t$. Such policies are termed feasible.

3.10.6 Method of analysis

Let $W(y,d) = g(y,d) - \beta c_{R_e} s(y,d)$. Suppose $f_t(x_1/Y)$ denotes the expected discount cost during periods 1, 2, ... t, when the initial inventory is x_1 and the feasible policy Y is followed. Then we have:

$$f_t(x_1/Y) = \varepsilon\{\sum_{i=1}^{t} \beta^i \left(c_{R_e}(y_i - x_i) + g(y_i, d_i) - \beta^{t+1} c_{R_e} x_{t+1}\right) .$$

The term $\beta^{t+1} c_{R_e} x_{t+1}$ represents the disposal value for items left after t periods (i.e., the cost of satisfying the demand remaining after completion of the t^{th} period). Bessler goes on to show that $f(x_1/Y)$, defined by

$$\lim_{t\to\infty} f_t(x_1/Y) = \sum_{i=1}^{\infty} \beta^i \varepsilon\left(G(y_i)\right), \quad \text{where} \quad G(y) = c_{R_e} y + \varepsilon\left(W(y,d)\right) ,$$

exists for all Y.

The problem is now reduced to finding a policy Y* such that

$$f(x_1/Y^*) = \min_{Y \varepsilon \hat{y}} f(x_1/Y)$$

where $\hat{y}$ is the class of feasible policies. If such a policy exists it is termed "optimal".

3.10.7 The optimal procedure

Bessler now proceeds to show that an optimal policy of particularly simple form exists for the problem described. If there exists a vector $\bar{y} = (\bar{y}_o, \bar{y}_1, \ldots \bar{y}_N)$ minimizing G(y) over the set Y_1, then the optimal policy is:-

$$Y_t^* (H_t^*) = \bar{y}$$

where H_t^* is the stock information available up to period t when following the optimal policy Y^*. Thus for a complex of N sub-stores, the optimal policy is defined by a critical vector $\bar{y} = (\bar{y}_o, \bar{y}_1, \ldots \bar{y}_N)$. Then if at the beginning of an order period the notional stock x_i is less than $\bar{y}_i$ at sub-store i, an order for $(\bar{y}_i - x_i)$ is placed. For all stores we then have a total ordered quantity of

$$\sum_{i=0}^{N} (\bar{y}_i - x_i).$$

Thus it must be shown that there exists a vector $y \varepsilon Y_1$ for which

$$G(y) = c_{R_e} y + \varepsilon\{g(y,d) - \beta c_{R_e} s(y,d)\}$$

is minimized. A theorem is given to establish this point.

The problem of determining an optimal policy has now been translated into one of minimizing a function of the single period costs. That is, the vector $\bar{y}$ for which $G(\bar{y}) = \min_{y \varepsilon Y_1} G(y)$ is sought where

$$G(y) = c_{R_e} (1-\beta) \sum_{i=0}^{N} y_i + \varepsilon\left(g(y,d) + c_{R_e} \sum_{i=1}^{N} d_i\right)$$

To minimize $\varepsilon\left(g(y,d)\right)$, $y(= y_o, y_1, \ldots y_N)$ is formulated and solved by an N-stage dynamic programming problem.

3.10.8 Algorithms derived

For the case N = 2 a very simple algorithm is developed. It is paraphrased as follows: "At each stage provide an additional unit to the sub-store for which the return is greater. If there is no advantage gained from a supply of an additional unit to either sub-store, then stop." This algorithm has the feature of substantially reducing computations required to determine y_N^* (the vector corresponding to minimal g(y)).

Bessler goes on to develop a computational procedure for the case N=3. For more than three sub-stores an optimal distribution $\bar{y}$ is obtainable from the general method of solution. It is stated that a very desirable result would be the determination of restrictions to be satisfied to have a simple algorithm similar for the case N=2. Although this problem has been actively considered, no solution is yet available.

3.10.9 Comments on the work of S. Bessler

The way in which Bessler takes into account the holding cost of the item is somewhat interesting. Whenever a sub-store orders a quantity, q say, then the cost of ordering for this sub-store is assumed to be Pq (where P is the delivered cost of the item). Clearly it is not sensible to associate the actual cost of an item with an ordering sub-store. Account should be taken of the cost of the capital tied up with stocks at the ordering sub-store, but this is just a percentage of the actual item cost.

It may be more expedient to trigger an "emergency" replenishment before a sub-store receives a back order. Thus in no sense are Bessler's rules optimal from an overall sense. However apart from the aforegoing criticisms, the "optimal" policy suggested with the assumptions requisite for Bessler's model is likely to be useful for slow-moving items.

3.11 The Work of D. Hoekstra

3.11.1 Introduction

Hoekstra considers[20] a multi-echelon supply system for aircraft engines. The ability to support a given state of readiness of the aircraft population is said to depend on the frequency of engine shortages, the length of time these shortages last and how many aircraft are grounded as a consequence.

Hoekstra tackles the problem from the point of view of "average customer waiting time", waiting time being the time a replacing facility has to wait for a replacement engine after removing an engine due for repair or overhaul from an aircraft.

The model considered is basically a specialized central store/sub-store problem except that the central store may be considered as the repair facility as well as a storage agent. When an item is sent to the central store for repair the sending sub-store is replenished as soon as possible from the central store.

ℓ_i is the lead time for replenishment to sub-store i from the central store* and t_R is the time to repair an engine at the central store (taken as the time period between when the engine leaves the sub-store and the time it is in repaired condition at the sub-store). Both are averaged times. t_r is the average time for repair at a sub-store. ℓ_{Di} is the average time in filling a demand at sub-stores as the result of stock not being available at the central store.

The rate of demand for sub-store i is λ_i in total, of which λ_i' are returned to the central store for repair (instead of repair within the sub-store). The total rate of replenishment orders on the central store will

---

* When the central store is not out of stock.

then be $\sum_i \lambda_i'$, termed λ_{RT}. The amount of stock in the central store (serviceable physical stock plus stock in the repair cycle) is S_M. r_i designates the stock level at sub-store i equalling sub-store repair stock, physical stock, and transit stock from the central store. S_N is the physical stock in the central store less backorders from the sub-stores; the probability of S_N is designated by $p(S_N)$. The stock at sub-store i which is the sum of serviceable stock less backorders is n_i, the probability of which is $p(n_i)$.

The average number of shortages in sub-store i is denoted by b_i and average time to fill a demand is t_w.

3.11.2 A note on central store availability

The mean number of items in the central store repair cycle will be $\lambda_{RT}t_R$. Clearly there will be times when the stock in the repair cycle exceeds $\lambda_{RT}t_R$ to an extent that the central store is depleted. When this occurs, replenishments to sub-stores are delayed.

The probability distribution of "net stock" S_N at the central store is:-

$$p(S_N) = \frac{e^{-\lambda_{RT}t_R}(\lambda_{RT}t_R)^{S_M-S_N}}{(S_M-S_N)!} \tag{1}$$

resulting in the average time to fill a demand when the central store has no stock of:-

$$\ell_{Di} = \frac{1}{\lambda_{RT}} \sum_{S_N=-1}^{-\infty} (-S_N)\, p(S_N) \tag{2}$$

3.11.3 Average time to fill a demand

The average number of items tied up in the pipelines attached to a sub-store i is:-

$$\mu' = (\lambda_i-\lambda_i')\, t_r + \lambda_i' (t_R + \ell_{Di}) , \tag{3}$$

but fluctuates about this because of the Poisson distribution in demands. The inclusion of the factor $\lambda_i'\ell_{Di}$ reflects the fact that the sending of an item for repair from sub-store to central store may not immediately be followed by a replenishment from the latter, since the latter may be depleted of serviceable stock.

Sub-store shortage will always result if the number of items in the pipelines attached to a sub-store i (notional stock plus repair stock) exceeds the level r_i.

The expression for the probability distribution of n_i, sub-store i's serviceable stock less backup, is given as

$$p(n_i) = \frac{e^{-\mu'} \mu'^{(r_i - n_i)}}{(r_i - n_i)!} \tag{4}$$

The average number of backorders in sub-store i is given as

$$b_i = \sum_{n_i=-1}^{-\infty} (-n_i)\, p(n_i) \ , \tag{5}$$

and the average time to fill a demand, averaged over all sub-stores, is

$$t_w = \frac{1}{\sum_i \lambda_i} \sum_i b_i \tag{6}$$

3.11.4 Approach to the solution

One problem which may require solving is that of allocating a given total I items over the complex to yield minimal average time to fill a demand, i.e. find M, r_i, (i = 1,2 ...) such that $t_w(S_M, r_i)$ is minimum, subject to $S_M + \sum_i r_i \leq I$.

Hoekstra considers the following as characteristics of the optimum solution:-

(i) Subtracting one item from any sub-store and adding to another sub-store will not improve the function t_w:

(ii) Subtracting one item from the central store and adding it to a sub-store does not improve t_w, whatever sub-store is considered.

3.11.5 The iterative procedure for the solution

The following iterative procedure is given as one for which convergence to the optimum results:

(i) $S_M = I$; $r_i = 0$ for all i. Compute t_w:

(ii) For k = 1,2,3..., compute t_{w_k} with S_M one less than before and r_k one more than before, leaving r_i unchanged, for all i ≠ k:

(iii) Is $\min_k t_w < t_{w_k}$?

If Yes; put $t_w = t_{w_k}$ and $r_k = r_k + 1$

Return to (ii):

If No; t_w is optimal.

3.11.6 Conclusions/Results

It is found that the decision as to how much stock to allocate to the central store has a distinct effect on the performance characteristics of the complex. Secondly, the stock level S_M being retained by the central store is said to be surprisingly low for optimum complex performance.

It is suggested that it is advantageous for the central store to be considered more as a repair facility than a storage depot, with stocks moving rapidly out of it once repaired. The point is made that for many types of item, the central store represents a relatively small part of the flow of stock. It is conjectured that the high level of protection against shortage at the central store (which characterises most "economic procurement quantity" models in use today) is largely unnecessary.

3.11.7 Comments on the work of Hoekstra

The work of Hoekstra considers a closed self-sustaining complex with no procurement problems, but the distribution of a given inventory investment over the central store and sub-stores. As such, the aid to management of an average time to fill a demand as a function of this inventory management is extremely valuable. The iterative procedure for obtaining the optimal distribution (i.e. of obtaining S_M, r_i) such that the average waiting time to fill a demand is minimised, is remarkably simple.

Although it appears sensible to replenish immediately a sub-store which ships an item to the central store for repair, no suggestion is offered that this procedure is optimal.

For the operations system considered by Hoekstra this paper does seem to represent useful control.

It does appear that equation (4) is not strictly correct. The reason for this is that the effective lead time is a random variable, and its mean $\overline{\mu}'$ given by (3) is used as a constant in equation (4). A more accurate version would be

$$p(n_i) = \sum_{\xi} \frac{e^{-\xi}}{(r_i-n_i)!} \xi^{r_i-n_i} \phi(\xi)$$

where ξ is the effective lead time and its probability density function is $\phi(\xi)$.

3.12 The Work of E. Berman

3.12.1 Introduction

Berman, in a paper published in Operations Research[22], considers the problem of controlling stock in a complex consisting of a number of stores with an initial even stock distribution and subject to a Poisson distribution of demand. No procurement from any central store or elsewhere is considered. Only redistribution between the stores is permitted.

In each time period, the redistribution decision between a pair of stores is made using an analytical solution which takes into account the redistribution cost, the stocks on hand, the cost of shortages, and the demand probability function at the two stores. In addition a control parameter β_t is incorporated representing the value of the redistribution in time periods beyond an artificial time horizon (i.e., it gives the fraction of the redistribution cost charged in the t^{th} period). The expected total cost of operation of the complex with different values of β is then determined by Monte Carlo Simulation. The ideas are extended to cover the case of the redistribution decision among three or more stores.

3.12.2 Establishing the analytical solution to the redistribution decision

The current redistribution decision is assumed responsible for shortages occurring between the time of arrival of stock redistributed now and the time of arrival of stock redistributed in the next decision period. The artificial time horizon for period t is the length of the period of the model plus the redistribution lead time.

Thence, the cost associated with a stock (notional) at store i of s in time period t is:-

$$C_i(s,t) = c'_{s_i} \sum_{x=s+1}^{\infty} (x-s)\, f_{it}(x)$$

where c'_{s_i} is the unit shortage cost* at store i and $f_{it}(x)$ is the probability density function of demand at store i between the present period t and the artificial time horizon for the t^{th} period.

3.12.3 Consideration of two stores

It is argued that if C(s,t) has a greater slope for store i than for store j then the consideration of a stock redistribution can only be from store j to store i, not vice versa.

A decision is to be made to ship a quantity q_r (≥ 0). The costs of this decision are

$$C_A(q_r,t) = C_i(s_i+q_r,t) + C_j(s_j-q_r,t) + \beta_t\left(c_{r_e} q_r + c_{r_f}\right) \quad \text{for } q_r > 0 \qquad (1)$$

* This takes into account the cost of an "emergency redistribution" as well as the actual cost because the item is needed and is not immediately available.

where s_i, s_j are the stocks at stores i,j before redistribution, and c_{r_e}, c_{r_f} are, respectively, the variable (i.e. per unit) and fixed costs of the redistribution.

If $q_r = 0$ (i.e. no redistribution),

$$C_B(q_r,t) = C_i(s_i,t) + c_j(s_j,t) \qquad (2)$$

3.12.4 Method of proceeding to the solution

Berman argues that the function $C_A(q_r,t)$ is either increasing or convex, whence in the former case, the solution is $q_r = 0$. In the case that the function is convex, the q_r minimizing the function $C(q_r,t)$ is obtained, say $\hat{q}_r$, and then $C_A(\hat{q}_r,t)$ is compared with $C_B(0,t)$. If $C_A(\hat{q}_r,t) < C_B(0,t)$ then the decision is a redistribution of size $\hat{q}_r$, otherwise there is no redistribution.

3.12.5 A note on the β_t function

The basic reason for this function is the need to allow for the fact that there exists the distinct possibility that the considered redistribution, if not occurring now, may well be required between the same two stores (and in the same direction) at some later period.

The β_t functions considered are characterised by the initial value of β (i.e. that in period one) and the shape of the function between this period and the last period considered in the model. Clearly β_t functions can be seen to reach a value of unity in the last period.

3.12.6 Extension of the ideas to multiple-store redistribution

Consideration of redistribution from two or more stores to one store is first made. The q_r items considered come successively from those stores which have the least value of $C_i(s,t)$ slope. If three stores are considered shipping to one store, equations (1), (2) are respectively modified thus:

For store 1, cost of a redistribution of q_r to this store,

$$C_A(q_r,t) = C_1(s_1+q_r,t) + C_2(s_2-q_2,t) + C_3(s_3-q_3,t) + C_4(s_4-q_4,t) + \beta_t\left(c_{r_e} q_r + k\, c_{r_f}\right) \qquad (3)$$

where k is the number of stores contributing to the q_r.

$$C_B(q_r,t) = C_1(s_1,t) + C_2(s_2,t) + C_3(s_3,t) + C_4(s_4,t) \qquad (4)$$

where $\begin{cases} q_2 + q_3 + q_4 = q_r \\ q_2 \geqslant 0,\ q_3 \geqslant 0,\ q_4 \geqslant 0 \end{cases}$

and are established according to the criterion given above.

In the case that redistribution is considered from one store to two or more stores, each incremental item considered for the various q_r must go to the store offering the greatest marginal saving (i.e., that having the greatest slope of $C_i(s,t)$).

3.12.7 Comments on Berman's work

Berman's work is designed only to cope with the rather specialized problem of interdistribution between stores where no further procurement from any source will arrive in the future. The length of the decision periods is unfortunately neither defined nor suggested.

It is very difficult to see the exact type of comparison which is suggested in the extensions of the ideas to multiple-store redistribution. It is not made clear whether shipment from the store with the lowest value of $C(s,t)$ is considered before shipment to the store with the highest value of $C(s,t)$.

Neither set of decisions necessarily yields the optimal solution. A very large number of combinations of redistribution combinations is necessary for this.

To illustrate this point, the optimal decision for a period t may be:

Ship 5 from store 2 to store 4

Ship 2 from store 3 to store 1.

Berman shows no way in which this type of result is obtainable except by the (obvious) method of considering every combination of inter-store redistribution available.

3.13 The Work of K. Borch

3.13.1 Introduction

In an article[23] in the Academy of Management Journal, Borch considers the problem of inventory decisions in a hierarchial inventory situation.

The situation considered in a simple model is that of a retailer and wholesaler.

3.13.2 The model

If demand exceeds stock, the retailer can order an amount up to z from a wholesaler and he makes a profit of a_2 on the resale of ~~these~~ each unit~~s~~.

a_1 is the profit he makes on stock demanded ex retail stock. The retailer's expected profit is given as

$$P(y,z) = a_1 \int_0^y x\, f(x)\, dx + a_1 y \int_y^{y+z} f(x)\, dx + a_2 \int_y^{y+z} (x-y) f(x)\, dx$$

$$+ (a_1 y + a_2 z) \int_{y+z}^{\infty} f(x)\, dx - b_1 y$$

where y is the retail stock held by the retailer, b_1 are the storage costs per unit, and f(x) is the probability density function of demand.

For the case that the retailer knows z, his optimal stock is determined by

$$\partial P/\partial y = a_1\{1-F(y)\} - a_2\big(F(y+z)-F(y)\big) - b_1 = 0$$

where $F(y) = \int_0^y f(x)\, dx$ and $F(y+z) = \int_0^{y+z} f(x)\, dx$

Further, $\partial P/\partial z = a_2\big(1-F(y+z)\big) \geqslant 0$,

and so the retailer's expected profit increases with z. Thus, the retailer gains by the wholesaler holding large inventories which can be called on when needed.

By similar analysis, it is shown that the expected profits of the wholesaler will decrease as the stock y held by the retailer increases.

3.13.3 Results of the situation and approach to the solution

The retailer cannot decide on optimal y unless he knows the wholesaler's choice for z. Similarly, the wholesaler cannot find an optimal z until he knows the retailer's choice for y.

Two methods of analysis are considered by Borch. The first of these analyzes the situation as a learning process, the second as a Two-Person Game.

The first method leads to an unsatisfactory solution, unsatisfactory since there exist a number of other combinations of y,z for which both parties receive higher profit. The second method employs the usual Game Theory assumption that parties are allowed to co-operate, and that they in some manner bargain their way to an arrangement which both accept as the best they can hope for in the given situation. This limits the range of considered (y,z) combinations, but to establish the eventual "agreed" combination, additional assumptions are required.

Borch continues by discussing ways in which to generalize this model. The introduction of price and the "general public" is cited.

3.13.4 Comments on the paper of Borch

The hierarchial decisions involved here are interesting since they do have some relevance to a central store/sub-store problem where the supply to the central store is by a wholesaler. The models suggested by Borch can then to some extent aid a simplified model of the central store/sub-store problem to be built in which some estimate of the stock available to supply the complex can be obtained.

3.14 The Paper of J. Magee

3.14.1 Introduction

In an article[18] in the Harvard Business Review Magee considers a specific central store/sub-store complex with supply from a factory under the same ownership as the complex.

It is strictly a case study and the way in which service was improved and production stabilized are the most important features.

Originally central store and sub-store stock control was very much a haphazard process. A suggested improvement was a reorder level based control for both sub-stores and central store, the reorder quantity for sub-stores equal to $\sqrt{2c_R\lambda_i/hP}$ where c_R is the fixed ordering cost, λ_i is the sub-store i demand rate and hP is the holding cost per unit time. The reorder quantity for the central store was made equal to $\sqrt{2c_P\lambda_T/hP}$ where c_P is the cost of a supply (procurement) and $\lambda_T = \sum_i \lambda_i$.

Under this control, production fluctuations were no larger than before, but the average change in production was equal to 80% of the average production level, leading to excessive production costs.

3.14.2 Stabilization of production: Control 2

A new control was established with cyclical review at sub-stores. The central store would order the issued quantity from production, and receive the order within two weeks or by the beginning of the next review period, whichever was the greater.

The review period was established by considering the annual cost of operation with different review periods and then choosing that period yielding minimum total costs. Assumptions were:-

(i) Sub-store buffer stock was such that there existed 0.25% chance of a shortage in any one week.

(ii) Central store buffer stock was set to allow a 1% risk that it "would be unable to replenish all sub-stores immediately".

(iii) Central store stock would average at half the average demand in a cycle plus its buffer stock.

(iv) Production change costs were proportional to change in production level between cycles - equivalent to change in overall demand on the complex between cycles.

3.14.3 Results of the new control

With the data relevant to the specific case considered about 60% of the former cost of operation was cut. Although total inventories were reduced by 70%, most of the savings was obtained from smoothing production. Further savings resulted since the reductions in production fluctuation made possible production deadlines to be met regularly, which cut the sub-store lead time allowing small reductions in sub-store buffer stocks.

3.14.4 A further suggestions for control: Control 3

In a new suggestion for control, the sub-stores "report" stocks at periodic time instants and the factory produces according to the demand. in the previous period. Replenishment to sub-stores only occurs from the central store when the "reported" demand "totals" an economic shipping quantity. It was hoped that (i) this control would reduce production fluctuation and buffer stock needs still further and (ii) fewer sub-store replenishments would be necessary.

3.14.4.1 Policy decisions required

3.14.4.1.1 Period between reporting demands for sub-stores

By considering the costs associated with reporting intervals of one week and two weeks respectively (these costs were taken as the sum of inventory holding cost, production change cost and clerical costs) and taking the interval giving the least cost, the period between reporting demands (it turned out to be 1 week) was decided upon.

3.14.4.1.2 Size of replenishment to sub-stores

By balancing inventory holding cost against replenishment costs and ignoring cost of backup, an economic order quantity is established.

3.14.4.2 Results of this control suggestion

Over 20% cost reduction of the previous control is the result quoted.

3.14.5 Ideas for further production stabilization: Control 4

The production level under Control 3 was being adjusted each week to account for the full change in inventory due to demand fluctuations in the complex. The proposal here is to limit the adjustment in production to some fraction of the difference between actual demand and average demand. As the production is smoothed, the buffer stock will be correspondingly greater. Study showed that the total cost of production change and buffer stock could be minimized with a response or reaction rate equal to .125, i.e. where the production adjustment is limited to .125 times the difference between actual demand and average demand in the period since the last report.

3.14.5.1 Results of the application of Control 4

Another total cost reduction by about 18% of that with the previous control is claimed.

3.14.6 Comments on the paper of Magee

As this paper is not intended to be a significant contribution to the theoretical literature, its non-generality cannot be a major criticism. In showing that considerable savings may be obtained by a sensible inventory control procedure, the paper does excellent work.

There are several minor criticisms, however. The first of these is that the economic order quantity for both sub-stores and the central store does not take into account the cost of backorders (which may or may not be significant in the case of Control 1. Clearly this decoupling of control and establishing of independent ordering rules is not very sensible anyway. In consideration of Control 2, inability of the central store to "replenish all sub-stores immediately" does appear to be somewhat ambiguous. Presumbly it means "being able to supply the whole of the order", and in this case, one presumes the central store would issue out all the stock it has.

Control 3 includes an economic order quantity which again does not account for backup. Further, it is likely that (for other than short-length periods between reporting demands) when replenishment to sub-stores occurs "reported demands" significantly exceed the economic order quantity. This in turn leads to either backup or unnecessarily high buffer stock, whence high inventory holding costs.

The general computation throughout the paper of an economic order quantity is to be criticised. It is not clear that the computation takes into account all the relevant costs. The amount delivered and when to deliver to sub-stores are interrelated, and clearly the shipment quantity should not necessarily always be the same when the distribution of stock over the complex is different.

It is not clear that the reporting of demands in Controls 3 and 4 for sub-stores means simultaneous reporting for sub-stores or lagged reporting times. Clearly there are some savings to be gained from the latter alternative.

The final point is that this case study deals with the special case of supply to the complex under the same ownership and most of the savings occurred from smoothing production. Control for this case is required to be adapted accordingly, and it is not necessarily true that this type of control is best suited to other cases where procurement comes from an independent concern or where production smoothness is not so important.

3.15 The Work of J.A. Cran

3.15.1 Introduction

This work has been treated by the present author rather differently than previous works, in that whilst presenting this author's work, discussion takes place alongside it. This is because it is felt that Cran's approach to the problem is a really useful step forward along interesting lines and the ideas for his control will be used as a basis of comparison for the ideas developed in this thesis.

3.15.2 Summary

Cran's work was triggered by the existence of a two-level distribution problem, i.e., a problem concerning the central store/sub-store complex, for the factory warehouse (central store) to the thirty sales branch warehouses (sub-stores) of Massey-Ferguson Ltd., Toronto.

Cran's model of the real complex consisted of a central store and ten sub-stores subject to equal random demand and separated from the central store by a two week (constant) lead time. ~~Like the author in this respect,~~ Cran (and also the present author) excludes the possibility of transshipments between stores. He gives the reason that the unit cost for such transshipments is so much greater than shipment from the central store.

Stock to the complex via the central store occurs, as in the author's model in one batch, the 'procurement' which, in Cran's case, is subject to a variable 'reorder' lead time of known distribution. Demands on the complex come only in the form of retail demands on the sub-stores, with backorders being accumulated. Procurements and deliveries between central store and sub-stores cost fixed amounts. Inventory holding charges are (as with the author's model) independent of whether the stock is held at central store or sub-store.

In his paper on this subject[14] Cran states that "at first glance it might seem that this system could be controlled on a two-level E.O.Q. basis" with each branch ordering from the central warehouse in E.O.Q's based on its own forecast sales and the redistribution set-up cost. ~~However, he fails to make clear that in the case of stochastic demand, the E.O.Q. formula gives too low a value for procurement quantity owing to the fact that it fails to take into account the expected cost of shortages~~. Cran then goes on to show the lack of rationale in a two-level E.O.Q. basis of control, this lying within the fact that when deciding on the quantity of shipment between central store and sub-store, the cost of holding stock should not be considered, as no additional holding costs result from that decision. One might add that even if additional holding costs did result from the decision on quantity of shipment the E.O.Q. rule would still not take into account expected costs of shortage at the other sub-stores, these, apart from other factors, being dependent on the amount of stock remaining at the central store.

However, two important points require making at this stage. Cran notes that in widespread commerical applications of the complex under consideration each store (including central store) is controlled independently on high/low limits (presumably on an (s,S) rule). Secondly, he recognizes that such independent commercial operation is irrational for a complex under single ownership, and that a more sophisticated control involving a 'rationing' to the sub-stores is required so as to offset the excessive cost of shortage which can result from independent store operation on (s,S) rules.

3.15.3 The proposed control rule

When a procurement arrives at the central store, distribution to sub-stores occurs at once according to the following procedure. The total stock in the system (including the procurement) is summed and back-orders subtracted. This gives "System Stock". Having a forecast of total system demand rate, the expected time until the total system stocks out, the "mean stock-time"* is calculated. Then, in order to find the value of the standard deviation of sub-store demand during the actual stock-time, he makes the following approximation, making the assumption that the distribution of demands is Poisson:-

$$\sigma^2_{STD_i} = (\bar{T}\sigma_{MSTD_i})^2 + (\lambda_i\bar{T})^2\,\sigma^2_{ST}$$

where σ_{STD_i} = standard deviation of sub-store demand during stock-time

$\bar{T}$ = mean stock-time

* If the total stock in the system is S and λ_T is the total system demand rate, mean stock-time is S/λ_T. Stock-time is the actual time until the system stocks out ($S=0$).

λ_i = forecast of sub-store demand rate during mean stock-time

σ_{ST} = standard deviation of stock-time

σ_{MSTD_i} = standard deviation of sub-store demand during mean stock-time

The method of stock allocation will now be described. Each sub-store is allocated its mean stock-time demand (i.e. expected demand in the mean stock-time) less a 'hold-back'. The sum of the hold-backs from all the sub-stores is kept at the central store and is used for redistributions to needy sub-stores as stock-out of the complex approaches.

The hold-back is defined as a control parameter multiplied by the standard deviation of the relevant sub-store demand during stock-time. Thus the allocation* to sub-store i is $\lambda_i \bar{T} - \bar{A}\sigma_{STD_i}$ where $\bar{A}$ is the control parameter which Cran wishes to optimize.

3.15.4 Criticism of "Mean Stock Time"

The logic behind the idea of "Mean Stock Time" is apparently to ensure that no sub-store has too much stock by the time the whole complex is short. The criticism of this is that when shortage cost is not small the whole complex is likely to be short quite rarely since a new procurement is likely to arrive well before stock-out time. This policy is then guarding against a danger which ~~in many instances~~ when shortages are costly does not occur.

There are a few instances where Cran's policy has relevance, and is likely to provide good control. This is where both the unit shortage cost is small and replenishment is cheap. Then good policies will necessarily entail sub-store shortages and the complex may have numerous deliveries without incurring high total delivery costs.

3.15.5 Criticism of holdback

There is little rational support for making the holdback proportional to the standard deviation of stock-time requirements. Presumbly, the logic is to ensure that each sub-store has (roughly) the same (small) chance of being overstocked when the whole complex is very short of stock. We have already shown however that this is guarding against an almost non-existent danger, and to do this to the same degree in every sub-store is not a major virtue.

3.15.6 Sub-store reorder point

Cran's proposed rule considers redistribution from the central store to a sub-store whenever the stock level of the latter reaches a reorder level. The latter is established as a minimum stock level for a branch such that redistribution can be effected from the central store in the three

* Note that this implies that shipment quantities equal allocation less notional stock for any sub-store.

week lead time (consisting of one week review plus two week redistribution) with little fear of a stock-out. The sub-store is then shipped the difference between the allocation and its virtual stock except when the allocation is less than three weeks' mean demand on the sub-store, when the allocation is altered to three weeks' mean demand. Redistributions 'continue on this basis until the central warehouse (i.e. central store) stock is exhausted'. (Presumably, Cran also means "or until the central store receives a procurement, and then the redistribution cycle recommences".) Allocation, in the case of redistributions, is calculated in the same way as for the initial distributions to sub-stores occurring at the time of procurement arrival. That is, for each redistribution necessary, mean stock-time is calculated, and the estimated value of the standard deviation of that sub-store's requirements during the actual stock-time computed from the given approximating equation, holdback being the factor 'A' multiplied by this standard deviation.

3.15.7 Procurements for the complex

Procurements for the complex are considered whenever the total system stock falls below a reorder level M. The amount of a procurement quantity is Q, and the three control parameters, viz. $\overline{A}$, M, Q of the model are to be optimized.

3.15.8 Some criticisms

~~To date, Several points of a critical nature are worth making~~. Firstly there is no absolute necessity to distribute stock as soon as a procurement arrives at the central store. ~~Secondly, in a fully computerised operation, one week of the lead time between the central store and sub-stores is eliminated, for the computer will have knowledge of the stocks within the sub-stores which is very much more up-to-date. (A data link with stock change information being transmitted to the computer centre is envisaged.)~~ ~~Thirdly~~ Secondly, the author feels that greater justification for raising the minimum sub-store allocation to three weeks should be offered. It is recognized that deliveries to sub-stores should carry a high value of stock if possible in order to reduce overall transport costs, but when overall stock in the complex (i.e. system Stock) is low enough that the calculated allocation is less than three weeks' mean demand, raising the allocation to an arbitrary three weeks can well increase the risk of run-out at other sub-stores. ~~Fourthly~~ Thirdly, it would be more helpful to readers if Cran were more specific about the establishment of the sub-store reorder level than quoting it as "that level such that redistribution can be effected in the three week lead time with little fear of run-out".

If it can be assumed that this implies that there will be only a small probability of the Poisson demand exceeding the reorder level in the three weeks lead time, this probability being decided upon by management and a policy variable (using such data as extra costs of holding stock versus drop in probability of run-out), then this is satisfactory, but should be made explicit. It might, however, be a better idea, to make such a probability-of-run-out a control variable, but inherent difficulties owing to the extra cost of optimizing a four-variable function are recognized.

3.15.9 Determination of optimal cost of parameters and the associated total cost of control

Cran obtained an approximately optimal combination of the Q, M, $\overline{A}$ parameters by simulation. He wished to obtain an improved method which could establish the optimal combination for a variety of combinations of costs and demand rate.

In order to do this a Total Cost function per unit time, T.C., was built:-

$$T.C. = \frac{\lambda_T}{Q}\left(c_P + \varepsilon(C_{R_0}) + \varepsilon(N_R) \times c_R\right) + \left(\frac{Q-\varepsilon(N_S)}{2} + (M-\lambda_T L)\right) hP \qquad \ldots \qquad (a)$$

where λ_T is the total demand per unit time

c_P is the procurement cost

$\varepsilon(C_{R_0})$ is the expected cost of shortage per procurement equal to $\varepsilon(N_S) \times c_S$

$\varepsilon(N_R)$ is the expected number of replenishments per procurement

$\varepsilon(N_S)$ is the expected number of shortages per procurement

L is the procurement lead time

hP is the holding cost, per unit time

c_R is the ~~unit replenishment~~ cost of a replenishment.

Cran then proceeds to obtain from the results of simulation graphs of the total number of replenishments γ_R versus System Stock S for several values of the holdback factor $\overline{A}$. The probability of reaching a particular value of System Stock and hence incurring the corresponding γ_R will be governed by both the reorder level M and the distribution of System lead time demand.

For a particular value of $\overline{A}$, the expected total number of replenishments per procurement, $\varepsilon(N_R)$, is obtained from the following simulation:

$$\varepsilon(N_R) = \text{function } (M,\overline{A}) = \sum_{S=M}^{\infty} \gamma_R(S,\overline{A}) \times f(M-S) \qquad (b)$$

where f(M-S) represents the probability that the total demand in the procurement lead time is (M-S).

By a similar method which uses the result of ten simulator runs for total number of backorders γ_S experienced versus Total System Stock S, Cran obtains by approximate integration a value for the expression:-

$$\varepsilon(N_S) = \text{function } (M,\bar{A})$$

He makes the point that strictly the γ_R, γ_S, are also functions of Q, but he considers the dependence to be sufficiently slight over the range of Q "involved in proceeding to a solution" that this effect can be ignored.

3.15.10 Differentiation of the total cost function

Setting the partial derivatives with respect to Q, M, and $\bar{A}$ respectively equal to zero, yields the following set of equations:-

$$Q = \sqrt{2\lambda_T\left[c_P + c'_S \times \varepsilon(N_S) + c_R \times \varepsilon(N_R)\right]/hP} \qquad (c)$$

$$\left(\frac{QhP}{2\lambda_T} - c'_S\right)\frac{\partial}{\partial M}\{\varepsilon(N_S)\} - c_R\frac{\partial}{\partial M}\{\varepsilon(N_R)\} = \frac{QhP}{\lambda_T} \qquad (d)$$

$$\left(\frac{QhP}{2\lambda_T} - c'_S\right)\frac{\partial}{\partial \bar{A}}\{\varepsilon(N_S)\} - c_R\frac{\partial}{\partial \bar{A}}\{\varepsilon(N_R)\} = 0 \qquad (e)$$

where c'_S is the unit cost of shortage. This analysis is only applicable where $c'_S\ \varepsilon(N_S) = \varepsilon(C_{R_O})$, i.e. shortage cost is dependent on number of run-outs only and there is no time dependent cost factor.

3.15.11 Method of obtaining the optimal combination of parameters

Equation (c), being very stable provides a value for Q when M,$\bar{A}$ are guessed. Substitution of this Q in equation (d) yields M. Iterations between these two equations are said to settle down quickly to limit values for Q and M. Substitution of these in equation (e) yields $\bar{A}$. The procedure now is to return to equation (c) and use the new M,$\bar{A}$ and commence iterations again.

This procedure can be used to obtain optimum Q, M, $\bar{A}$ for various (c'_S, P, h, c_P, c_R) combinations.

Cran is particularly interested in knowing the cost of operation for different unit costs. P as the value of service (gauged as fraction of demands which are immediately met) is varied. The method by which these results are obtained is interesting. The equations (d) and (e) are evaluated with various unit shortage costs c'_S. The performance of the complex with regard to backup thus correspondingly varies, and the relevant parameters M, $\bar{A}$ and Q are obtained. However since the actual cost of filling a backorder is essentially fixed, the total cost of operation in equation (a) is evaluated with the true value of c'_S.

3.15.12 Comments on the method of determination of optimal combination of parameters

It is clear that the method given is restrictive. Firstly, it is able to deal only with those rules of demand in the stores complex for which the graphs $\varepsilon(N_R)$, $\varepsilon(N_S)$ were established. There is no evidence to suggest that these curves will be identical for the stores complex if the individual demand rate λ_i changes and similarly these curves would require to be re-established for any other stores complex in which the individual λ_i were not identical with those employed in the one specific case considered by Cran.

Secondly, the $\varepsilon(N_R)$, $\varepsilon(N_S)$ curves are said to be dependent on Q, $\overline{A}$, M, but the dependence on Q is slight "over the range of Q involved in proceeding to a solution". The implication is that to obtain Q for any one value of hP, the curves may be considered as essentially independent of Q. However when coming to use the curves to obtain the optimal combination of parameters for different values of hP (other than that for which the $\varepsilon(N_S)$, $\varepsilon(C_{RO})$ curves were constructed) it will be seen from equation (c) that the range of Q can be quite high, with Q varying as the inverse square root of hP. Clearly, the dependence of $\varepsilon(N_R)$ and of $\varepsilon(N_S)$ on Q over the new range involved in the change of hP is not slight. Hence it may be wrong to use the ~~same~~ original curves for the new iterations.

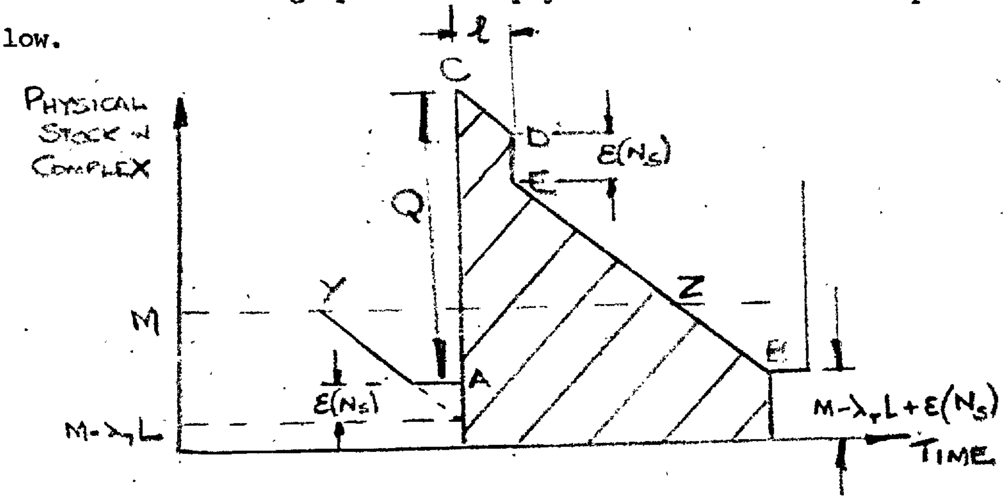
Thirdly, by utilizing the $\varepsilon(N_R)$, $\varepsilon(N_S)$ curves in proceeding to a solution for the optimal parameter combination, the same reorder level for the same sub-store must be used throughout (there can be no postponements of a delivery to try and save the cost of a delivery by waiting until the procurement arrives at the central store) otherwise undesirable randomness is likely to be introduced into these curves. Similarly a change in policy to one which ships out plenty of stock when the complex has adequate stocks, cannot easily be incorporated if reliance is to be placed on the $\varepsilon(N_R)$, $\varepsilon(N_S)$ curves.

Fourthly, the method of obtaining the optimal combination of parameters does rely at present on the fact that overall shortage cost is directly proportional to the number of experienced shortages. To cater for the case when overall shortage cost is time dependent, two curves would have to be considered, viz. (i) the expected number of shortages versus S, and (ii) the expected cost of shortages versus S. It is likely however that the latter curve may require many more simulation runs to provide a good average, since it is likely to be much more variable.

3.15.3 Criticisms of total cost of control equation (a)

Equation (a) is presented as the total cost equation for operation. It purports to account for the occurrence of backorders in the holding cost term. If the number of shortages which is expected per procurement is $\varepsilon(N_S)$, then we expect (since this backup is captive) that $\varepsilon(N_S)$ of the Q items arriving in the central store are earmarked for immediate shipment out of the complex.

The stock time graph for the physical stock in the complex is given below.



The average value of stock held in the complex is the value

$M - \lambda_T L + \varepsilon(N_S)$ + Area ABEDC/AB

$$\text{Area ABEDC} \doteq \tfrac{1}{2}AB\left(Q-\varepsilon(N_S)\right) + \varepsilon(N_S)\ell$$

$$\text{Average stock} = M-\lambda_T L + \varepsilon(N_S) + \tfrac{1}{2}Q - \tfrac{1}{2}\varepsilon(N_S) + \frac{\varepsilon(N_S)\ell}{AB}$$

Now AB is the average time complex demands $Q-\varepsilon(N_S)$ items

$$= \frac{Q-\varepsilon(N_S)}{\lambda_T}$$

whence average stock

$$= M-\lambda_T L + \frac{\varepsilon(N_S)}{2} + \tfrac{1}{2}Q + \frac{\varepsilon(N_S)\ell\lambda_T}{Q-\varepsilon(N_S)}$$

If the product $\varepsilon(N_S)\lambda\ell_T$ is small with respect to $Q-\varepsilon(N_S)$ the last term can be ignored, but the average holding cost per unit time resolves into

$$\left\{\frac{Q + \varepsilon(N_S)}{2} + (M-\lambda_T L)\right\} hP \quad ,$$

in contrast to the expression for average holding cost given by Cran in equation (a).

3.16 Summary of Chapter Three and Introduction to Chapter Four

This chapter has presented the works taking the steady state approach in controlling the central store/sub-store complex. The works by authors who do not deal with this particular complex but with similar stores complexes are also reviewed. It was felt worthwhile to include them since some of the ideas presented are relevant. For example the papers of Hadley and Whitin and of Shakun deal with a chain of stores without a central store. In both cases, the present author shows how the incorporation of a central store is likely to improve control.

The model due to Hanssmann, and the work related to it by Lampkin is applicable where demands are normally distributed in review periods, and both central store and the set of sub-stores are controlled by cyclical review systems with simultaneous reviews.

The model due to Lawrence applies where demands are from a Poisson process, sub-stores work on a base stock policy, and the central store stock is controlled by a reorder level system.

Lampkin considers several control policies applied to his model in which the demands on sub-stores are from the same Poisson process and where the lead time out to sub-stores from the central store is zero. From consideration of control with the different policies over a wide range of cost combinations, he concludes from his model that a reorder level based policy at sub-stores and a trigger taking into account the levels of stock at the central store and at the sub-stores are likely to prove to be valuable ideas.

The model due to Cran is applicable where the sub-store demands are from a Poisson process, and where the sub-stores are controlled on a reorder level policy. Cran triggers the order for the complex on the combined stock of central store and sub-stores and pioneers a rationing rule which modifies the amount to be shipped to an ordering sub-store by an amount proportional to the standard deviation of the demand of the sub-store in the "mean stock time". This latter concept is the expected time until there would be zero stock in the complex if the complex were not to be replenished from outside.

In the following chapter, the author's own model is presented. The assumptions are discussed, and the data are given. The latter are initially taken to be comparable with one of Lampkin's cost combinations, and the combination of assumptions plus data is referred to as the "model".

CHAPTER FOUR

ASSUMPTIONS OF THE INITIAL MODEL OF THE COMPLEX

4.1 Assumptions of the Model of the Central Store/Sub-stores Complex in which the Control Rules were Developed

4.1.1 Sub-stores have identical demand distributions

This is no doubt an assumption which has significant practical limitations, but, as has been mentioned above, this does not interfere with the usefulness of the model either in the comparison of operating rule costs or in indicating the method by which optimal parameters are calculable. It is admitted, however, that some concepts introduced may need slight modification when considering non-identical demand distribution on sub-stores.*

4.1.2 Demand on sub-stores is Poisson distributed and stationary

This neglects within the model the possibility of seasonal variation and of predictable growth or decay in the demand rate, and hence the control rules adopted in their present form are not applicable to these cases. It will be shown later**, however, that the rules developed are robust enough to deal with minor unpredictable changes in demand rate.

4.1.3 Fully computerised operation of the complex

This means that all data related to the substores within the complex will be held centrally and will be up-to-date. In practice this implies that a real-time operative computer will control the complex, and, of course, no autonomous sub-store decisions will be allowed.

This may not be the situation in practice for complexes of this nature at present but the author is noting that such 'automatic' control is rapidly growing, and there is a need for methods to control such systems.

4.1.4 Demand on sub-stores is for one item per demand

Once again this assumption limits the practical significance of the model. However, when the control rule is applied to a more real-life complex, where several items might be requested on every occasion of demand, its basic form will not change.

4.1.5 (i) Lead time between supplier and central store is constant
~~(ii) Lead times between central store and sub-store are constant and the same for each sub-store~~

In some complexes of the type under discussion we may find that ~~either of~~ the assumption~~s~~ above is not applicable. However, the basic control

---

* Example, "Free Stock" (see Chapter 5)
** Chapter 16.

rules developed here are independent of ~~these points~~ this point.

4.1.6 <u>No shipments of stock between sub-stores</u>

There may be times in practice when the shipment of stock from one sub-store with a high level of stock to another requiring replenishment seems a good idea, but this has not been considered, as it presents an additional complication to the main elements of the control rule. However, it is recognized that, in the event of a run-out at a sub-store concurrent with a run-out at the central store, the best decision might well be to ship between sub-stores.

4.1.7 <u>Items are treated independently</u>

This assumption results in the model dealing with one item only. The practicality of this assumption is not, as might appear at first sight, limited to complexes wherein only one item is stored.

The question to be asked is:-

"Under what circumstances can the inventory positions of other items be ignored when dealing with an item?"

For the case of slow-moving items or where not many items are held, it is unlikely that a shipment of another or other items will be a good idea at the same time as the shipment at present undertaken. The value of using the present shipment to include some other item or items will diminish as the cost of the item goes up and as the cost of delivery goes down.

In the case that the complex stocks many items there are likely to be shipments to each sub-store regularly. In this case the cost of a shipment of an order, whenever the order is made, will only be the relevant marginal cost, viz. packing, stock movement records, requisitioning costs and loading, unloading and binning costs. Whereas for slow-moving stock one might, in a rare instance, save the whole cost of a lorry's movement, in this latter case of many items, nothing is to be gained.

We thus have two practical cases where the present model is applicable to a multi-item central store/sub-store complex.

It is in the case where there are several items (neither few nor many) that treating items independently is open to criticism. In this situation, the decision at every sub-store order time will have to be made as to whether or not the stock of any other item held at this sub-store should be replenished or not. The factors to be considered for this decision are the savings available from the possible saving of deliveries and the penalty of cost of maldistributing stock by replenishing this store with other items earlier than otherwise. Although it is an interesting problem, interdependence of items will not be considered further in this thesis.

4.1.8 Run-outs at sub-stores can be dealt with only by sub-stores

It may occur that a run-out has occurred at a sub-store and that the overall cost of run-out can in practice be cut by dispatching the goods direct from central store to the demanding customer than via the sub-store. In the general case, however, when there exists stock in the central store over and above the sum of sub-store back-up the shipment to the sub-store is greater than its value of back-up, so that it will then have some stock on hand. This, then, will reduce the possibility of incurring a saving by dispatching direct from central store to customer as indicated above. The limiting factor will be the relative costs of delivery, run-out etc. The possibility of such saving has not been considered.

4.1.9 Cost of stock-holding proportional to cash value of average stock held

Strictly the costs of holding stock are the costs of labour to man stores, the overhead and maintenance of the stores, and the opportunity costs of keeping capital tied-up in stock. With the reasonable assumption that over the range of stock that will be held within the complex, labour, overhead, and maintenance costs will remain independent of stock on hand, then the cash tied up in stocks remains the only significant factor. At any one instant the cash value of stock held in the complex will represent capital tied up in stocks, and in the long run this cost will be the cost of holding unit stock times the time-averaged value of the stock$\sim$time function.

4.1.10 Demands on sub-stores are captive

In the case that the demands on the sub-store of the complex occur from plant under the same ownership as that of the complex, run-out costs are clearly measurable in terms of lost production, and it is likely that their being proportional to time outstanding is a good estimate, at least within a simplified model.

When, however, the demands on the complex sub-stores are from outside of the complex ownership, two possibilities occur. Either the demand is dropped, the cost of run-out being then equal to the tangible loss in sales profit plus cost of loss in customer goodwill or the demand is held 'captive' and the customer seeks supply when the latter is possible. The longer the demand is outstanding on the sub-store in this case, the greater the loss in customer goodwill. The actual value of this loss of goodwill is considered quite significant by sophisticated management for it is evident that run-out will breed lack of confidence and lower the future volume of the complex's business. In the case of 'captive' demands, it may be expedient to make the delivery to the customer by means other than would normally be the case, in order to reduce incremental run-out costs by the demand being outstanding for longer than is physically necessary.

The latter point will be ignored as it represents a complication to the main formulation of the control rules, but its significance can be invoked at a later stage. For the purposes of a simplified model, only the situation where demands are captive will be considered, as this is the situation considered by Cran[14], with whose work the control rules suggested by the author are to be compared. The control rules need no modification to deal with the situation of non-captive sub-store demands.

4.1.11 Cost of supply to sub-stores is constant; same for each sub-store

Even in the case that the transporation medium is such that the delivery costs are of the form:- Constant plus Constant times quantity delivered, the quantity-dependent part of the cost may be ignored for comparative purposes, since the total quantity delivered to any store is independent of the control rules when demands are captive and hence this cost would be incurred for any control decision made. It is unlikely that the cost function will involve powers of the quantity delivered; hence for the great majority of practical cases, the shipment cost is unaffected by shipment quantity.

There is, further, an implicit assumption that the costs of supply to sub-stores are independent of the day or time-of-day at which supply takes place. Clearly, this is an approximation (though, it is suggested, a good one) for one can envisage a sub-store delivery on a Friday afternoon in congested traffic costing more than a delivery at a more peaceful time in the week.

Rather more important an assumption is that costs of supply to different sub-stores are the same. In the case of postal deliveries, this assumption is reasonable. In the case that the fixed costs of delivery vary between sub-stores, it would be prudent to reflect this in the control rules, but this point is ignored in the model.

This means that the results from consideration of this model are strictly not applicable to models where there are different costs associated with deliveries to sub-stores. In answering this criticism, the author simply feels that to commence with, restrictive assumptions have to be made, and this particular one is thought not to detract significantly from the value of the work.

4.1.12 Cost of complex procurement (i.e. central store supply) constant

The quantity of the complex procurement will be invariate*, and provided the supplier to the complex does not change its geographical location, cost is only liable to vary insignificantly (on time of day or day of week of replenishment) though the more likely situation is that a fixed cost will be quoted for every procurement, if this comes from an outside supplier.

When, on the other hand, the procurement comes from the production of a factory under the same ownership as that of the complex, the problem of costing the procurement is more difficult. In addition to the costs of transportation and paperwork etc. the costs of change-over of tooling for the batch-production run (assuming that the factory is engaged in batch production) should be taken into account.

The timing of a central store order on its supply channel (viz. procurement order) to take into account the savings possible by making the order when the change-over cost is minimal represents a significant task to develop, and appears to be beyond the scope of the present work. Suffice it to say that for the purpose of the simplified model and for models of systems (i.e. complexes) to be considered under the proposed control rules, the most realistic cost of procurement will be the mean of the costs over a large period of change-over costs, transport and paperwork costs, taking the case of supply from a factory under common ownership. If supply is from an outside supplier, the mean charge should be used.

4.1.13 Costs of run-out proportional to time backorder is outstanding

In many cases of the stores complex serving a production facility it is reasonable to suppose that costs are being incurred throughout the full length of time the demand is outstanding. In other cases (for example, where a demand occurs for an item of equipment for a machine

---

* developed in the control rules.

just finishing a batch) the costs of a backorder are not a function of the whole length of time the demand is outstanding but more strictly a function of the time the facility placing the demand is rendered redundant by this failure to supply.

In most cases of supply to a production facility, it is felt that it is reasonable to assume that if a backorder occurs, the cost is a function of the total time the demand is outstanding. The exact nature of this function is likely to differ for different production facilities, but the one that is felt most likely to represent the majority of cases is the function proportional to time outstanding. It is recognized that a function equalling "constant + time-proportional factor" may also suit many cases and even a quadratic time function is possible in other instances, but since a unique form has to be assigned to the model, the choice of the first-mentioned is made.

Additionally recognized is the fact that when the complex does not serve a production facility the cost of the backorder may well not be dependent on the length of time the demand is outstanding. Since there are naturally many business instances falling into this category, a constant penalty for each shortage cannot be dismissed. Indeed, in Chapters 16 and 17, we investigate whether the control rules developed work well under the assumption of a constant for each shortage.

4.1.14 Penalty function for shortage identical for each sub-store

This assumption is made, as with many others, not to particularise the model, but to ensure it does not become overburdened with complications which are not likely to make a great deal of difference to the overall manner of operation. This is not to say that certain modifications to the control rules will not be necessary to incorporate different shortage penalty functions at the various sub-stores, but it is felt that for the purposes of this thesis, the modifications which would be entailed are not major.

4.2 Data for Model One

For the first model, these were taken as those employed in a data configuration considered by Lampkin[3], in order that some kind of comparison might be made between the results of this work and those with Lampkin's best method of control.

Hence we have:

Item Value = 1 Unit

Cost of Procurement = .5

Cost of Supplying a Sub-store = .3

Cost of Sub-store Run-out = 40% Item Value per day

Cost of Stock Holding = 10% Value of Average Stock per year

Mean Demand on Sub-store = 10/Year, Poisson Distribution

Number of Sub-stores = 5

Working Days/year = 250

Lead Time for System = .4 year

Lead Time for Sub-store = zero.

4.3 Summary of Chapter and Introduction to Chapter ~~Three~~ Five

Chapter Four discusses the assumptions that were felt to be necessary to establish a workable model of the stores complex. At a later stage*, some of these assumptions will be relaxed.

In **Chapter Five** policies for controlling the complex are considered, and a first method of control is established.

---

* Assumption 4.1.1 in Chapters 16, 17.
Assumption 4.1.2 in Chapter 16.
Assumption 4.1.13 in Chapters 16, 17.

CHAPTER FIVE

FIRST ALTERNATIVE METHOD OF CONTROL

5.1 Initial Considerations

The restrictive nature of assumptions necessary to build a simple enough model to develop and test control rules has been discussed in Chapter 4. Despite these restrictions, it is argued that rules developed and tested on this model will be easily generalised to control real complexes of stores. The method of establishing the optimal values of the control parameter for models of actual commercial complexes is similar to that for obtaining the optimal values of the control parameters in the author's simplified model. Some modifications to the control rules to make them applicable to more general complexes are discussed in Chapters 16 and 17.

There are five main decisions to be made in controlling the complex. The solutions to the following problems form the mainstay of the control rules proposed. The point is once again made that there are no autonomous sub-store decisions, all decisions being made centrally with overall knowledge of the complex stock positions.

5.2 Five Main Problems of Control

1. How much stock should the complex reorder? i.e. what is the "Procurement Quantity", Q?
2. At what instant should the complex reorder?
3. At what instant should a sub-store reorder?
4. How much stock should be allocated to an ordering sub-store?
5. Should distribution be made to sub-stores necessarily immediately stock arrives at the central store?

To begin with, it was recognized that, inevitably, there would be a necessity to search over a wide range of combinations of the control parameters to obtain the approximately-optimal combination resulting in minimal total-cost of operation. This, in turn, led to the recognition that much effort and much computing time could be saved by diminishing the field over which search for the optimal parameter combination would be necessary.

5.3 Reorder Quantity for the Complex

An expression for the rationally correct value of the Reorder Quantity of the complex Q can be obtained by considering the cost of running the complex per unit time.

If the procurement quantity is Q, then there will be a holding charge for the inventory in which capital is tied up. This will occur if the procurement arrives from an independent supplier or from a factory batch production line under the same ownership as that of the complex, since the decision to order a procurement means further capital to be tied up in stocks. The annual value of this

$$\text{Inventory Holding Charge} = \frac{hP(Q+2\overline{B})}{2}*$$

where h = rate per annum cost of capital to operating company
P = cost per unit inventory
Q = procurement quantity
$\overline{B}$ = average buffer stock held in complex.

Now with every cycle (the latter being defined as the time between arrivals at the central store of successive procurements) will be associated certain costs, viz. the cost of procurement, the replenishment costs of the sub-stores, and an expected cost of run-out for the sub-stores. The number of procurements per annum will average at $(\overline{\lambda}_T/Q)$ where:-

$\overline{\lambda}_T$ = mean total demand on sub-stores per annum

Then, if c_P = cost of procurement

and if c_R = cost of a replenishment (i.e. delivery) to a sub-store

then if $\varepsilon(N_R)$ = expected number of sub-store deliveries per procurement

and $\varepsilon(C_{RO})$ = expected cost of run-out at sub-stores per procurement

we have

Annual Total Cost of Operation of Complex

$$= \frac{hP(Q+2\overline{B})}{2} + \frac{\overline{\lambda}_T}{Q} \{c_P + c_R \times \varepsilon(N_R) + \varepsilon(C_{RO})\}$$

Now $\overline{B}$ is independent of the value of Q, but $\varepsilon(N_R)$ will depend on the rules governing sub-store distribution. These have not, as yet, been discussed in this thesis, but suffice it here to say that they are dependent on the total stock within the system. Thus it is seen that $\varepsilon(N_R)$ is dependent on Q, but the relationship is complex. It is probably more difficult to discover the relation between $\varepsilon(C_{RO})$ and Q, other things being equal, but it is likely that for high Q (i.e. long time between procurements), $\varepsilon(C_{RO})$ will be higher than for shorter periods of time, i.e., for lower Q. The variability of $\varepsilon(C_{RO})$ over the range for which Q can seriously be considered is of small magnitude, and so for convenience

* Here the effect on the holding charge of the occurrence of shortages is neglected.

and introducing small error, $\varepsilon(C_{RO})$ will be taken as a constant. The expression for the total cost of operation will be considered again in 5.9, after showing how an approximating value for $\varepsilon(N_R)$ is estimated.

5.4 Reorder Level for the Complex

The reorder level for the complex, M, should obviously be related to the form of the criterion which triggers reorder for the complex. However, the adopted trigger criterion is not a simple function of the stock levels at the various sub-stores, and hence the mathematical complexity of achieving an analytically optimal value for M is enormous, and M will be considered as a control parameter to be optimized.+

5.5 The Trigger Criterion for the Reorder for the Complex (Procurement Order)

Lampkin[3] concludes from a number of simulations involving several triggers that the trigger "... should contain information, if possible, about stock at the central store and sub-stores". It is not difficult to see this point. It is not prudent to trigger a procurement if there is liberal stock in the central store yet little in the sub-stores. On the other hand, it is unwise (because of the extra inventory holding costs) to make a procurement order if the central store has little stock while the sub-stores have plenty. Lampkin's triggers* fall into the required category. The (s,S) system worked by the total stock in the complex is similar to the trigger used by Cran, except that the amount of the procurement in the former case may exceed (S-s), whereas the latter keeps a fixed value.

5.5.1 Why total stock is not a good criterion for triggering a procurement order

There is one great failing in triggering a procurement on the total stock in the complex. This is that stock distributions of very different strategic value are treated as the same.

In the first model, the expected number of demands on each sub-store between the procurement order placement and receipt of stock from the procurement in a needy sub-store (viz. L + ℓ, i.e. 100 days) is 4 each. The reorder level of the complex would be 35, for a buffer of 15. The following stock configuration in the complex can be envisaged:-

Configuration 1	19,2,3,8,13,	Sub-store Stocks (Notional)
	0,	Central Store Stock

---

* Policy-types E,F,G,H in Review of Ref.3, see 3.7.5.

\+ In Chapter 17, this problem is solved on an analytical basis.

System Stock, defined as total physical stock in the complex less total backorders, is 45. Reorder cannot occur before 10 units are demanded in the complex. In this time the expected sub-store demand is 2 each, resulting in a situation where the expected stock levels of two of the stores are zero and one unit, respectively. Yet the central store has zero stock and no stock can get to these needy sub-stores for 100 days, in which time an expected four more demands will be made on each. Thus, when the configuration of stock is as in this example, the indicated buffer is painfully inadequate. It is to be noted that the possibility of achieving such a stock configuration in a real stores complex is not minimal; forecast demands help towards this situation.

On the other hand, we do not wish a system reorder level much in excess of 35 because this undoubtedly represents excessive costs of inventory holding when stock configurations (in which stocks are fairly evenly distributed) similar to the following example occur.

Configuration 2	)	4,7,2,5,5,	Sub-store Stocks (Notional)
	)	15,	Central Store Stock

Here, System Stock is 38. Provided that a good method of allocating stock to sub-stores is in operation, it is likely that the complex can afford to reach a System Stock level of 35 or less before making an order for a procurement, without much likelihood of run-out at a sub-store.

In one case, then, we have a complex of total stock 45 where reorder is overdue and one where total stock is 38, where reorder is not yet required. Thus information of the total stock in the system alone is insufficient as a good trigger for system procurement. It is recognized that this problem is not alleviated by raising the system reorder level to take into account stock configurations of the first type indicated above, for excessive inventory charges will accumulate, as has been shown, when configurations like Configuration Two occur.

5.5.2 Suggestions for the trigger for the complex supply

Thus a trigger for the complex is required that will better indicate the need of the complex to be replenished. An important rôle of this trigger is that it should be closely related to the likelihood of the system acquiring penalty owing to a run-out at one or more sub-stores. It is apparent that with the first stock configuration considered, the reorder point of the system should have been reached, yet it has not, the prime reason being that the whole system falsely appears to be 'safe' because of the high stock levels of 19 and 13.

Two suggestions come to mind to conquer this problem.

5.5.2.1 Suggestion One

The first suggestion is to ignore the "excess stock" of any sub-store, "excess stock" being taken in the following formula:-

"Excess stock of a sub-store"

= Max(0,(Notional sub-store stock - Average stock of sub-stores))

where Average stock of sub-stores = System stock/No. of sub-stores

Thus for Configuration 1,

System stock - Sum of Excess stocks = 45-10-0-0-0-4
= 31

and for Configuration 2,

System stock - Sum of Excess stocks = 38.

Although this appears to be a fair indication of the stock level in the complex to be used for a trigger for the second configuration, the first value of 31 appears still to be too high, for there exists in this configuration the two stores with stocks two and three respectively which will be out of stock in an expected time of fifty days and seventy-five days respectively, yet even if a procurement is ordered at this value of the trigger (i.e. 31) there is an expected high cost of run-out that will be involved.

Now it is recognized that for most complexes in commercial undertakings, the ratio of the run-out cost to inventory holding cost is very high, and the first model of the complex thus established this ratio as .4 times Item Cost per day to 10% times Item Cost per year, i.e. ratio = 1000 (taking working year as 250 days). Hence the above first suggestion of a trigger criterion is unsatisfactory because it still overestimates the 'safety' of the system relative to run-out. This suggestion can be seen to be further modified by taking the value of (System stock - Sum of Excess stocks), viz. 31, for the first-considered configuration, dividing by the number of sub-stores, and obtaining the integer value of the result, thus 31/5 = 6.2 giving 6. Neglecting the stocks of stores above this value, we have 6+2+3+6+6 = 23, the new "modified system stock value". However, these modifications to obtain a more realistic value of stock within the system to compare with a trigger quantity, although heuristically acceptable, are believed to be inferior to the next suggestion to be considered.

5.5.2.2 Suggestion Two

A second suggestion, which gets over the criticisms lodged against the first, recognizes that the sub-store with a stock of 2 in the second stock configuration is not much more likely to incur any backorder cost (owing to running out of stock).than is the sub-store with stock 5, or the one with stock 7, because stock exists in the central store to supply this sub-store. The suggestion to obtain a 'realistic stock level' of the system is to obtain a new stock configuration by a hypothetical "equalising stock distribution" from the central store. Thus in the case of configuration 2, there are 15 items in the central store. A stock of 2 is required to bring the stock level of the sub-store of lowest stock value up to that of the sub-store of next-lowest stock value. Thus for configuration 2 originally

Central store	Decrease by 2	Increase to 4
15		4, 7, 2, 5, 5

giving: 13 in central store and 4, 7, 4, 5, 5 in the sub-stores.

The next step is to consider the stock levels of the sub-stores with stocks of 4 raised in equal increments, if possible to the stock level of the sub-stores of next-lowest stock value, i.e.5, and so-on, until the stock in the central store is exhausted.

"If possible" means "if there is sufficient stock left in the central store for this to be possible".

Thus, for configuration 2, we have

Central Store Stock	Sub-store Notional Stocks
13	4, 7, 4, 5, 5 →
12,	5, 7, 4, 5, 5 →
11,	5, 7, 5, 5, 5 →
10,	6, 7, 5, 5, 5 →
9,	6, 7, 6, 5, 5 →
8,	6, 7, 6, 6, 5 →
7,	6, 7, 6, 6, 6 → ...

It is clear that finally with 0 at the central store, the following configuration obtains:-

0,	8, 8, 8, 7, 7.

One notices, then, that if a stock level of some store were actually, say, 13, then a configuration like ...

0, 8, 13, 8, 7, 7 might result.

By this hypothetical share-out process, that stock which may be neglected in the figure to represent the total stock state of the system stands out. In the last example considered where the stock levels are:-

8, 13, 8, 7, 7 with 0 at central store

we can compute a representative level of stock by ignoring stocks above the minium stock, whence we would have:-

7, 7, 7, 7, 7 with 0 at central store.

A representative stock = 7 + 7 + 7 + 7 + 7 = 35,

or we can ignore stocks in excess of one unit above the minimum stock thus:-

8, 8, 8, 7, 7 with 0 at central store.

An alternative measure of stock is thus 8 + 8 + 8 + 7 + 7 = 38.

The latter method of computation was thought to be more reasonable and the stock level thus obtaining was termed "Free Stock", denoted in the Thesis by F, and sometimes termed "F-value".

5.6 Answer to the Question: "What Quantity should be Allocated to a Sub-store at its Reorder Point?"

5.6.1 The Cran allocation

Cran[14] proposes to allow the ordering sub-store an allocation equal to the mean demand until the system stock reaches zero minus a hold-back amount. The latter equals a factor times the standard deviation of the stock-time demand for that sub-store. The lack of rationale implicit in the consideration of both standard deviation of stock-time demand, and stock-time demand itself, fostered some new thinking.

5.6.2 Allocation when a procurement order is not outstanding

5.6.2.1 A criterion of allocation

A suggested criterion for allocation when the complex does not have an order for a procurement outstanding is to ensure that an ordering sub-store receives an allocation* such that there is a fixed (high) probability that the allocation will last the sub-store until stock from the next procurement is available there.

---

* i.e. the sub-store is shipped an amount such that the resulting sum of transit stock and physical stock equals the allocation quantity.

Making this probability high will induce a relatively high number of items in a delivery. Thus the probability that the sub-store needs to order again before new stock arrives at the central store is lowered, and hence overall delivery costs are reduced.

Too high a value for this probability will result in an unnecessarily large stock level at the sub-store, and System Stock will exceed Free Stock. In this situation (which will be called "maldistribution of stocks") either stocks or backorders or possibly both will be excessive. An unnecessarily low value for the probability will give excessive replenishment costs.

The same probability rule will be used to determine allocation quantities to sub-stores both when they reach their reorder level, and when they are being replenished immediately after the central-store receives its procurement (if this is to be done). Nothing has yet been said to indicate the relative merits of a "necessarily replenish" ruling and of a "not necessarily replenish" ruling for the situation immediately following central store replenishment; these will be considered later. The probability in question will be required to be optimized as a control parameter.

At this stage, a difficulty arises. Firstly, the distribution of the demand on a sub-store between the time it reorders and the time stock from the next procurement is available at that sub-store (this time is referred to as the coverage time*) is not exactly known.

5.6.2.2 The problem of the distribution of sub-store demand in the coverage time

As an approximation, the distribution of sub-store demand in the coverage time is taken to be a normal distribution in which the mean is equal to the mean demand on the sub-store until complex reorder point, plus mean demand in the sum of procurement and sub-store lead times. (Note that this demand is the same for both "necessarily replenish" and "not necessarily replenish" policies.)

Now since reorder point of the complex is triggered by Free Stock, a first approximation to the mean demand on sub-stores to the time of reorder for the complex is given by (F-M)/N

where F denotes Free Stock

M the Reorder Level Free Stock Trigger for the Complex

and N is the number of sub-stores.

---

* It is equal to the time to arrival at the central store of the next procurement plus sub-store lead time.

(This assumes equal mean demand on sub-stores, as with the model under consideration.)

Hence the best approximation to the mean demand at a sub-store in its coverage time is:-

$$(F-M)/N + \lambda_i(L + \ell)$$

where λ_i is the mean sub-store demand rate (per unit time)
L is the lead time from supply to central store
and ℓ is the sub-store lead time.

5.6.2.3 The variance of the distribution of sub-store demand in coverage time and an expression for the allocation quantity

The variance of this distribution is difficult either to describe or to obtain mathematically. A consideration and discussion of an attempt to obtain it by simulation is given in Appendix Two.

With the assumption of a normal distribution for sub-store demand in the coverage time, a fixed probability of the allocation lasting the sub-store until stock from the next procurement is available there is equivalent to allocating the sub-store its coverage time mean demand plus a factor times the standard deviation of its coverage time demand. If this factor is the control parameter, z, then the allocation quantity A(F,M,z) is given by:-

ALLOCATION QUANTITY

$$A(F,M,z) = \frac{(F-M)}{N} + \lambda_i(L + \ell) + z\sigma$$

where σ is the square root of the variance of the demand in the coverage time. An approximation for σ is the square root of the variance of the demand in the coverage time if the complex were triggered on System Stock.*

5.6.2.4 Rules of operation in difficult circumstances

Rule 1: When the intended shipment quantity exceeds the central store stock level, the former is made equal to the latter.

Rule 2: In the case of "necessarily replenish" control, the stock of a sub-store may already exceed its allocation. In this case, no shipment is made.

Rule 3: When the sum of intended shipment quantities at any one time exceeds the central store stock, backorders are dealt with first, and then the method of allocating the rest of the central store stock is to see that the sub-store stocks are equalised as much as possible.

---

* The method of obtaining this variance is given in Appendix One.

5.6.3 Shipment quantities to sub-stores when a procurement order for the complex is outstanding

In this case, the subject of rationing is very important, and any delivery to an ordering sub-store must not put other sub-stores in jeopardy of running-out of stock.

5.6.4 Suggestions for ration quantity

5.6.4.1 Suggestion One

First thoughts led to a delivery quantity equal to:

$$\text{Central store stock} \times \frac{\text{Expected Extra Stock for ordering sub-store}}{\text{Sum of Expected Extra Stocks for all sub-stores}}$$

where "Expected Extra Stock" is taken as

Max (0, (Expected Demand in Coverage Time - Notional Stock Level))

Let us first consider a stock configuration for which we really would like this rationing rule to show up well, i.e., a situation close to the time of arrival of procurement stock into the central store.

Example 1 : Central Store Stock 7
Sub-store Stocks -1, 0, 0, 0, 1
Coverage Time $T + \ell =$ 26 days
Sub-store No.1 is at reorder point

Delivery Quantity is computed as $\left(\frac{2.04}{2.04+1.04+1.04+1.04+0.04}\right) \times 7$

= 3 (to nearest integer)

resulting in:

Central Store Stock 4
Sub-store Stocks 2, 0, 0, 0, 1

This does appear to be a sensible sort of delivery quantity for this stock configuration example. To ensure that this ration rule deals with other stock configurations in the correct fashion, we would need to build into the delivery quantity formula an element involving the relative weights of shortage cost and delivery cost. Without analysis (which is not presented here) or empirical weighting this appears impossible.

An example of a situation in which this particular suggestion for a ration rule tends to "overdeliver" is given below.

Example 2: Central Store Stock 7
Sub-store Stocks 5, 7, 12, 8, 2
Coverage Time $T + \ell$ = 70 days (in this time mean sub-store demand is 2.8)
Sub-store No.5 is at reorder point.

The "Expected Extra Stock" for all sub-stores is:

0, 0, 0, 0, 0.8, respectively.

The Delivery Quantity for sub-store No.5 is (0.8/0.8) x 7 = 7 resulting in:-

Central Store Stock 0
Sub-store Stocks 5, 7, 12, 8, 9.

5.6.4.2 Suggestion Two

At this stage, the usefulness of the Free Stock concept came to mind. If Free Stock indicates the 'useful' level of stock in the system and ignores amounts of stock in individual sub-stores over a "certain level", then surely a ration quantity equal to this latter "certain level" would not be overdistributing? The sub-store in question would then be no more likely to run out than any of the other stores. A case of overdistribution or 'maldistribution' can be argued in as much as there exists the possibility that this store will, as a result of this distribution, still have stock in hand, when the central store is depleted and another sub-store has run out of stock. There is a small possibility of this occurring - we do not expect such a rationing rule to guard against every case, but on average it is expected to work well. If the other extreme of 'underdistribution' is adopted and the ration rule specifies a quantity less than that 'certain level' obtained from the Free Stock calculation mentioned above, this type of control runs the risk of incurring an extra replenishment to this sub-store. As has been said before, the relative costs of run-out and delivery figure here, but for a general purpose rule the rationing formula specifying the individual stock level in the Free Stock calculation appears likely to be satisfactory and worth testing by simulation.

It is interesting to see how this formula works when controlling the stock configuration of Example 1 (in 5.6.4.1). For this example,

Free Stock = 7 = (1 + 1 + 1 + 2 + 2)

The Ration Quantity* is then either 1 or 2. Arbitrarily, the decision

---

* Ration Quantity is defined as the sum of delivery quantity and present notional stock at the sub-store.

was made that when the ration quantity value, computed as (Free Stock/ Number of Sub-stores) is non-integer (as in the case above) it will be rounded down. (It was considered better in general to underdistribute rather than overdistribute.)

The resulting stocks for this configuration are:

Central Store	5
Sub-store Stocks	1, 0, 0, 0, 1, respectively.

This, it is suggested, is a sensible sort of resulting distribution. Considerations of other differing stock configurations led to the conclusion that this rationing rule for sub-store delivery quantities when a procurement is on order would be employed. This latter ration rule will be known as the "Share" Rule.

5.7 Reorder Point for Sub-Stores

Lampkin[3] concludes that cyclical review based systems of control for sub-stores are not advantageous, and usually have worse performance than control with a reorder level. In the work here the consideration of cyclical review systems is excluded, and attention is restricted to reorder level control for sub-stores.* In the case of the first model (in which lead time to sub-stores is zero) there is no need to replenish a sub-store before it occurs a backorder since this backorder can be filled immediately (assuming the central store has stock) without incurring shortage cost. In the case of non-zero lead time, the reorder level can be made a control parameter. (At the first stage of the work consideration was limited to complexes where the sub-store lead times and demand distributions are identical, so this control parameter could be made identical for all sub-stores.)

5.8 Should sub-stores necessarily be replenished as soon as the procurement arrives at the central store?

In general, it was thought that a firm answer to this question cannot be given. One notes that if, at the time of procurement arrival, sub-store stock levels are low enough, the reorder level control itself will require stock to be sent out to the substore.

The advantage of replenishing sub-stores immediately upon arrival of the procurement at the central store is that this reduces the overall

---

* i.e. a shipment of stock is considered whenever the sub-store notional stock has dropped to its reorder level.

number of times that sub-store stocks are allowed to run down to their reorder levels and hence reduces the cost of backorders in this manner.* However, by making this decision to replenish sub-stores earlier than otherwise, there is the likelihood of one or more sub-stores having more stock than they need later on when other sub-stores are in need of stock. That is, we have a case of maldistribution throughout the complex which in turn leads to higher stock holding costs (since Free Stock will be low when the total stock in the complex is high) or, alternatively, run-out costs at the sub-stores in need.+

It was proposed to test the merits of the conflicting advantages of the two types of control by simulation of the complex separately using both policies termed "necessarily replenish" and "not necessarily replenish" for sub-stores at the time of stock arrival at the central store.

It was recognized that the merits of "necessarily replenish" ruling in reducing these backorders of the type specified below* would be absent in the case of zero lead time to sub-stores.

5.9 Reorder Quantity of the Complex

From the considerations of 5.3, we wish to obtain the reorder quantity of the complex.

Differentiation of the expression for the annual total cost of operation obtained in 5.3 and placing the derivative equal to zero, yields:

$$hP/2 = \frac{\lambda_T\{c_P + c_R \times \varepsilon(N_R) + \varepsilon(C_{RO})\}}{Q^2_{OPTIMAL}}$$

from which $$Q_{OPTIMAL} = \sqrt{\frac{2\lambda_T(c_P + c_R \times \varepsilon(N_R) + \varepsilon(C_{RO}))}{hP}}$$

where $Q_{OPTIMAL}$ is the optimal choice for the procurement quantity.

Initially several points were noted. Firstly, for different costs of replenishment and run-out, $\varepsilon(N_R)$ and $\varepsilon(C_{RO})$ would tend to vary, and thus the method of determination of $Q_{OPTIMAL}$ was unlikely to lend itself to very general treatment. Secondly, the Annual Total Cost of Operation of the Complex versus Q curve is quite flat in the optimal region, and the slope magnitude is less for $Q > Q_{OPTIMAL}$. The implication is that

---

* i.e., these backorders occurring as a result of the sub-store demand exceeding the numerical value of its reorder level in the sub-store lead time.

\+ The central store stock may be depleted.

a bias toward overestimating $Q_{OPTIMAL}$ is acceptable, with a fair degree of certainty that the resulting Q is close to the optimal. Further, the parameter Q is one of most insignificant related to the overall problem of controlling the complex, and it was felt that successive iterations by simulation in order to perfect Q were not worth the computing time involved.

We would like to choose $\varepsilon(N_R)$ such that the complex suffers this chosen value of $\varepsilon(N_R)$ sub-store deliveries per procurement for optimal choice of the control parameters. A few trial runs on the author's first model using a Q value computed,by taking $\varepsilon(N_R)$ equal to 10 and $\varepsilon(C_{RO})$ equal to 0, showed that the number of sub-store deliveries rose to well above 10 for combinations of the control parameters which were far off optimal, but dropped back appreciably for more sensible parameter combinations.

It was noted that the cost of run-out per cycle, C_{RO}, tended to be minimal for sensible choice of the parameters; hence $\varepsilon(N_R)$ was taken as 10 and $\varepsilon(C_{RO})$ taken as zero, and the resulting Q used for the procurement order quantity.

As a last note, it is pointed out that simulations of the complex using a reorder quantity of the complex that is non-optimal for optimal choice of the control parameters will not purposely affect the comparative performance of the complex operated under different types of control rules. It is expected that the Q values employed* will be no nearer to the optimal for Cran's control than they are to the optimal for the author's control rules.

5.10 Summary of Chapter and Introduction to Chapter Six

At the outset of this chapter the five main problem areas of control policy are specified. Each one is dealt with in turn.

In 5.3 and 5.9 the annual total cost of operation of the complex is considered, and it is shown how the procurement quantity that the complex orders from its supply is computed from this cost function. In 5.4 the level at which the complex will make its reorder is considered, and 5.5 follows on by considering criteria for triggering the order for the complex. Methods by which a "representative" stock for the whole complex may be obtained are discussed, and the final choice of "Free Stock" as trigger is made.

---

* These will, in the first instance, be the same for Cran's control rules as for the author's control rules.

The problem of what quantity should be allocated to an ordering sub-store is considered in 5.6. It is shown that this problem is appropriately divided into two policy considerations, firstly policy when a procurement is not on order and, secondly, policy whilst a procurement is on order. In the former case the quantity to be allocated, known appropriately enough as the "allocation quantity" is established as the mean demand on the sub-store in the coverage time plus a parameter, z, times the standard deviation of the demand in this time. A formula is given at 5.6.2.3. In the latter case, a policy of rationing is shown to be all-important, and suggestions for a ration quantity formula are considered. The adopted ration quantity known as "Share" is taken as the integer value of (Free Stock ÷ Number of Sub-stores).

In 5.7 the reorder point for sub-stores is discussed, and Lampkin's conclusion that reorder-level based systems of control at sub-stores are to be preferred is cited. It is shown that for zero lead time at sub-stores, the stock level for reorder (both virtual and notional) can be minus one. The final consideration of whether all sub-stores should necessarily be replenished immediately following the arrival of the procurement at the central store is then considered. It is suggested that the merits of each be tested by simulation.

A series of policies to cover all the problem areas has now been suggested and in the next chapter the merits of the proposals will be tested by studying the results of a simulation of the operation of the complex under the suggested control policies and comparing them with control as proposed by Cran.

CHAPTER SIX

THE COMPARATIVE SIMULATIONS

6.1 Summary

The control rules established in Chapter Five were compared with those of J.A. Cran[14] in Experiment One by simulating the complex under each type of control for a period sufficient enough for any non-generality engendered by the starting stock configurations to be ironed out and for a representative average annual cost of operation to be established. These simulations were executed by electronic computation either employing the simulation language C.S.L.("Control and Simulation Language") for use on the Imperial College IBM 1401-7090 installation or the Fortran language using a special set of subroutines ("Simon" simulation) for use on the Management Engineering Section's IBM 1130 installation.

6.2 Techniques of Comparing Methods of Control in the Simulations

The criterion of comparison was, in the main, the Total Cost of Annual Operation, this being the sum of the costs incurred per annum in procurements, deliveries to sub-stores, run-out costs and costs of holding stock in the complex. This total cost was taken as the average annual cost from a ten-year simulation (in some cases a four-year simulation).

In order to make the starting conditions as equivalent as possible between simulations of different M (reorder level of complex) for all methods of control of the complex, it was decided to make the times of triggering a procurement identical. This was ensured by starting the complex off with a total stock of M+35, M+10 of which was at the central store, and 5 each at the 5 sub-stores.

The times of demands on sub-stores came from a random number generating stream, which produced random variables from a Poisson Distribution with the mean as specified in the model data. In establishing the average annual cost of operation of the complex, the average over more than one such random number generating streams was taken. This was to ensure that any peculiarities or specialities in the timing of demands did not affect the appraisal of performance of the complex.

6.3 Description of the Simulation for Model One*

6.3.1 General description

The simulation started off with the stock configuration as described in 6.2. Times of the first demand at each sub-store are obtained from

* A similar procedure is adopted for all other experiments.

the Poisson Distribution using the random number generating stream. The clock is updated to the first of these times, the stock level of the sub-store receiving the demand is reduced by 1, and the time until the next demand at this sub-store computed as a random variable from the same Poisson distribution. The time of the next-occurring demand over all sub-stores is now noted and the clock updated to this time, the sub-store stock is reduced by 1, and so on, until either the Free Stock (in the case of the author's control) or System Stock (in the case of Cran's control) reaches the level M, or until a sub-store reaches its reorder level. In the former case, a time equal to the central store lead time hence is associated with the arrival of the procurement for the complex. In the latter case, a delivery of stock to the sub-store is made, the desired delivery quantity being decided by the relevant control rule ('Allocation' or 'Share' in the case of the author's control rules). If the desired quantity exceeds the central store stock, it is made equal to the latter. If the central store is depleted of stock, then the sub-store at its reorder level must wait until stock arrives at the central store before shipment to it is further considered. If, at any time, a sub-store cannot meet a demand, then this demand is 'shelved' or held captive until the store first receives stock, at which time the time period within which the demand was held captive is noted and then multiplied by the cost per unit time of run-out to compute the cost of that particular backorder.

6.3.2 Inventory holding costs

The method by which the inventory holding costs were computed is worth noting. If "h" times the capital tied up in stocks is the annual stock-holding cost then the inventory holding cost over a Y-year period is equal to Yh times capital tied up in inventory. The latter (inventory capital) will equal the cost per item times quantity of stock items held. The quantity of stock held will vary from its maximum at time of procurement arrival to its minimum just before procurement arrival, but its time-averaged value will be the time-averaged value of the total physical stock in the complex (this takes the stock levels of sub-stores with backorders as zero, and is denoted by G). To obtain this from the simulation, the following method was employed:

At the time of the first demand on the complex, the G-value before this demand was met was multiplied by the time the complex existed with this G-value. The G-value is then decreased by 1 (since the sub-store stock has been reduced by 1). Whenever a stock change occurs in the complex to affect the G-value of the complex, then this computation is made.

Mathematically, we have:

Y year Inventory Cost

$$= Yh \times \text{Cost of Unit Stock} \times \text{Average G-value}$$

$$= Yh \times \text{Cost of Unit Stock} \times \int_{t=0}^{t=250Y \text{ days}} G(t).dt \,/250Y$$

$$= \frac{h}{250} \times (\text{Cost of Unit Stock}) \times (\text{Area of G-time function in unit days})$$

6.4 Illustrations of Simulation of the Complex with Data of Model I

6.4.1 "Control and Simulation Language" simulation using author's control rules - Case of "Not Necessarily Replenish" policy

SECTORS: INITL (Initial Sector) plus following activity sectors (ordered list):-

Title of Sector	Description of Purpose of Sector
CWDD	If procurement is due, brings procurement into central store.
DSS	If a sub-store demand is due, demand is made on sub-store.
SYRO	If complex is at its reorder level, order for a procurement is made.
SSRO	If a sub-store is at its reorder level, redistribution is ordered.
SSDD	If a sub-store delivery is due, stock is brought into sub-store.
RESU	If simulation time-of-finish is due, Total Cost of Operation calculated.

The flow diagrams for these sectors are given in Figs. 1 to 6 respectively.

FIG.1

SECTOR CWDD

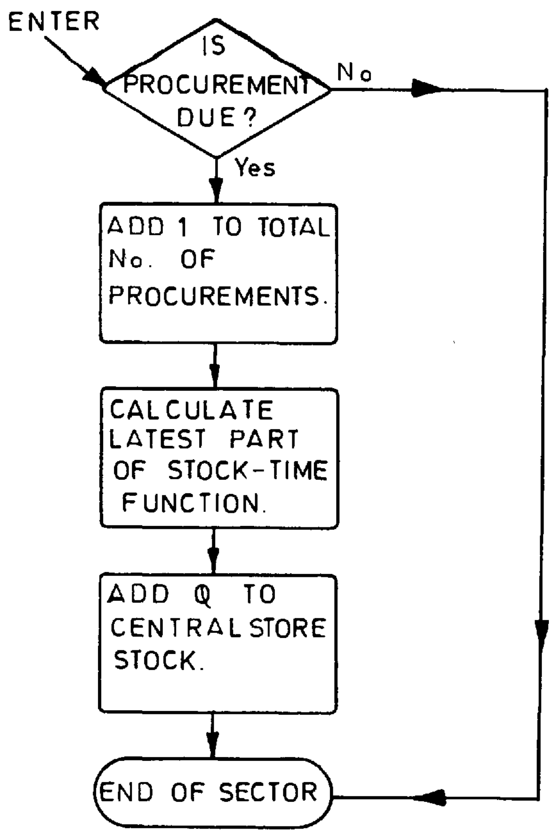


FIG. 2
Sector DSS

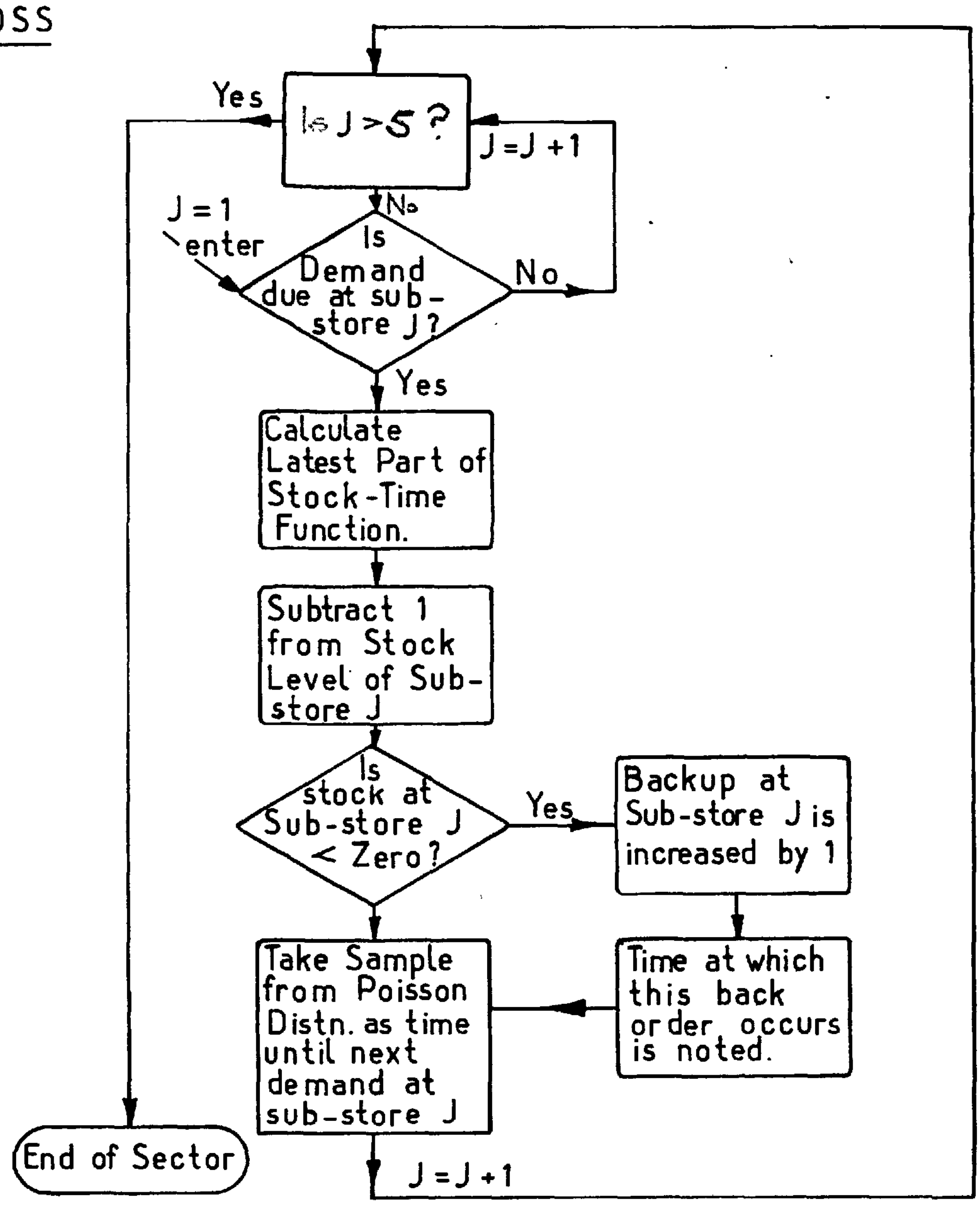


FIG. 3
Sector SYRO

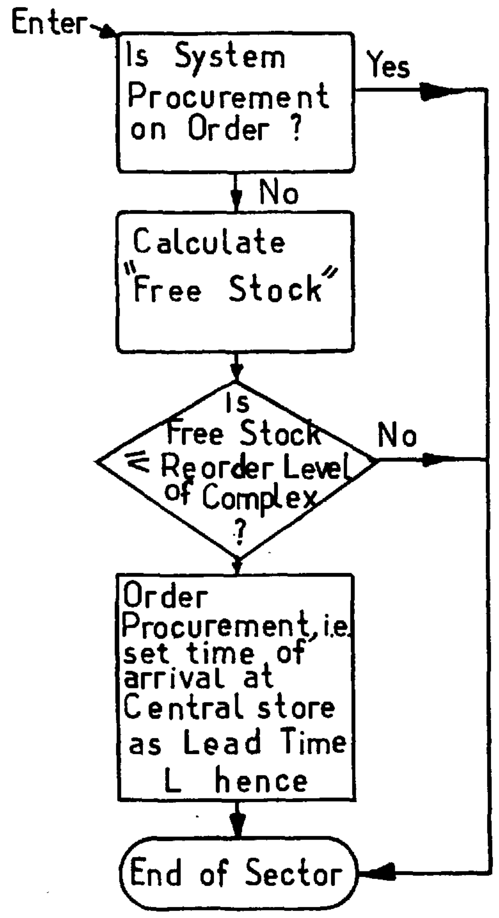


FIG. 4

Sector SSRO

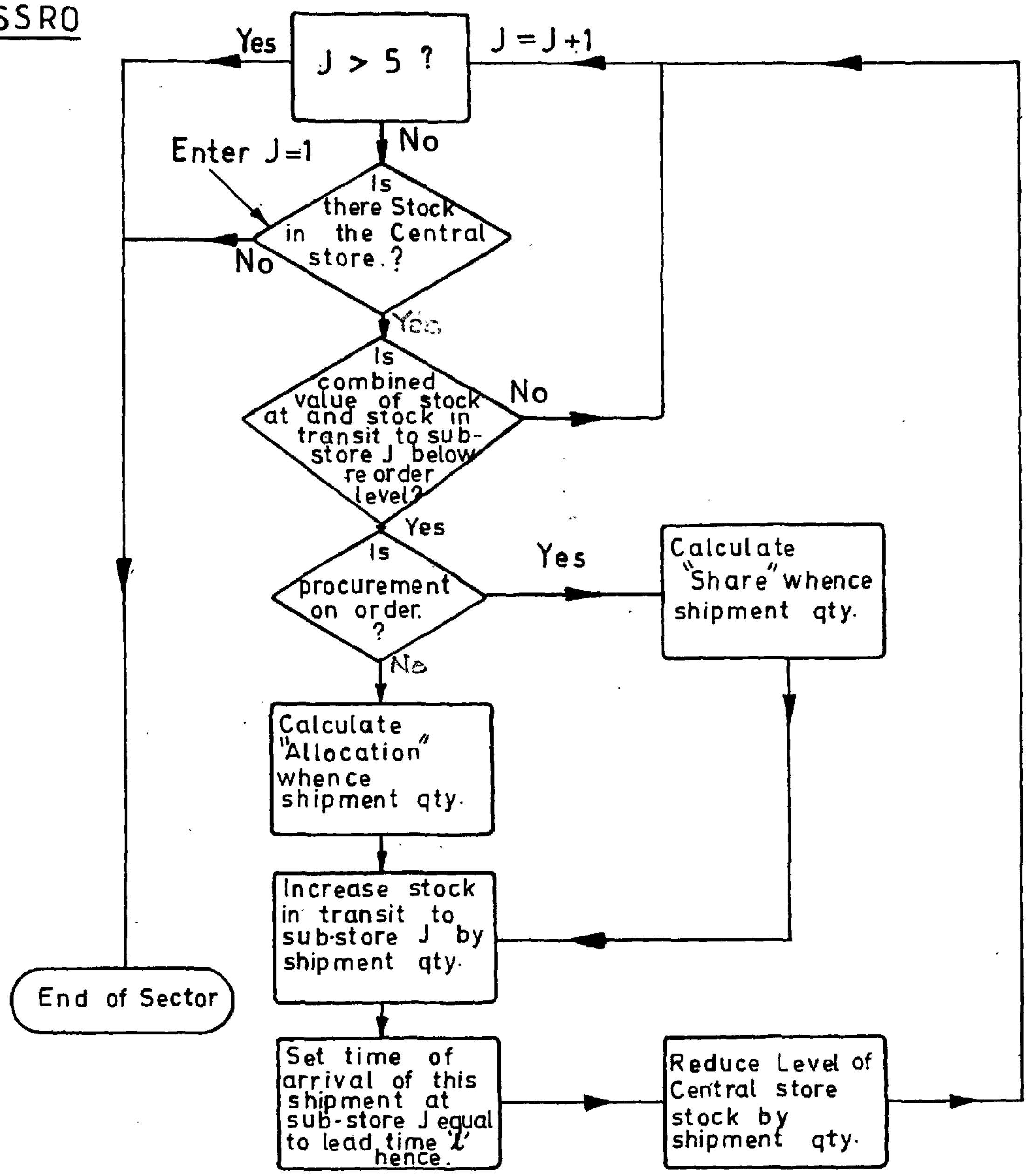


FIG. 5
Sector SSDD

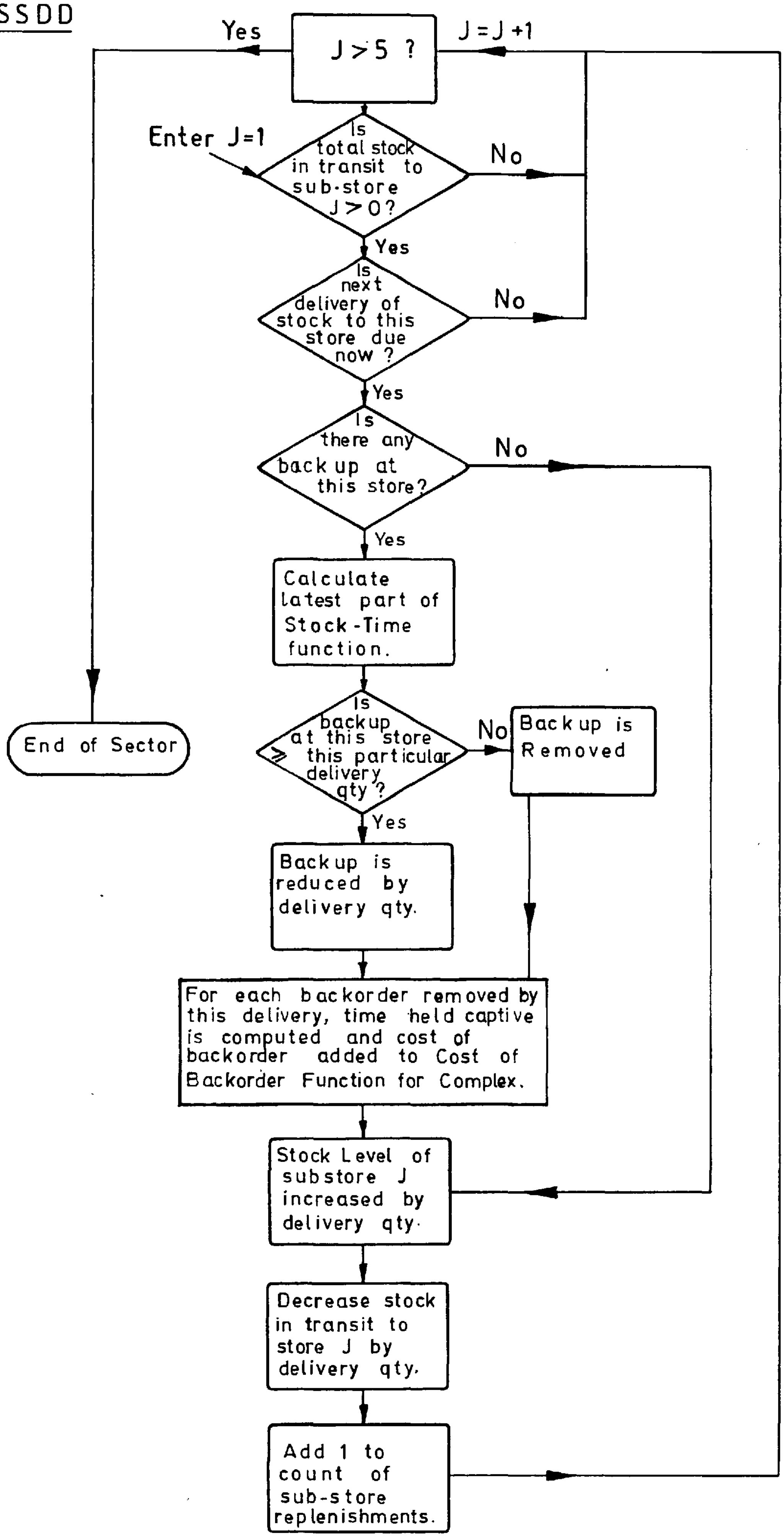
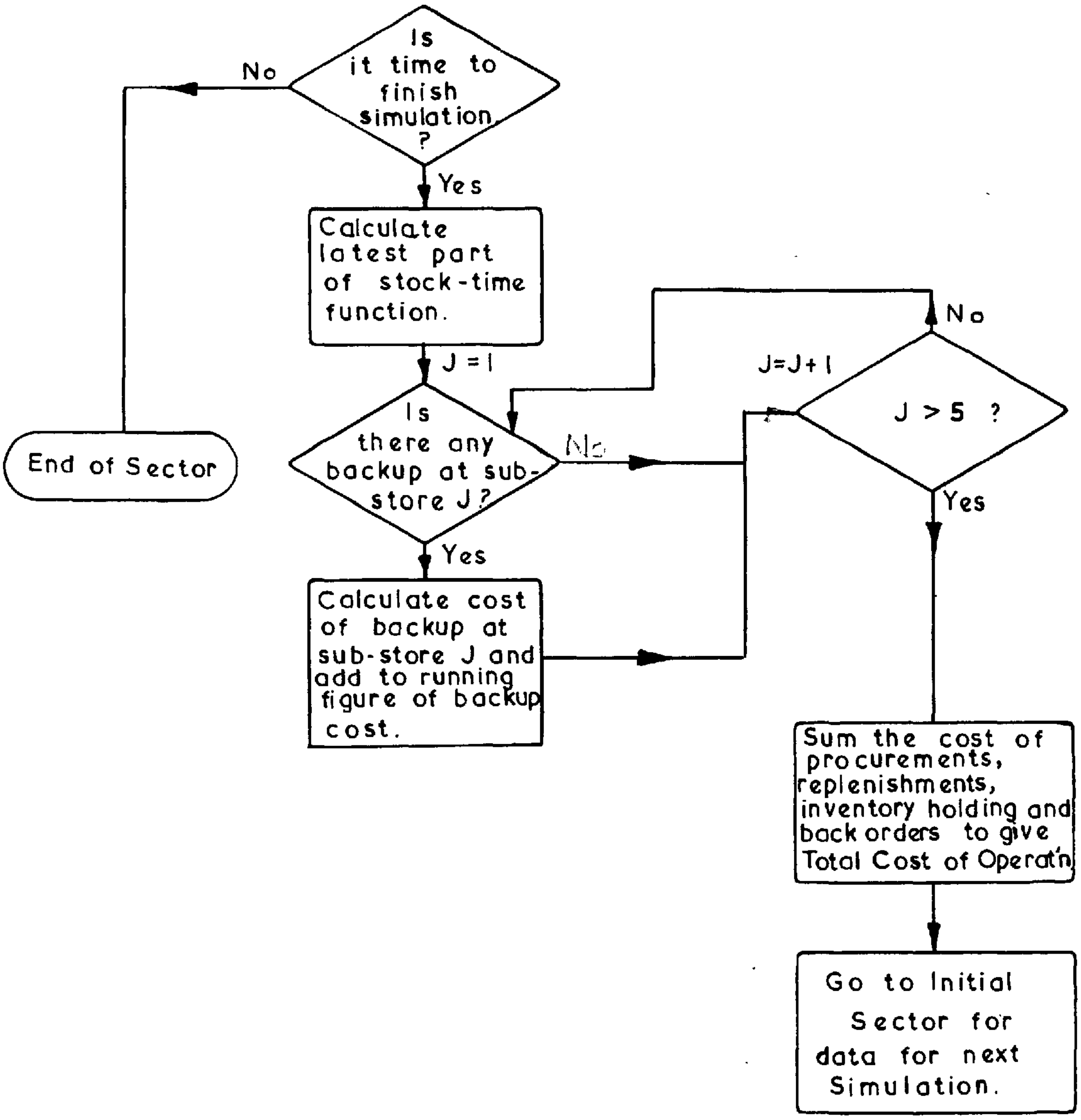


FIG. 6

Sector RESU



6.4.2 The complete program for the C.S.L. simulation for Model I with control 1A (see 6.8)

This is given in Program 1.

6.4.3 The "Simon" simulation program

The flow diagram for the "Simon" simulation program for Model III is very similar to that for the C.S.L. program. The activities CWDD, DSS, SSDD, RESU are considered in the A-phase; the timeset containing times associated with each activity is scanned in the A-phase and the clock updated to the least of these times. The B-phase constitutes the execution of that activity, followed by the C-phase where the activities SYRO and SSRO are considered. Return is then made to the A-phase. As with the C.S.L. program, the initial data necessary to describe the model and commence the simulation is at the beginning of the program, and comes before the A-phase.

6.4.3.1 Simon simulation process

Initial data, including sampling from Poisson distribution to obtain times of first demands on each of substores

A-Phase. Scan timeset containing starting times of activities CWDD, DSS, SSDD, RESU. Find minimum time and update clock to this value

B-Phase. Execute activity with minimum time found in A-Phase (if RESU activity, then simulation is terminated)

C-Phase. Test for SSRO and/or SYRO if activity of B-phase results in a possibility of SSRO or SYRO test being positive

Return to A-Phase.

6.5 Method of Search for Optima of Simulations

At first, a search for low values of the total cost functions for different combinations of the control parameters was made. This indicated those parameter combinations for which the total cost function was expected to be near-optimal. Simulations incorporating these parameter combinations were made, from which further interesting combinations were shown up. For the author's first two models, three random number generating streams were employed each in four-year simulations over which the annual cost of operation was averaged. Convexity of the total cost function with respect to both z and HBF** was generally assumed. Because of simulation noise, it proved impossible to predict the optimum total cost by interpolation between near-optima, and thus a decision had to be taken as to the increments between z values and HBF values to be employed in optima searches.

For Model One, where the optimum value of M turned out to be 35 in the case of the author's control rules (and 40 for Cran's control) the range of F for which the Allocation Rule is expected to be used is 35 to 75. If the increments of z are i, then the differences between allocations are, on average, $i\sqrt{.24(F-35)+4}$ ranging from 3.7i at F = 75* through 3.0i at F = 55 to 2.0i at F = 35+. If corresponding increments of HBF are j then the differences between Cran's allocation quantities are $j\sqrt{\text{System Stock}}$ ranging from $j\sqrt{80}$ = 8.9j at System Stock = 80* to 6.3j at System Stock = 40.

The j-values in the optimum region were made 0.1 (i.e., increments of HBF of 0.1 were considered). In order to make the search for the optimum for the author's control rules with about the same degree of thoroughness as with Cran's (to ensure that comparisons were meaningful) an i-value of 0.3 (corresponding to increments in z of 0.3) was employed in the optimum region.

6.6 Concept of a Test Stream of Random Numbers

It was noted that, one of the chosen random number streams used for generating time intervals between demands gave rise to a period of rather high demand in the sum of the sub-store and central store lead times (in the case of Model One, 28 demands on sub-stores in a time period of 100 days with an expected demand of 20). Thus the mean performance over three simulations with the same control parameters but different random number generating streams was largely governed by the performance with this particular stream, for these rules which were poor in coping with tendencies of sub-stores to run out were liable to result in heavy run-out costs. Since the mean performance was largely governed by the performance of the complex under rather higher demand than usual, the stream producing the latter was considered as a test stream; that is, the performance of the complex was first considered for this stream alone.

---

* Expected F-value or System Stock value immediately after procurement delivery at central store.

\+ Expected F-value or System Stock value at complex reorder level.

** The Cran hold-back factor (see Appendix 1.3)

6.7 A Note on the Presentation of the Results

The symbols are given in the Summary of Symbols in Appendix Eleven. 'Area' indicates the area under the stock-time function for the length of the simulation.

A breakdown of the costs is given for the lowest-cost combination of parameters.

6.8 Summary of Experiment One

4 Year Simulation of Model I with Controls 1A, 1B, 1C using IBM 1130 System with "Simon" Simulation Language

Model I Description (as detailed in 4.2.1)

Item Value = 1
Cost of Procurement = .5
Cost of Supplying Sub-Store = .3
Cost of Sub-Store Run-out = 40% x Item Value per day
Cost of Stock Holding = 10% Value of Average Stock per year held
Mean Sub-store Demand = 10/year; Poisson distribution
Number of Sub-stores = 5
Working Days/year = 250
Lead Time for Complex = .4 year
Lead Time for Sub-stores = 0

Control Descriptions

All controls have sub-store reorder level of -1, and central store reorder quantities equal to 60. For each control, reorder level is the parameter "M".

Control 1A: Author's suggestions with "Not Necessarily Replenish Policy"

Sub-store Reorder Quantity
Case 1: Procurement on Order : "Share" Ration Rule (see 5.6.4.2)
Case 2: Procurement not on Order: "Allocation" Rule (see 5.6.2.3)

Criterion of Reorder Level for Complex : "Free Stock" (see 5.5.2.2)

Control 1B: Author's suggestions with "Necessarily Replenish Policy"

Sub-store Reorder Quantity)
Criterion of Reorder Level for Complex) as for Control 1A.

Control 1C: Cran's Control Method

Sub-store Reorder Quantity: Cran Allocation Rule (see 3.15)
Criterion of Reorder Level for Complex: "System Stock"

6.8.1 Table 6-1

Control 1A Applied to Model One

Results for Test Stream No.1729

M	z	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{RO}$	Area	T.C.
30	-0.6	3	32	30.0	39472	56.8888
	-0.3	3	27	28.0	39499	53.3996
	+0.0	3	22	24.4	39586	48.3344
	+0.3	3	20	68.4	39808	91.8232
35	-0.3	3	28	0.0	44397	27.6588
	+0.0	3	24	0.0	44397	26.4586
	+0.3	3	22	8.4	44418	33.2672
40	+0.0	3	25	0.0	49397	28.7588
	+0.3	3	23	0.0	49397	28.1588
	+0.6	3	19	4.8	49452	31.7808
	+0.9	3	16	18.8	49627	44.9508

Total Costs for Various Stream Numbers

M	z	1729	1921	1147	Mean
35	-0.3	27.6588	27.0452	27.0196	27.2412
	+0.0	26.4586	26.4452	25.8196	26.2411
	+0.3	33.2672	26.7464	25.5196	28.5107
40	+0.3	28.1588	28.2452	27.8196	28.7045
	+0.6	31.7808	27.7460	26.3196	28.6155

Breakdown of Costs at Optimum

M	z	Stream	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{RO}$	Area	T.C.
35	0.0	1729	3	24	0.0	44397	26.4586
		1921	3	23	0.0	45113	26.4452
		1147	3	23	0.0	43549	25.8196
		Mean	3	23.3	0.0	44353	26.2411

Mean Total Cost 26.24

6.8.1.1 Comments on the application of Control 1A to Experiment One

The optimum value of M for the above control ("not necessarily replenish" policy), with M being considered in 5 unit intervals turns out to be 35. The next lower value of M, viz. 30, results in a heavy cost of run-out. It is interesting to note that this is not because the total demand in the complex in the combined lead time L_C exceeds 30 in any simulation. The maximum demand in this time noted was 28, and the run-out is occurring because of a maldistribution of stock within the complex.

6.8.1.2 Maldistribution Types

It was considered useful to establish a categorization of maldistribution types to enable the workings of the various types of control to be more clearly viewed.

6.8.1.2.1 Maldistribution Type 1

If, at the time of complex reorder, System Stock exceeds Free Stock, then a maldistribution has occurred because an extra cost of stockholding results. This is known as "Maldistribution Type 1".

6.8.1.2.2 Maldistribution Type 2

If a run-out is experienced at a sub-store, as a result of the central store being depleted of stock, and there exists stock in any one or more of the other sub-stores (which could be used to replenish the depleted sub-store, but for the condition prohibiting inter-sub-store replenishments), then a maldistribution of type 2 is said to have occurred.

This type of maldistribution can be seen to stem from two sources in the case of the author's control. The sub-store(s) with stock at the time backorder costs are being incurred at another sub-store can be considered to have been shipped too high a quantity at its last replenishment. If the last shipment quantity was decided by the 'allocation' rule then the chosen z value may well be too high for the M-value being employed.* If the last shipment quantity was chosen by the 'Share rule'+ then the latter, in combination with the other parameters employed, may be responsible for the run-out (whether the latter

---

* Type 2A Maldistribution
\+ Type 2B Maldistribution

is a bad thing will of course depend on the relative costs of run-out and distribution, but in general it is to be expected that it is expedient to try to prevent Maldistribution Type 2).

6.8.1.2.3 Maldistribution Type 3

This type of maldistribution occurs in the cases of non-zero lead time, ℓ, to sub-stores where the sub-store receiving a distribution incurs a demand in time ℓ exceeding the stock level at which the order for stock from the central store was placed. It was considered as a maldistribution because the latter might have been prevented if shipment occurred at a higher reorder level. The probability of this type of maldistribution occurring is easily obtained, if required, from the Cumulative Poisson Distribution.

6.8.1.3 An illustration of the three types of maldistribution

Consider the following stock configurations within the complex

	Sub-store Stocks	Central Store Stock		
Configuration No.1	12, 6, 7, 6, 5	11	F = 44	System Stock = 47
Configuration No.2	9, 6, 7, 6, 5	11	F = 44	System Stock = 44
Configuration No.3	0, 6, 7, 6, 5	20	F = 44	System Stock = 44

Configuration No.1 will incur maldistribution cost of type 1 on inventory holding if such an excess of System Stock over Free Stock F exists at complex reorder level.

Configuration No.2 may appear equivalent, superficially, to Configuration No.3. However, in the case of configuration No.2, maldistribution of type 2 is more likely because we have the likelihood that before any stock from the next procurement can reach sub-stores, any of sub-stores No.2 through 5 may have incurred shortage costs, whilst sub-store No.1 still has a finite stock level. If the sub-store 1 had been allocated a quantity less than 9 (whether this was from the allocation rule - Maldistribution Type 2A, or from the rationing rule - Maldistribution Type 2B) then some or all of these Maldistribution Type Two costs might have been saved.

Sub-store	Lead Time ℓ days	Central Store
4		11
	5 in transit	

The figure above represents a sub-store and its link to the central store. The probability of its run-out in the lead time* is the probability of its experiencing a demand of 5 or more items in ℓ days, which is:

$$\sum_{j=5}^{\infty} e^{-\mu} \frac{\mu^k}{k!} \qquad \text{where } \mu = \lambda_i \ell = 1 \text{ for Model Three}$$

This probability equals $\left(1 - e^{-1} (1 + \mu + \mu^2/2 + \mu^3/3 + \mu^4/4)\right)$

$$= 1 - 3.083/e$$

The associated expected cost of run-out, $c_3\psi(s,\ell)$, is obtained by use of the expression for $\psi(\bar{S},\bar{L},\bar{\lambda})$ given in Appendix 5.

6.8.1.4 Further comments on the results of Control 1A applied to Model I

No type 3 maldistribution was in evidence in the results of Table 6-1, since Model One has the sub-store lead time, $\ell = 0$.

Type 1 Maldistribution was noted for those z-values which were too high for the M-value employed. This was seen to be as low as z = -0.3 for M = 30.

Maldistribution Type 2 was in evidence for M-values of 30 and 35 and for the higher z-values (+0.6 and over) of the M = 40 simulations. For M = 30, M = 35, it was not removable by adjustment to z since low z-values, whilst tending to ensure an evenness of stock distribution amongst sub-stores at complex reorder point, were hampered by the fact that the 'Share' rule would now operate on more sub-stores and at an earlier point in time than for higher z-values. This is a poor situation, because 'Share' used thus is a poor rationing rule. Too high a z-value gets over this difficulty but leads to maldistribution by the time of complex reorder level, and tends to leave too little (if any) stock in the central store for emergency 'Share' distributions. The z-value around 0.0 tends to minimize the sum of these disadvantages, and the optimum turns out to be z = 0.0 at an M-value of 35.

As was expected, for a given M, replenishments tend to drop in number, as z is increased. An interesting phenomenon not disclosed in the results list is that as z is decreased to well below zero for close-to-optimal M, backup increases because of the early application of 'Share' (as mentioned in the previous paragraph).

---

*i.e., of it incurring maldistribution, type 3.

6.8.2 Table 6-2

Control 1B Applied to Model I

Results for Test Stream No. 1729

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
35	-0.3	3	30	2.4	44403	30.2612
	+0.0	3	27	2.0	44402	29.3608
	+0.3	3	25	12.8	44512	41.6048
40	+0.0	3	28	0.0	49397	29.6588
	+0.3	3	25	0.0	49397	28.7588
	+0.6	3	21	5.2	49613	32.8452
	+0.9	3	19	20.4	49737	47.4948

Total Costs for Different Stream Numbers

M	z	1729	1921	1147	Mean
35	-0.3	30.2612	27.8460	27.6196	28.5756
	+0.0	29.3608	27.3452	26.7196	27.8085
	+0.3	41.6048	27.3464	25.8196	31.5903
40	+0.3	28.7588	27.9452	28.7196	28.4745
	+0.6	32.8452	29.4320	29.3288	30.5353

Breakdown of Costs at Optimum

M	z	Stream	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
35	+0.0	1729	3	27	2.0	44402	29.3608
		1921	3	26	0.0	45113	27.3452
		1147	3	26	0.0	43549	26.7196
		Mean	3	26.33	0.7	44355	27.8085

Mean Total Cost 27.81

6.8.2.1 Comments on the application of Control 1B to Model I

The main benefit of "necessarily replenish" policy over "not necessarily replenish" is absent when the sub-store lead time is zero. An attribute to the credit of this former policy is that it is known that just after procurement arrival each sub-store has sufficient stock to last it for an expected time. Therefore the probability of a sub-store needing replenishment immediately after the next complex reorder point (at this time the 'Share' ration rule is likely to be too high) is calculable. Except for low z, this probability will be much lower than with "not necessarily replenish" ruling. This advantage has to be balanced against the disadvantage of replenishing earlier than necessary (additional overall replenishment costs and higher maldistribution costs types 1 and 2A).

The optimum turns out to be (M=35, z=0.0), as for "not necessarily replenish" ruling, whilst the replenishment costs are up on average. However, the run-out costs, on average, are down, leading to an overall total cost result worse than for a "not necessarily replenish" policy.

6.8.3 Table 6-3

Control 1C Applied to Model I

Results for Test Stream No. 1729

M	HBF	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{RO}$	Area	T.C.
35	0.7	3	45	0.0	44397	32.7588
	0.6	3	36	2.8	44404	32.8616
	0.5	3	29	6.4	44413	34.3652
	0.4	3	23	9.6	44421	35.7684
40	0.6	3	40	0.0	49397	32.2588
	0.5	3	29	0.0	49397	28.9588
	0.4	3	25	0.8	49399	29.5596
	0.3	3	20	3.6	49406	30.8624
	0.2	3	18	20.4	49448	47.0792

Total Costs for Various Stream Numbers

M	HBF	1729	1921	1147	Mean
35	0.6	32.8616	29.7452	29.1196	30.5755
	0.5	34.3652	27.6452	27.3196	29.7767
	0.4	35.7684	27.9464	26.1196	29.9448
40	0.5	28.9588	29.9226	29.3196	29.4003
	0.4	29.5596	28.7226	27.8196	28.7006
	0.3	30.8624	27.5226	27.7212	28.7021

Breakdown of Costs at Optimum

M	HBF	Stream	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
40	0.4	1729	3	25	0.8	49399	29.5596
		1912	3	24	0.0	50113	28.7226
		1147	3	23	0.0	48549	27.8196
		Mean	3	24	0.27	49354	28.7006

Mean Total Cost 28.70

6.8.3.1 Comments on the results of the application of Control 1C to Model I

Maldistribution Type One is absent in Cran's control. Although for an M-value of 35, Cran's control can result in zero backup, a high hold-back factor HBF is required, and inevitably high replenishment costs result. The disadvantage of the latter tends to outweigh the advantage of the former, and the optimum is at the next higher value of M, viz., 40, (where zero backup is noted for the three stream simulations at a HBF value of 0.5).

A reduction in HBF to 0.4, whilst resulting in a run-out of 2 item-days for one particular random number stream, leads to an overall lowering of replenishment costs, and it is at this value that the Optimum Total Cost is to be found.

In general with Cran's control, as HBF is reduced, so are replenishment costs but at the expense of higher shortage costs.

6.8.4 Significance of difference between cost of operation for Model One under the different control types

6.8.4.1 Control 1A compared with Cran's Control 1C for Model I

The result due to Cran will be tested here against the author's better control policy, viz. Control 1A, by the "t" significance test.

Control 1A Cost	Control 1C Cost	Difference d_i	Stream No.
26.4586	29.5596	3.1010	1729
26.4452	28.7226	2.2774	1921
25.8196	27.8196	2.0000	1147

The average difference $\overline{d}$ = 2.4595

The value of t is given by $\overline{d}/\surd\{\Sigma(d_i-\overline{d})^2/\overline{N}(\overline{N}-1)\}$ where $\overline{N}$ is the number of observations.

Therefore $t = 10.21$, dof = 2.

The improvement afforded by the author's control "not necessarily replenish" policy over Cran's policy is judged significant at the 0.5% level.

6.8.4.2 The extent to which the above improvement in performance is due to the use of a "not necessarily replenish" policy as opposed to a "necessarily replenish" policy

The 't' significance test will be applied to the results of the author's control with a policy of "necessarily replenish" and to Cran's control to see whether there is a significant difference in this comparison.

Control 1B Result	Control 1C Result	Difference d_i	Stream No.
29.3608	29.5596	0.2288	1729
27.3452	28.7226	1.3774	1921
26.7196	27.8196	1.1000	1147

Average difference $\bar{d}$ = .8987

t is given by $\bar{d}/\sqrt{\{\Sigma(d_i-\bar{d})^2/\bar{N}(\bar{N}-1)\}}$

$= 2.54$, dof = 2.

The improvement is judged significant at the 7% level.

6.9 Conclusions from the Simulation of Model One

The advantages of "not necessarily replenish" as an alternative individual policy over "necessarily replenish" policy for the author's control are obvious in the case of zero lead time so it was not surprising that the former turned in a better performance. The "t" test was also used to test the significance between Cran's control and the author's control with a "not necessarily replenish" policy, and it was encouraging to see that the improvement over Cran's result could be judged significant at as low a level as 0.5%.

It is apparent that further consideration of "necessarily replenish" policies for zero lead time at sub-stores is not really valuable and thus further consideration in this case was dropped.

6.10 Chapter Summary and Introduction to Chapter Seven

This chapter introduces the method of comparison for the author's and Cran's control policies, simulation (by electronic computer) techniques for making possible the comparison are given, and this is followed by a general description of the simulation. Details of flow diagrams for procedure in the C.S.L. Language are presented.

The method of search for the optimum combination of parameters is discussed and the concept of a test stream of random numbers is introduced.

Three control types were considered and described as Experiment One. These are all on Model Number I and consider the operation of the model of the complex under (a) the author's control with a "not necessarily replenish" policy for sub-stores following procurement arrival, (b) the author's control with a "necessarily replenish" policy, and (c) Cran's control. Tables of results for different parameter combinations are given and a breakdown of costs for the minimum cost combination for each control is given. Discussion of the various simulations is accompanied by a categorization of Types of Maldistribution.

The author's control as in (a) proved better than with (b), which in turn gave better performance than Cran's control. "t" -significance tests were applied to the results, and "necessarily replenish" policy with the author's control was judged significantly better than Cran's control at the 7% level. As was to be expected, "not necessarily replenish" policy again improved performance.

In Chapter Seven the data of the first model are altered to be such as to favour Cran's control. Two controls are considered in this chapter and together form Experiment Two. Cran's control is compared with that of the author with a "not necessarily replenish" policy.

CHAPTER SEVEN

MODEL TWO AND EXPERIMENT TWO: COMPARISON OF CRAN'S CONTROL AND AUTHOR'S CONTROL WHEN MODEL DATA FAVOUR CRAN

7.1 Introduction to Model Two

It was recognized that, in Model One, Cran's control tends to be relatively expensive because, compared with the author's control, deliveries to sub-stores tend to be more numerous at the same M-value and the same backup figure, and since individual deliveries at a cost of 0.3 each are rather costly, so the total delivery cost is correspondingly excessive. Working on a System Stock trigger for complex reorder, Cran tends to incur more run-outs than for the same M-value on the author's control. Now, if both run-outs and deliveries are not so costly, Cran's performance will tend to be better. The author's control will still be tending (by means of the ideas behind the 'Allocation Rule') to reduce delivery numbers (although it is admitted that the z-value parameter allows the control to enjoy a not insignificant flexibility in this region) and the Free Stock trigger will act to minimise run-outs; the savings resulting from these tendencies will be smaller than for Model One. It was expected, in addition, that the superiority of a Free Stock trigger over a System Stock trigger would be lessened in the case of high average demand rate.

Thus, in order to test the performance of the author's control against Cran's under conditions favourable to Cran, the changes to the data of Model One are:

Replenishment Costs	: Reduced to 10% original value
	: Now each cost 0.03
Run-out Costs	: Reduced to 25% original value
	: Now each cost 0.1 per item-day shortage
Demand Rate	: Increase in Ratio 2.5:1
	: New Mean Time between sub-store demands: 10 days

The performance of the author's "not necessarily replenish" ruling control and Cran's control are compared in Experiment Two.

7.2 Summary of Experiment Two

4 Year Simulation of Model II with Controls 2A, 2C using
IBM 1130 System with "Simon" Simulation Language

Model II Description (as specified in 7.1)

Item Value = 1

Cost of Procurement = .5

Cost of Supplying Sub-store = .03

Cost of Sub-store Run-out = 10% x Item Value per day

Cost of Stock Holding = 10% Value of Average Stock per year held

Mean Sub-store Demand = 25/year; Poisson Distribution

Number of Sub-stores = 5

Working Days/year = 250

Lead time for Complex = .4 year

Lead time for Sub-store = 0

Control Descriptions

All controls have sub-store reorder level of -1, and central store reorder quantities equal to 85. For each control, reorder level is the parameter "M".

Control 1A: Author's suggestion with "Not Necessarily Replenish Policy"

Sub-store Reorder Quantity:

Case 1: Procurement on Order: "Share" Ration Rule (see 5.6.4.2)

Case 2: Procurement on Order: "Allocation" Rule (see 5.6.2.3)

Criterion of Reorder Level for Complex: "Free Stock" (see 5.5.2.2)

Control 1C: Cran's Control Method

Sub-store Reorder Quantity: Cran Allocation Rule (see 3.15)

Criterion of Reorder Level of Complex: "System Stock"

7.2.1 Table 7-1

Control 1C Applied to Model II

Results for Test Stream No. 1729

M	HBF	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{RO}$	Area	T.C.
60	0.7	6	122	1.6	49783	28.1731
	0.6	6	105	1.7	49784	27.7635
	0.5	6	91	1.6	49783	27.2431
	0.4	6	72	2.1	49788	27.1751
65	0.5	6	74	0.0	54767	27.0268
	0.4	6	61	0.0	54767	26.6368
	0.3	6	55	0.7	54774	27.2596
	0.2	6	47	2.9	54796	29.2284

Total Costs for Different Stream Numbers

M	HBF	1729	1921	1147	Mean
65	0.6	27.4468	27.6832	26.9479	27.3593
	0.5	27.0268	27.3232	26.6479	26.9993
	0.4	26.6368	27.0232	27.8943	27.1848
60	0.7	28.1731	26.4844	29.3616	28.0064

Breakdown of Costs at Optimum

M	HBF	Stream	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
65	0.5	1729	6	74	0.0	54767	27.0268
		1921	6	71	0.0	55483	27.3232
		1147	6	74	0.1	53320	26.6479
		Mean	6	73	0.03	54523	26.9993

Optimal Mean Total Cost 27.00

7.2.2 Table 7-2

Control 1A Applied to Model II

Results for Test Stream No. 1729

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
65	-0.3	6	54	0.0	54767	26.5262
	+0.0	6	48	0.3	54770	26.6480
	+0.3	6	44	3.1	54798	29.3392

Total Costs for Different Stream Numbers

M	z	1729	1921	1147	Mean
65	-0.3	26.5262	27.3556	26.3791	26.7536
	+0.0	26.6480	27.0649	25.9879	26.5669

Breakdown of Costs at Optimum

M	z	Stream	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
65	0.0	1729	6	48	0.3	54770	26.6480
		1921	6	49	0.4	55487	27.0649
		1147	6	52	0.1	53320	25.9879
		Mean	6	49.7	0.27	54526	26.5669

Optimal Mean Total Cost 26.57

7.3 Comments on Model Two

With Model Two, neither Cran nor the author were expected to achieve a good performance with an M-value of 60, since in the central store lead time a demand of 62 leading to a minimum achievable backup of 16 item-days was experienced with the random number generating stream 1729. At the next higher M-value, viz. 65, at a HBF value of 0.5, Cran was able to cut average backup costs to 0.03, while incurring average replenishment costs of 2.19 (corresponding to 73 sub-store deliveries on average). Lowering the HBF factor to 0.4 resulted in delivery total savings but backup costs rose. Cran's optimum was for an HBF of 0.5, at which parameter, costs of 27.00 in total resulted.

At the same M-value of 65, the author's control incurred a small backup cost for each random number generating stream at the optimal z of 0.0, yet this was for an average sub-store replenishment number of 50. Although unit delivery costs were small in this Model, the cost savings resulting from the savings in delivery totals was significant enough to establish a performance superior total-cost wise to that of Cran's control.

As with Model One, reduction of z from the optimum leads to higher run-out costs of maldistribution type 2B and an increase in z beyond the optimum leads to higher maldistribution costs type 2A.

7.4 Significance Testing the Results

The t-test is applied to the results for the optima

Author's Result	Cran's Result	Difference d_i	Stream No.
26.6480	27.0268	0.3788	1729
27.0649	27.3232	0.2583	1921
25.9879	26.6479	0.6600	1147

$$t = \bar{d} \div \sqrt{\{\Sigma(d_i-\bar{d})^2/\bar{N}(\bar{N}-1)\}} = 3.63 \qquad \text{dof} = 2$$

The improvement afforded by the author's control is judged significant at the 5% level.

7.5 Conclusions from Experiment Two

It is felt that the Rationing Rule "Share" can lend itself to improvement but the next issue will be the consideration of how the author's control compares with Cran's in the case of non-zero lead time at sub-stores, the more general case.

7.6 Chapter Summary and Introduction to Chapter Eight

This chapter has considered the application of both the author's and Cran's control suggestions to a model of the complex with data specifically designed to favour Cran's control - Model Two. It is encouraging to see that the improvement afforded by use of the author's suggestions is significant at the 5% level.

The next chapter deals with the more general problem of non-zero lead time at sub-stores.

CHAPTER EIGHT

MODEL THREE AND EXPERIMENT THREE: COMPARISON OF CRAN'S CONTROL WITH AUTHOR'S TWO ALTERNATIVE TYPES OF CONTROL FOR THE CASE OF NON-ZERO SUB-STORE LEAD TIME

8.1 The Problem of Sub-store Reorder Level in the Case of Non-Zero Sub-store Lead Time

The new problem arising in considering a model with non-zero sub-store lead time is that of determining the reorder level of sub-stores. Whereas, in the case of zero lead time, backup was only like to occur if the sub-store reached its reorder level and the central store was empty at the same time, in this case backup will occur if the sub-store's demand in the lead time exceeds the stock level at which it reorders. The latter, it will be remembered, results in Maldistribution Type 3 costs.

Initially, the sub-store reorder level was made a fixed quantity for all stock configurations; i.e., it was made an additional control parameter.

8.2 Model III and the 'Allocation' Rule

Model III is identical to Model I with the exception that sub-store lead time is now 25 days. The coverage time demand on a sub-store is assumed* to be distributed thus:-

Mean $= .2(F-M) + \lambda_i L_C$

$= .2(F-M) + 5$ for Model III

Variance $= .24\,(F-M) + \lambda_i L_C$

$= .24(F-M) + 5$ for Model III

Allocation Quantity is then:

$$.2(F-M) + 5 + z\sqrt{(.24(F-M)+5)}$$

for Model III

8.3 A Note on the Simulation for Experiment Three

The simulation runs in this experiment were carried out for a simulated period of 10 years and the averaged 10 year total cost was obtained from the mean of four runs with different random number generating streams being employed for sampling times between sub-store demands from the Poisson distribution relevant to this model.

8.4 A Note on the Parameter Intervals Considered in the Search for Optima of Experiment Three

The hold-back factor interval considered in the optimal regions was 0.05, and the corresponding z interval to be employed to make the searches

---

* Appendix One shows how this result is established.

directly comparable was 0.1, calculated in the same way as for models One and Two (as indicated in 6.5).

8.5 Summary of Experiment Three

10 Year Simulation of Model III with Controls 2A, 2B, 2C
using IBM 7090 Computing System with "C.S.L." Simulation Language

Model III Description

Item Value = 1
Cost of Procurement = .5
Cost of Supplying Sub-store = .3
Cost of Sub-store Run-out = 40% x Item Value per day
Cost of Stock Holding = 10% Value of Average Stock per year held
Mean Sub-store Demand = 10/year; Poisson Distribution
Number of Sub-stores = 5
Working days/year = 250
Lead Time for Complex =.4 year
Lead Time for Sub-stores = 25 days

Control Decisions

All controls have a parameter "ROL_{SS}" for sub-store reorder level, and central store reorder quantity equal to 60. For each control, reorder level is the parameter "M".

Control 2A: Author's suggestions with "Not Necessarily Replenish Policy"

Sub-store Reorder Quantity

Case 1: Procurement on Order: "Share" Ration Rule (see 5.6.4.2)
Case 2: Procurement not on Order: "Allocation" Rule (see 5.6.2.3)

Criterion of Reorder Level for Complex: "Free Stock"

Control 2B: Author's suggestions with "Necessarily Replenish Policy"

Sub-store Reorder Quantity)
Criterion of Reorder Level for Complex) as for Control 2A

Control 2C: Cran's Control Method

Sub-store Reorder Quantity: Cran Allocation Rule (see 3.15)

Criterion of Reorder Level of Complex: "System Stock"

8.5.1 Table 8-1

Control 2A Applied to Model III

Results for Test Stream No.1115

M	ROL_{SS}	z	$\overline{N}_R$	$\overline{C}_{RO}$	T.C.
40	2	0.0			96.29
		0.1			100.59
		0.2			98.95
		0.5	71	16.0	91.09
		0.6	60	37.2	110.69
40	3	-0.2	105	11.2	95.50
		0.0	93	6.0	86.98
		0.1	94	0.0	81.49
		0.2	85	5.2	84.29
		0.3	79	1.6	78.59
		0.4	76	9.2	85.30
40	4	0.1			93.10
		0.2			91.60
		0.3			87.34
45	2	0.5	69	49.6	127.92
		0.6	67	34.0	111.71
		0.7	60	22.0	98.51
		0.8	61	14.8	91.62
		0.9	57	34.4	111.77
		1.0	54	29.2	105.93
45	3	-0.2	111	4.4	95.28
		0.0	98	0.4	87.37
		0.2	87	0.4	84.07
		0.3	85	2.8	86.07
		0.4	78	0.4	87.80
		0.5	80	2.8	84.93
		0.6	75	13.2	93.91
		0.7	68	2.4	81.50
		0.8	62	4.4	82.09
		0.9	59	16.0	94.16

cont..

Results for Test Stream No.1115(cont.)

M	ROL_{SS}	z	$\bar{N}_R$	$\bar{C}_{RO}$	T.C.
45	4	-0.2	128	3.2	99.18
		0.0	116	4.4	96.78
		0.2	102	2.4	90.86
		0.6	84	1.2	84.81
		0.7	75	0.0	81.42
		0.8	70	0.0	80.18
		0.9	63	1.2	90.79
		1.0	58	15.6	96.03

Total Costs for Different Streams

M	ROL_{SS}	z	1115	1729	1147	1921	Mean
40	3	0.1	81.49	95.56	82.64	86.05	86.43
		0.3	78.59	92.10	78.87	83.47	83.26
45	3	0.4	81.80	90.45	82.01	87.65	85.48
		0.7	81.50	83.44	78.04	78.76	80.43
		0.8	82.09	81.23	77.38	86.29	81.75
45	4	0.7	81.42	86.79	81.04	80.60	82.31
		0.8	80.18	83.44	80.13	81.66	81.35

Breakdown of Costs at Optimum: Average over 4 Streams

M	z	ROL_{SS}	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{RO}$	Average Stock	T.C.
45	0.7	3	7.75	61	1.6	56.66	80.43

Optimal Mean Total Cost 80.43

8.5.2 Table 8-2

Control 2B Applied to Model III

Results for Test Stream No.1115

M	ROL_{SS}	z	$\overline{N}_R$	$\overline{C}_{RO}$	T.C.
40	2	0.0	87	21.2	100.49
		0.1	82	18.8	96.81
		0.2	77	3.2	79.81
		0.3	73	3.2	78.80
		0.4	70	7.2	82.48
		0.5	66	27.2	102.24
		0.6	61	54.0	127.80
40	3	-0.2	107	4.4	89.29
		+0.0	93	4.0	85.08
		0.1	91	7.2	87.90
		0.2	87	3.2	82.81
		0.3	81	3.6	81.60
		0.4	75	7.6	84.38
40	4	-0.2	129	9.2	100.70
		+0.0	111	0.4	87.27
		0.1	-	-	99.10
		0.2	91	15.2	96.02
		0.4	76	25.6	102.84
45	2	0.4	72	2.8	82.80
		0.5	68	3.6	82.51
		0.6	65	9.2	87.41
		0.7	62	7.6	86.18
		0.8	58	16.0	93.74
		0.9	53	25.2	103.81
		1.0	57	28.8	109.42
45	3	-0.2	111	3.6	94.48
		0.0	100	4.4	91.98
		0.2	89	0.4	85.27
		0.3	81	0.0	82.40
		0.4	80	0.0	82.39
		0.5	77	0.0	81.61
		0.6	73	0.0	80.60
		0.7	65	0.0	79.47
		0.8	61	11.2	88.65

cont..

Results for Test Stream No. 1115 (cont.)

M	ROL_{SS}	z	$\bar{N}_R$	$\bar{C}_{RO}$	T.C.
40	4	-0.2	130	2.4	98.97
		+0.0	119	3.2	96.48
		0.2	97	0.0	87.27
		0.4	81	0.0	85.39
		0.6	-	-	84.85
		0.7	-	-	80.21
		0.8	-	-	90.46

Total Costs for Different Random Number Streams

M	ROL_{SS}	z	1115	1729	1147	1921	Mean
40	2	0.2	79.81	101.91	87.28	76.62	86.40
		0.3	78.80	99.76	84.78	77.25	85.15
40	3	0.2	82.81	84.94	82.97	77.62	82.08
		0.3	81.60	81.08	83.07	77.55	80.82
		0.4	84.38	93.80	89.57	80.16	86.98
40	4	0.1	99.10	104.36	82.09	81.86	91.35
45	2	0.5	82.51	92.10	89.88	80.83	86.33
45	3	0.6	80.60	88.64	82.23	83.40	83.72
		0.7	79.47	80.08	83.06	85.01	81.90
45	4	0.6	84.85	84.15	84.63	83.91	84.39
		0.7	80.21	89.77	84.58	83.74	84.57
		0.8	90.46	86.61	86.64	83.57	86.07

Breakdown of Costs at Optimum: Average Over 4 Streams

M	z	ROL_{SS}	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{RO}$	Average Stock =Holding Cost	T.C.
40	0.3	3	7.75	70.5	3.1	52.70	80.82

Optimal Mean Total Cost 80.82

8.5.3 Table 8-3

Control 2C Applied to Model III

Results for Test Stream No. 1115

M	ROL_{SS}	HBF	$\overline{N}_R$	$\overline{C}_{RO}$	T.C.
40	3	0.2	74	17.6	92.39
		0.3	101	4.4	87.28
		0.4	126	11.2	101.58
		0.5	135	43.2	136.32
40	4	0.2	82	47.6	124.82
		0.3	116	11.2	98.58
45	3	0.1	58	21.6	96.59
		0.2	70	9.2	87.78
		0.25	84	1.2	83.97
		0.3	91	0.4	85.27
		0.4	105	4.0	93.08
		0.5	123	10.0	104.48
45	4	0.1	57	22.8	97.49
		0.15	-	-	78.57
		0.2	75	1.2	81.27
		0.25	-	-	86.07
		0.3	103	0.0	88.47
50	3	0.15			82.37
		0.20			84.27
		0.25			84.37
		0.30			86.37
		0.35			91.22
		0.40			90.77
50	4	0.20			83.57
		0.30			90.77
		0.40			94.37

Total Costs for Different Random Number Generating Streams

M	ROL_{SS}	HBF	1115	1729	1147	1921	Mean
45	3	0.25	83.97	106.84	130.98	79.72	100.38
45	4	0.15	78.57	121.66	165.02	76.72	110.49
		0.20	81.27	86.42	131.18	79.12	94.50
		0.25	86.07	89.32	132.08	81.22	97.17
50	3	0.15	82.37	85.32	116.07	79.92	90.92
		0.20	84.27	87.12	99.25	81.12	87.94
		0.25	84.27	85.02	100.15	82.02	87.86
		0.30	86.37	86.52	92.64	84.12	87.41
		0.35	91.22	88.73	88.37	85.32	88.41
		0.40	90.77	93.92	88.03	87.42	90.03
		0.50	-	99.02	-	91.43	-
50	4	0.20	83.57	-	100.95	-	-
		0.30	90.77	-	96.44	-	-
		0.40	94.37	-	91.83	-	-

Breakdown of Costs at Optimum: Average Over 4 Streams

M	HBF	ROL_{SS}	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Average Stock	T.C.
50	0.30	3	7.5	68	2.7	60.56	87.41

Optimal Mean Total Cost 87.41

8.6 Comments on Model Three and Experiment Three

Cran's control is unable to prevent backup with modest use of buffer stock. Low HBF tends to result in high initial deliveries, reducing central store stock appreciably to begin with (after procurement arrival). High HBF tends to result in, as well as high delivery numbers and hence costs, more sub-stores than otherwise requiring a second (or even third) delivery before next procurement arrival. This means an overall large number of times that reorder level is experienced by sub-stores, and hence a corresponding larger chance of run-out, overall, than otherwise. Cran's optimum requires a buffer stock of 35, corresponding to an M-value of 0.3. Raising HBF over this value results in higher delivery costs, and lowering HBF incurs a heavier contribution to run-out. Cran's sub-store reorder level at optimum was 3. A level of 2 led to heavy run-out costs in the lead time, whilst a level of 4 although reducing the effect of run-out of Maldistribution Type 3 in the lead time, inevitably led to Maldistribution Type 2, by replenishing too early.

As with Cran's control, the author found that for both his "necessarily replenish" and "not necessarily replenish" policies, optimal balance between the Maldistribution Types 2,3 was obtained for a sub-store reorder level of 3.

A shift in the optimal z (compared with the author's control for zero sub-store lead time) was noted for both types of policy considered. The shift was upwards (to 0.7 for "not necessarily replenish" and 0.3 for "necessarily replenish"). At these z-values, application of the 'allocation' rule did not lead to excessive maldistribution of type 2A, and the number of times reorder level was experienced (and hence maldistribution costs type 3) was cut, compared to close-to-zero z-values. For the case of "not necessarily replenish", the author required an M-value of 45, corresponding to an overall buffer stock of 20 for his optimum. Whatever the (z, sub-store reorder level) combinations applied to an M-value of 40, run-out led to excessive costs. In the case of a "necessarily replenish" policy, run-out costs of maldistribution type 3 were cut down by the reduction in number of times sub-stores experienced running-down to their reorder levels, but, of course, at the expense of maldistribution type 2A costs. The optimal M-value was found to be lower than for a "not necessarily replenish" policy.

As was to be expected, the allocation parameter, z, was not able (without excessive costs) for a "necessarily replenish" policy to be as high as a "not necessarily replenish" policy. At optimal it was 0.3, compared with 0.7, with, as expected, higher overall replenishment costs. Additionally, Maldistribution Type One was experienced for both types of the author's control at their optima, and, as would be expected, was greater for the case where lower buffer stock was retained (viz., the "necessarily replenish" policy case).

8.7 Testing to See if the Observed Improvement over Cran is Significant

The improvement of the "not necessarily replenish" policy over the "necessarily replenish" policy is obviously insignificant. Here the t-test is applied to the results of the author's control with the former policy and to the results of Cran's control to see whether these two results are "significantly" different.

Control 2A	Control 2C	d_i	Stream No.
81.50	86.37	4.87	1115
83.44	86.52	3.12	1729
78.04	92.64	14.60	1147
78.76	84.12	5.36	1921

The value of t is given by $\overline{d}/\sqrt{\{\Sigma(\overline{d}-d_i)^2/\overline{N}(\overline{N}-1)\}}$ where $\overline{N}$ is the number of observations.

Hence t = 4.58, dof = 3.

The improvement afforded by the author's control is judged significant at the 1% level.

8.8 Summary of Chapter and Introduction to Chapter Nine

This chapter has considered the application of the author's control (and also Cran's control) to a more general model of the complex in which the lead time to sub-stores is non-zero. The latter modification to the model results in an additional consideration, that of sub-store reorder level. This, in the first instance, has been made an extra control parameter.

The simulations considered constitute Experiment Three and the results indicate that there is little to choose between the two types of policy considered by the author when applied to this model. However, the improvement of either over Cran's control is significant at as low a level as 1%.

To avoid the involvement of an extra parameter for a reorder level at the sub-stores, an attempt at some analysis to establish a policy for this problem was made for those instants when the order for the complex is outstanding. This work is described in the next chapter along with some experiments to see whether performance is improved using the newly adopted approach.

CHAPTER NINE

INTRODUCTION TO THE FIRST DYNAMIC PROGRAMMING MODEL TO ESTABLISH THE REORDER LEVEL OF SUB-STORES AS A FUNCTION OF TIME TO PROCUREMENT ARRIVAL AT THE CENTRAL STORE (FOR THE CASE OF PROCUREMENT ON ORDER). INCLUDES THREE EXPERIMENTS TO IMPROVE OVERALL CONTROL USING THIS MODEL.

9.1 Introduction

For the case when a procurement to the complex is on order, the problem of when to order a shipment of stock to a sub-store may be resolved into the problem of balancing the cost of run-out at the sub-store by replenishing it too late (as a result of maldistribution type 3) against the cost of a delivery which if replenishment does not occur now, may be saved. If delivery (a replenishment) occurs whilst the procurement is on order, the delivery quantity is small since the Free Stock is low. Hence it is likely that a delivery will again be necessary shortly after the procurement arrives.

On the other hand, if replenishment is postponed until after the procurement arrives, then it follows that the cost of a replenishment is saved. There is the further consideration that the expected cost of run-out at other sub-stores is not quite independent of the decision whether to replenish any particular sub-store or not, since the latter decision will affect the availability of stock to buffer the other sub-stores' stocks. Maldistribution type 2B, with some expected associated cost, is liable to result in general. In order to establish a general decision method for the reorder level of sub-stores whilst a procurement order is outstanding, the maldistribution type 2B costs will be ignored.

9.2 Establishment of the Dynamic Programming Decision Method

9.2.1 Costs associated with replenishing a sub-store and postponing replenishment

The problem in 9.1 is resolved as follows: consider a notional stock level at the sub-store under consideration equal to s (this will equal actual sub-store stock plus any stock in transit to that sub-store) whilst the time until the procurement is delivered at the central store is T.

There are two alternative decisions for any (s,T) couple; either a decision to replenish the store at that time instant, or a decision to wait until the next time instant before considering the situation again. Let the cost of the former decision be $C_A(s,T)$ and of the latter, $C_B(s,T)$. The cost of the better of the two decisions, to be known as the "Cost of the Best Decision", will be denoted by $C(s,T)$.

Then $C(s,T) = \text{Min}\ \{C_A(s,T);\ \ C_B(s,T)\}$

If, of course, $C(s,T) = C_A(s,T)$, then it is cheaper to replenish now and the Decision Function associated with the decision is unity, i.e. $D(s,T) = 1$. If $C(s,T) = C_B(s,T)$, then the decision is to wait until the next time instant before review, the associated decision function being zero, i.e. $D(s,T) = 0$.

If the expected number of item-days of shortage for a sub-store with a notional stock level s when the decision is made to replenish is denoted by $\psi(s,\ell)$, ℓ being lead time from central store to sub-store, and the cost per item-day of shortage is c_s, then we have:-

$$C_A(s,T) = c_s\psi(s,\ell) + c_R$$

where c_R is the cost of a sub-store replenishment.

$C_B(s,T)$ is the cost of the decision to wait before further review until the next time instant. Strictly, this is an infinitessimal time δT hence, in which time the sub-store sustains an expected demand of $\lambda_i\delta T$, where λ_i has its usual meaning in this thesis, that is, the mean rate of demand per unit time on a sub-store. Now assuming that in this time δT, no more than one demand may be received at any one sub-store, we have two states which may exist at time δT, viz.:-

state 1 $(s,T-\delta T)$
state 2 $(s-1, T-\delta T)$

The probability of occurrence of state 1 is $(1-\lambda_i\delta T)$ and that of state 2 is $\lambda_i\delta T$.

Hence
$$C_B(s,T) = (1-\lambda_i\delta T)\, C(s,T-\delta T) + (\lambda_i\delta T)\, C(s-1, T-\delta T) \qquad (1)$$

For the purposes of simulation where the individual time units are in days, equation (1) may be modified thus:-

$$C_B(s,T) = (1-\overline{\lambda}_i)C(s,T-1) + \overline{\lambda}_i C(s-1,T-1)$$

where $\overline{\lambda}_i$ now represents mean sub-store rate of demand per day.

9.2.2 Boundary conditions

To establish the dynamic programming method, it is necessary to have knowledge of the bounding conditions, i.e. what the cost of the best decision is when T=0. For the case of "necessarily replenish" ruling, sub-store deliveries are always ordered immediately the central store receives its procurement at T=0.

Then we have:

$$C(s,0) = c_s\,\psi(s,\ell) \quad \text{for "necessarily replenish" ruling.}$$

In the case of "not necessarily replenish" ruling, no decision to replenish sub-stores is taken as a matter of course at T=0. However, the expected stock level of a sub-store at T=0 will, in general, be sufficiently low for the sub-store to require a replenishment in the next cycle (i.e. before further stock arrives at the central store from the next-ordered procurement). It is not an unreasonable assumption, then, that the cost of the best decision at T=0 is equal to the cost of replenishing at that time. (Analysis formulated later for the case of the model data of Model Three shows that for the expected sub-store stocks at time of procurement arrival within the complex, deliveries will be required immediately to be sent out to the sub-stores.)

Hence $C(s,0) \;=\; c_s\psi(s,\ell)$

also for the case of "not necessarily replenish" ruling.

The nature of $\psi(s,\ell)$ is described in Appendix No.5. Appendix No.6 tabulates the function $\psi(s,\ell)$ for a particular case and shows how the decision functions are calculated for a few interesting cases.

9.2.3 Decision functions of (s,T) states from the dynamic programming method applied to Model III

For Model III, $c_R/\lambda_i c_s \ell \;=\; .3/(.04 \times .4 \times 25) = .75$ and the types of calculation illustrated in Appendix No.6 give for all T, D(0,T) = 1.

The other decision functions over the region for which T is defined as calculated are:-

s	T	D(s,T)
1	1	0
	2 → 100	1
2	1 → 6	0
	7 → 100	1
3	1 → 22	0
	23 → 100	1
4	1 → 61	0
	62 → 100	1
5	1 → 100	0

9.3 The Problem of Sub-store Reorder Level for the Case when Complex Procurement is not on Order

The large amount of computing done for Experiment Three (described in Chapter Eight) owing to the extra control parameter (viz. sub-store reorder level) convinced the author of the value of eliminating the latter in subsequent work.

When a procurement is not on order, we do not know the time until the central store receives further stock. However, the time until the complex orders a procurement can be estimated and hence an estimated T (time until procurement arrival in central store) can be obtained. If the Free Stock level at any time is F, with complex reorder M, then, the estimated time until complex reorder is $(F-M)/\lambda_T$, λ_T representing, as usual, the mean rate of demand on the complex. The estimated T is thus $(F-M)/\lambda_T + L$.

The range of considered T was 0-400. Within this range, the decision functions for s-values from 1 upwards were calculated and the complex was simulated using the obtained decision functions for sub-store reorder level control and employing the data of Model III. The simulation is known as Experiment 4 and embraced both the author's alternatives of "not necessarily replenish" ruling and "necessarily replenish" ruling.

9.4 Summary of Experiment Four

10 Year Simulation of Model III with Controls 3A, 3B
Using IBM 7090 System with "C.S.L." Simulation Language

Full Description of Model Given in 8.5

Control Decisions

All controls have a central store reorder quantity equal to 60, and have reorder level as a parameter "M".

Control 3A: Author's suggestions with "Not Necessarily Replenish Policy"

Sub-store Reorder Quantity
Case 1: Procurement on Order: "Share Ration Rule" (see 5.6.4.2)
Case 2: Procurement not on Order: "Allocation Rule" (see 5.6.2.3)

Criterion of Reorder Level for Complex: "Free Stock"

Reorder Level for Sub-stores

Case 1: Procurement on Order: "1st D.P. Model Decision Rule" (see 9.2)
Case 2: Procurement not on Order:

An estimate of the T parameter in 1st D.P. Model is made and the decision rule worked on this T.

Control 3B: Author's suggestions with "Necessarily Replenish Policy"

Sub-store Reorder Quantity)
)
Criterion of Reorder Level for Complex) as for Control 3A.
)
Reorder Level for Sub-Stores)

9.4.1 Table 9-1

Control 3A Applied to Model III

Results of Test Stream No.1115

M	z	$\bar{N}_R$	$\bar{C}_{RO}$	T.C.
40	0.3	82	31.2	109.53
	0.4	81	26.4	64.43
	0.5	70	4.0	79.57
	0.6	60	63.2	137.89
45	0.4	83	0.0	83.29
	0.5	80	0.0	82.51
	0.6	73	3.2	83.80
	0.7	71	0.0	80.84
	0.8	61	0.0	78.03
	0.9	57	17.2	95.65

Total Costs for Different Stream Numbers

M	z	1115	1729	1147	1921	Mean
40	0.5	79.57	112.41		78.82	
45	0.6	83.80	82.93	82.51	82.18	82.85
	0.7	80.84	81.82	80.28	81.95	81.22
	0.8	78.03	81.42	80.44	83.16	80.76

Optimal Mean Total Cost 80.76

(occurring at M = 45, z = 0.8)

9.4.2 Table 9-2

Control 3B Applied to Model III

Results for Test Stream No.1115

M	z	$\overline{N}_R$	$\overline{C}_{RO}$	T.C.
40	-0.1	125	1.6	91.89
	0.0	114	0.4	87.77
	0.1	101	8.0	91.70
	0.2	91	10.4	91.22
	0.3	82	40.4	118.74
45	0.3	87	2.8	87.00
	0.4	84	0.0	83.59
	0.5	84	0.0	82.21
	0.6	72	3.2	83.50
	0.7	66	0.0	79.91
	0.8	63	11.2	90.46

Total Costs for Different Stream Numbers

M	z	1115	1729	1147	1921	Mean
45	0.5	82.21	86.19			
	0.6	83.50	83.27	82.14	82.20	82.78
	0.7	79.91	81.95	82.48	82.54	81.72

Optimal Mean Total Cost 81.72

(occurring at M = 45, z = 0.7)

9.5 Comments on Experiment Four

Both types of alternative policy used by the author, "not necessarily replenish" and "necessarily replenish" policies, did not perform as well in Experiment Four as they did in Experiment Three.

Comparative results are given below for mean optimal total costs over the 10-year simulation.

	Necessarily Replenish	Not Necessarily Replenish
Expt. 3	80.82	80.43
Expt. 4	81.72	80.76

As the policy of "necessarily replenish" failed to perform as well as "not necessarily replenish" in both experiments, it was thought of little value to consider this as a viable type of control any further.

The main reason why the results of Experiment 4 were poor (compared with Experiment 3) overall became apparent when studying the details of the simulation. Sub-stores were being replenished too early, that is, often at stock levels of six or seven, when the complex did not have a procurement order outstanding. Costs of maldistribution of types 1, 2A (see 6.8.1.2) are definitely not negligible in the case of replenishment to sub-stores before procurement order. (It will be remembered that the Dynamic Programming Method was established for T-values up to L, i.e. for use whilst a procurement is on order, when there would be no resulting maldistribution of type 1 or 2A possible, (with 'Share' being the rule employed for deciding reorder quantity).

9.5.1 Change in the formula for reorder level at sub-stores when an order is not outstanding

In order to circumvent the problem, the reorder level of sub-stores for the case when the procurement is not on order was made equal to that in existence when the procurement is ordered; this lends to continuity in the reorder level vs. time to procurement arrival function. For the data case of Model Three, 9.2.3 shows this stock level to be equal to 4. The complex was simulated with the above change in Experiment Five in 9.6.

9.5.2 A modification to the 'Share' ration rule

Another proposal, the merits of which are considered in Experiment Six, was to modify the "Share" rationing rule such that the resulting notional stock level at the ordering sub-store does not exceed the value that the allocation rule would yield.

It will be recalled that:-

Allocation = Mean Coverage Time Demand

+ z x Standard Deviation of Coverage Time Demand.

When this is translated into the terms of the case when a procurement is on order, due in time T, then this allocation is equal to $\lambda_i(T+\ell) + z\sqrt{\lambda_i(T+\ell)}$. Hence the new ration rule for the case of five sub-stores is:-

$$\text{Share MK.II} = \text{Min}\{I(F/N) \;,\; (\lambda_i(T+\ell)+z\sqrt{\lambda_i(T+\ell)})\}^*$$

* where I indicates "Integer value of".

The expected benefits of this proposal are the reduction in costs of maldistribution of type 2B (as a result of ensuring that "Share" is not excessive in its delivery quantities), but it is expected that further delivery numbers and also, possibly, an increase in maldistribution costs type 3 will result. The greatest benefits are of course likely to be noted where the maldistribution type 3 is absent and replenishment costs are low (Model Two fits this category).

The value of this proposal was tested independently of any improvements resulting from the use of the First Dynamic Programming method, in order to see whether the overall performance for the case of fixed reorder level at sub-stores could be improved. It was also tested in the case of the First Dynamic Programming method. Both tests are considered in Experiment Six in 9.7.

9.6 Summary of Experiment Five

10 Year Simulation of Model III with Control 4A Using IBM 7090 System with "C.S.L." Simulation Language

Full Description of Model Given in 8.5.

Control Decisions

The central store reorder quantity equals 60 and the reorder level for the complex is the parameter "M".

Control 4A: Author's suggestions with "Not Necessarily Replenish Policy"

Sub-store Reorder Quantity

Case 1: Procurement on Order: "Share Ration Rule" (see 5.6.4.2)
Case 2: Procurement not on Order: "Allocation Rule" (see 5.6.2.3)

Criterion of Reorder Level for Complex: "Free Stock"

Reorder Level for Sub-stores:

Case 1: Procurement on Order: "1st D.P. Model Decision Rule" (see 9.2.)
Case 2: Procurement not on Order:
Reorder Level made equal to that corresponding to T = L (when procurement order is initiated) from the 1st D.P. Model.

9.6.1 Table 9-3

Control 4A Applied to Model III

Total Costs for Different Stream Numbers

M	z	1115	1729	1147	1921	Mean
45	0.6	-	92.84	-	80.00	-
	0.7	82.10	81.58	78.96	78.36	80.25
	0.8	78.43	78.75	78.21	79.78	78.79
	0.9	81.75	82.41	-	-	-

Optimal Mean Total Cost 78.79

(occurring at M = 45, z = 0.8)

Breakdown of Costs at Optimum: Average Over 4 Streams

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Holding Cost = Average Stock	Mean
45	0.8	7.75	57	1.33	56.49	78.79

Optimal Mean Total Cost = 78.79

9.6.2 "t" test to determine whether the observed improvement of Control 4A over Control 2A is significant for Model III

Opt. Result For Control 2A	Opt. Result For Control 4A	d_i	Stream No.
81.50	78.43	3.07	1115
83.44	78.75	4.69	1729
78.04	78.21	-0.17	1147
78.76	79.78	-1.02	1921

The value of t = 1.21; dof = 3.
The difference is insignificant at the 5% level.

9.6.3 Comments on Experiment 5

As was expected, throughout the range of (M,z) combinations worth considering, the maldistribution of types 1, 2A, was much reduced by cutting the reorder level of sub-stores whilst a procurement order was not outstanding to 4. Maldistribution costs of type 2B were, in general, down; this was due to the fact that by postponing shipments to sub-stores until their stock levels dropped to 4, more stock was held in the central store when the procurement order was made; if "Share" did tend to be a little over-generous, this did not tend to lead to as much run-out at other sub-stores as was the case with a lesser amount of stock being held at the central store. The increase in maldistribution type 3, as a result of any such above-average demand in the lead time to sub-stores was insignificant for cases of replenishing sub-stores whilst the procurement order was not outstanding.

It was concluded that the establishment of the reorder level for sub-stores whilst a procurement order was not outstanding at that level of reorder obtained from the Dynamic Programming Method at time of complex reorder (T = L) was a valuable idea.

It was clear where the savings were occurring. The original Dynamic Programming approach of Experiment Four in 9.4.1 led to decisions to replenish sub-stores at the following stock levels:-

Sub-store notional stock	Replenish if estimated T
7	$T \geq 298$
6	$T \geq 206$
5	$T \geq 125$
4	$T \geq 62$

In general, shipments were ordered at stock levels of six and five and the average system stock existing at the complex reorder level of 45 (optimum) worked on Free Stock was in the region of 48 or 49. When the stock levels, 7, 6, 5 were excluded for consideration of sub-store reorder, the system stock at complex reorder level dropped to the region of 46, 47. This represented a saving of about 2 units of stock on average, resulting in a two unit cost saving over 10 years.

9.7 Summary of Experiment Six

10 Year Simulation of Model III with Controls 5A, 5B using IBM 7090 System with "C.S.L." Simulation Language

Full Description of Model Given in 8.5.

Control Decisions

Both controls have central store reorder quantity equal to 60, and the reorder level is equal to the parameter "M".

Control 5A: Author's suggestions with "Not Necessarily Replenish Policy"

Sub-store Reorder Quantity

Case 1: Procurement on Order: "Share MK II" Ration Rule (see 9.5.2)

Case 2: Procurement not on Order: "Allocation Rule" (see 5.6.2.3)

Criterion of Reorder Level for Complex: "Free Stock"

Reorder Level for Sub-stores

Case 1: Procurement on Order:)
Case 2: Procurement not on Order) the same parameter "ROL_{SS}"

Control 5B: Author's suggestions with "Not Necessarily Replenish Policy"

Sub-store Reorder Quantity

Case 1: Procurement on Order: "Share MK II" Ration Rule (see 9.5.2)

Case 2: Procurement not on Order: "Allocation Rule" (see 5.6.2.3)

Criterion of Reorder Level for Complex: "Free Stock"

Reorder Level for Sub-stores:

Case 1: Procurement on Order: "1st D.P. Model Decision Rule" (see 9.2)

Case 2: Procurement not on Order:
Reorder Level made equal to that corresponding to T = L (when procurement is initiated) from the 1st D.P. Model.

9.7.1 Table 9-4

Control 5A Applied to Model III

Total Costs for Different Stream Numbers

M	ROL_{SS}	z	1115	1729	1921	1147	Mean
40	3	0.1	111.11	105.11	82.57	83.62	95.60
		0.2	105.63	100.05	79.55	78.19	90.85
		0.3	99.43	102.06	82.35	79.52	90.84
		0.4	102.83	83.67	-	79.03	-
40	4	0.3	101.65	91.85	80.98	78.36	88.21
		0.4	100.47	90.06	79.09	79.23	87.21
		0.5	99.82	76.76	77.77	82.59	84.23
45	2	0.8	102.19	95.96	80.03	91.92	92.52
45	3	0.6	94.32	99.58	79.52	80.30	88.43
		0.7	87.48	80.44	78.51	80.21	81.66
		0.8	83.84	85.77	76.66	86.57	83.21
45	4	0.7	90.62	93.77	79.68	79.26	85.83
		0.8	85.28	80.32	79.23	81.86	81.67

Optimal Mean Total Cost 81.66

(at M = 45, z = 0.7, ROL_{SS} = 3)

9.7.2 Table 9-5

Control 5B Applied to Model III

Total Costs for Different Stream Numbers

M	z	1115	1729	1147	1921	Mean
40	0.4	98.62	89.08	89.37	79.02	89.02
	0.5	92.08	95.19	86.67	79.72	88.44
45	0.6	-	93.03	80.81	86.98	-
	0.7	-	82.72	81.48	86.16	-
	0.8	-	82.62	80.74	84.36	-

9.7.3 Comments on Experiment Six

At its best performance modification to the rationing rule of 'Share' to 'Share MkII' as proposed in 9.5 was unable to improve the results of the simulations. This applies to both tests of Experiment Six, that is, for fixed reorder level at the sub-stores and for the dynamic programming method for reorder level at sub-stores. The simulations showed that when the modified share rule "Share Mk II" was employed, the shipment quantities were, as expected, lower than when "Share" was employed. This indeed aids the positions of the other sub-stores with regard to run-out since more stock is held back at the central store, thus reducing maldistribution type 2B.

However, an interesting phenomenon is noticed. In the case of fixed reorder level at sub-stores, even though a sub-store is down to this level of this level of stock, its 'Share Mk II' ration quantity is likely to be less than this level, and hence replenishment is postponed until the notional stock level of the sub-store drops below the ration quantity designated by 'Share MkII', whence the difference (generally one unit) is shipped. In the case of the dynamic programming model for obtaining the sub-store reorder levels, the model is based on the assumption that delivery will always occur at the step-line in 9.2.3 or below it. This implies then that the rationing rule must always allocate to the ordering sub-store a quantity in excess of the notional stock level at which reorder takes place. One can see that, with the employment of 'Share MkII' as a rationing rule this is not the case. Because sensible (close to optimal) z-values are in the region of 0.7 (higher z-values necessarily exclude themselves because of the associated costs of Maldistribution Type 2A) a sub-store with a notional stock level of 2 at a T-value of, say, 10,* will be allocated the minimum of

* 9.2.3 shows that this store is at its reorder level.

$\{I(F/N)\ ;\ \left(\lambda_i(T+\ell)+z\sqrt{\lambda_i(T+\ell)}\right)\}$, which is most likely to be the second quantity, viz. $1.4 + z\sqrt{1.4}$ equalling (when rounded), 2. Hence shipment does not take place, contrary to the decision of the dynamic programming model.

It is clear then, that by employment of 'Share' thus modified, maldistribution of type 3 is more common than with the 'Share' rule and the heavy resultant shortages necessarily preclude this proposal for modifying the rationing rule from further consideration.

9.8 Summary of Results for Controls on Model I

	Optimal Cost	Optimal z or HBF	Optimal M
Control 1A (Table 6.1 at 6.8.1)	26.24	z = 0.0	35
" 1B (Table 6.2 at 6.8.2)	27.81	z = 0.0	35
" 1C (Table 6.3 at 6.8.3)	28.70	HBF = 0.4	40

Control 1A (author's "not necessarily replenish" policy) significantly differs from Control 1C (Cran's control) at the 0.5% level.

Control 1B significantly different from Control 1C at the 7% level.

9.9 Summary of Results for Controls on Model II

	Optimal Cost	Optimal z or HBF	Optimal M
Control 1A (Table 7.1 at 7.2.1)	27.00	HBF = 0.5	65
Control 1C (Table 7.2 at 7.2.2)	26.57	z = 0.0	65

Control 1A significantly different from Control 1C at the 5% level ("t" significance test).

9.10 Summary of Results for Controls on Model III

	Optimal Cost	Optimal ROL_{SS}	Optimal z or HBF	Optimal M
Control 2A (Table 8.1 at 8.5.1)	80.43	3	z = 0.7	45
Control 2B (Table 8.2 at 8.5.2)	80.82	3	z = 0.3	40
Control 2C (Table 8.3 at 8.5.3)	87.41	3	HBF = 0.3	50

Control 2A (author's suggestions) significantly different from Control 2C (Cran's control) at the 1% significance level.

	Optimal Cost	Optimal z	Optimal M
Control 3A (Table 9.1 at 9.4.1)	80.76	0.8	45
Control 3B (Table 9.2 at 9.4.2)	81.72	0.7	45
Control 4A (Table 9.3 at 9.6.1)	78.79	0.8	45

Control 4A (application of dynamic programming model for sub-store reorder level) improves over fixed reorder level Control 2A, but insignificantly at the 5% level. ("t" significance test.)

	Optimal Cost	Optimal z	Optimal M
Control 5A (Table 9.4 at 9.7.1)	81.66	0.7	45

9.11 Summary of Chapter and Introduction to Chapter Ten

It is stated initially that when the complex has an order outstanding, the problem of establishing sub-store reorder level may be resolved into the problem of balancing the cost of shortage at the sub-store by delivering too late against the cost of the delivery which may be saved if replenishment does not occur now. A Dynamic Programming method is developed to give the reorder level of sub-stores as a function of time until the procurement arrives at the central store.

The experiments using this Dynamic Programming method still faced the problem of what to do about the reorder level of sub-stores when the order for the complex was outstanding. The first of these experiments, Experiment 4, estimated this time T to be $(F-M)/\lambda_T + L$ and considered the author's control using the two alternative policies of "necessarily replenish" and "not necessarily replenish". The results of the latter experiment* were poorer than for Experiment 3, and this effect was traced to the fact that obtaining the reorder levels for sub-stores as a function of the estimated time until procurement arrival made use of an assumption which just does not hold true. It was proposed to make sub-store reorder level for T-values greater than L equal to that level corresponding to T = L in the Dynamic Programming method in the next experiment, Experiment Five.

In Experiment Six, another proposal (that of making the Ration Rule formula not exceed the formula which the 'Allocation' Rule would yield when the procurement is on order) was also tested for Model Three for the cases of fixed reorder level for sub-stores and for the Dynamic Programming method for sub-store reorder level. The result of Experiment Five was encouraging for the Dynamic Programming method but the ideas for modifying

*It was seen that "necessarily replenish" was again the poorer policy, so it was excluded from further consideration.

the rationing rule 'Share' were unsuccessful.

In Chapter Ten, an attempt at some analysis to improve the Rationing Rule is given.

CHAPTER TEN

REORDER QUANTITY FOR SUB-STORES WHEN A PROCUREMENT IS ON ORDER: ALTERNATIVE TO 'SHARE' RATION RULE

10.1 Introduction to Chapten Ten

In Chapten Ten, an attempt at some analysis to establish the reorder quantity for sub-stores when a procurement is on order is presented. Hitherto, the reader will recall, the rationing rule known as 'Share' has been employed. Attempt at modification to 'Share Mk II' was not successful in the case of Model Three.

Basically, the main idea is to balance the cost of a further possible replenishment to the sub-store in question as a result of issuing too little stock against the cost of run-out at other sub-stores in the complex as a result of Maldistribution Type 2B. Since the best results so far have been obtained by employment of the Dynamic Programming approach of Chapter Nine, attention will be restricted to control types using this method for sub-store reorder levels whilst the procurement is on order within the complex.

10.2 A Specific Case to Illustrate the Ideas

The situation considered is an actual case from the simulation of the complex with Model Three. Sub-store No.1 is at its reorder level and the problem is to determine the ration quantity. The stocks for all the stores are given in the table below:-

	Sub-store Stocks*	Central Store Stock
Notional	3,4,6,6,6	3
After Hypothetical Redistribution	5,5,6,6,6	0

F = 28

10.2.1 The costs associated with a particular ration quantity

Consider a ration quantity for sub-store No.1 of 3 units (i.e. no shipment). The resulting Free Stock is 28, the corresponding individual sub-store "fair share" defined by F/N being 5.6. The assumption is made that those sub-stores whose present notional stock levels exceed this "fair share" value will suffer zero cost of run-out until stock from the next procurement is available. The expected costs of run-out, in this time are assumed, therefore, to be confined to sub-stores Nos. 1,2. We cannot, however, say that the expected cost of run-out at sub-store No.1 is that resulting from a stock level of 3 with $(T+\ell)$ days to go before next possible delivery date. This is because there exists

*First number refers to store 1, second to store 2, etc...

a stock of 3 at the central store for possible delivery to sub-store 1. The same applies to sub-store 2 with its stock level of 4.

The assumption is now made that the position with respect to run-out of sub-stores Nos.1 ,2 and the central store is equivalent to that for a special, single store. This store will have a stock level equal to the combined level of the individual notional stocks of sub-stores Nos. 1 and 2 and the central store stock itself and a rate of demand equal to double (since we are dealing with two sub-stores) that of a single sub-store. Thus, in the case considered above, the expected costs of run-out of the complex stemming from an allocation of 3 to sub-store No.1 is that of a store of stock level 10 and the rate of demand equal to .08 (=2 x .04) in the case of Models One and Three. The value of this expected cost of run-out $c_s\psi(\overline{S} = 10; \ \overline{L} = T+\ell)$ for a rate of demand value $\overline{\lambda}$ of .08 may be computed from the function described in Appendix Five.

10.2.2 Implications of merging the stock levels of stores

The implication, of course, of this assumption is complete interchangeability of stock between the two sub-stores and the central store. This means that by considering the two sub-stores as a single store one of the sub-stores cannot experience shortage whilst the other holds stock. Unless deliveries are always in unit amounts, then it is possible for one store to run-out whilst the other has stock (even if not present at the store itself, then in transit to it). Thus, treating the two sub-stores as a single store assumes zero cost of run-out resulting from maldistribution within the two stores themselves. It also assumes that the two stores do not experience maldistribution of type 3 (maldistribution due to the finite time stock takes from the central store to a sub-store). It is expected that the performance of the complex utilizing this assumption will be influenced strongly by how close to the truth the assumption comes.

In order to compute the expected extra cost of replenishment associated with this allocation under consideration, recourse is made to the table at 9.2.3. This is the representation of the reorder level vs. T function. If the reasonable assumption is made that the sub-store being replenished at this stage will sustain, at the most, one extra delivery before the procurement arrives at the central store, then the expected extra costs of replenishment equal the cost of a single replenishment times the probability of again crossing the reorder level vs. T step function line. The method of calculating this latter probability $P(s,\ell)$ is given in Appendix 7B.

10.2.3 Obtaining the ration quantity "Share Mk III"

The first ration quantity considered is that equal to the present value of the sub-store notional stock level.* The sum of expected cost of run-out and expected cost of extra delivery is obtained as described above. The ration quantity is then raised by one and the corresponding sum of these two costs computed. As soon as this cost sum becomes greater than the previous-computed cost sum, the computations are complete, and the chosen ration quantity** is equal to the one with the minimum cost sum. This, of course, assumes convexity of the cost sum functions.

10.2.4 A table of results for the decision method in the case of the reorder point described in 10.2.1 is given below:-

Ration Quantity	P'(s,ℓ)*** Probability of Another Reorder	(1) Expected Cost of Extra Reorders	"Fair Share" Resulting from Allocation	No. of Stores Below Fair Share Stock Level	Hypothetical Distribution Stock Level	(2) Expected Cost of Run-Out	(1) + (2)
3	1.0	0.300	5.6	2	10	.006	.306
4	.931	0.279	5.6	2.	10	.006	.286
5	.819	0.246	5.6	2	10	.006	.252
6	.795	0.239	4.8	1	4	.246	.485

The optimal ration quantity is 5, resulting in a shipment quantity of 2.

10.3 Summary of Chapter and Introduction to Chapter Eleven

In this chapter, the ration quantity for sub-stores whilst a procurement is on order is reconsidered. Attention is restricted to the

---

* This means that initially the cost of not making a replenishment at all at this stage is computed. This takes the probability of another reorder before stock arrives in the central store as unity.

** To be known as Share Mk III.

*** See Appendix 7B.

author's sub-store reorder level policy which obtains this level from Dynamic Programming analysis.

Different ration quantities, from the value of the reorder level upwards are considered. It is assumed that the sub-store under consideration will experience, at most, one more reorder level in the time until the procurement arrives. It is shown how, given the reorder level vs. time function, the probability of another replenishment can be calculated, and thus the corresponding cost of extra deliveries as a result of the considered ration quantity is obtainable.

As the ration quantity alters, so does the resulting Free Stock level of the complex. Shortage costs are considered for only those stores whose notional stock will be less than the "fair share" level (computed, as usual, by F/N). The sum of the expected cost of shortages and extra replenishments is obtained for each ration quantity. Convexity of this sum with respect to ration quantity is considered, and the optimal choice of ration quantity is taken as "Share Mk III" where the cost sum turns upwards.

Chapter Eleven goes on to consider the problem of sub-store reorder level for the case when a stock order for the complex is <u>not</u> outstanding.

CHAPTER ELEVEN

REORDER LEVEL OF SUB-STORES IN THE CASE WHEN A PROCUREMENT IS NOT OUTSTANDING

11.1 Introduction to Chapter Eleven

At this stage, some sense of dissatisfaction was felt with regard to the rather arbitrary sub-store reorder level in the case of a procurement not on order being made equal to the sub-store reorder level at time $T = L^*$ obtained from the Dynamic Programming Model. Some relevant analysis was required.

Since an order for a procurement is not outstanding, the prime concern at this moment when considering the reorder level of a particular sub-store is not with the prevention of shortages at other sub-stores, but rather the prevention of run-out at this particular sub-store whilst attempting to ensure that, as a result of replenishing now, maldistribution is not engendered. Whilst stating that the immediate concern is not with other sub-stores' shortage possibility, the author wishes the reader to recall that this problem is considered important once the procurement has been ordered.

It can be seen that the tendency of the allocation rule to issue a stock quantity to a sub-store which results in maldistribution is related to both the present notional stock level of the sub-store, and the allocation quantity itself, the latter being a function of F, M, z. For a given couple of control parameters (M,z), then the decision whether to replenish a sub-store must depend on both notional stock and F. The problem can be seen to resolve into a Dynamic Programming Decision Model somewhat similar to that employed when the procurement order is outstanding.

11.2 Formulation of the Dynamic Programming Decision Model to Establish the Reorder Level of Sub-stores Whilst a Procurement Order is Not Outstanding

The cost of the decision to replenish a sub-store at a notional stock level s, when Free Stock is F, is

$$C_A(s,F) = c_R + \text{Total Expected Costs of Maldistribution}$$

$$= c_R + c_s\psi(s,\ell) + \text{Expected Costs of Maldistribution Types 1,2} \quad (1)$$

The cost of the decision to wait one day before reviewing the situation (one day being as in the previous Dynamic Programming model the minimum time unit of simulation) is:-

---

* i.e. T = 100, in the case of Model Three.

$$C_B(s,F) = \overline{\lambda}_i C(s-1,F-1) + 4^{*}\overline{\lambda}_i C(\bar{s},F-1) + (1-5\overline{\lambda}_i)C(s,F) \qquad (2)$$

where $C(s,F)$ represents the cost of the best decision for any $(\bar{s},F)$ couple.†

$C(s,F)$ is then the least of the two above costs:

$$C(s,F) = \text{Min}\{C_A(s,F);\ C_B(s,F)\}$$

The decision $D(s,F)$ is 1 ("Replenish now") or 0 ("Wait") according to whether $C(s,F)$ equals $C_A(s,F)$ or $C_B(s,F)$, respectively.

Note that $C_B(s,F)$ is defined on the assumption that by the next day, there is the chance that only one demand at most may occur on the complex (a reasonable assumption in the case of a total mean demand of 0.2 per day as for Model Three). The probability of this demand occurring; (a) on a sub-store other than the one under consideration is $4\overline{\lambda}_i$, resulting in a state $(\bar{s},F-1)$; (b) on the sub-store in question, is $\overline{\lambda}_i$, resulting in a state $(s-1,F-1)$. The probability of no demands at all is $(1-5\overline{\lambda}_i)$ resulting in the initial state (s,F); hence equation (2).

11.3 Boundary Conditions

The cost of the best decision for an (s,M) couple is clear for s-values less than or equal to the reorder level when T = L (say $s \leq s_o$ and $s_o = 4$ for Model Three) from the previous Dynamic Programming model. The decision function for these s-values is unity and the costs $C(s,M)$, $s \leq s_o$, are $c_R + c_S\psi(s,\ell)$ + Expected Costs of Maldistribution Type 2B.

Now the latter maldistribution costs (resulting from the over-distributing effect of 'Share') as in the case of the previous Dynamic Programming model, will be ignored. (It will be recalled that the main considerations of the former Dynamic Programming model were those of balancing the cost of run-out of the sub-store in question against the cost of delivery which might be saved by not delivering until the procurement arrives at the central store.) It was envisaged that this was likely to be a good assumption where the "right sort**" of buffer stock (reflected in the M-value of the complex) was employed. This assumption was supported both in the control of Model Three with the employment of the Dynamic Programming approach and with 'Share' in Experiment Five, where an extremely small run-out cost at the optimum was noted. (This was of Maldistribution Type 2A.) By far the major factor for close-to-optimal couples was maldistribution of type 1.

---

* In general, for N sub-stores, "4" is replaced by (N-1).

** Implying close-to-optimal.

† This is an approximation. It is possible for a level (s-1) to be reached with the Free Stock remaini at F.

Now we have D(s,M) for s-values greater than the reorder level corresponding to T=L from the previous Dynamic Programming model (i.e. $s > s_o$ in general) to be equal to zero. The cost of the best decision in these cases is less, then, than the cost of immediately replenishment, i.e. $C(s,M) < c_s\psi(s,\ell) + c_R; \quad s > s_o$.

But $C(s,M) \geq c_s\psi(s,\ell)$, since whatever time at which the sub-store is replenished, it must sustain an expected cost of run-out equal to $c_s\psi(s,\ell)$.

Thus $\quad c_s\psi(s,\ell) + c_R > C(s,M) \geq c_s\psi(s,\ell)$

The approximation:

$C(s,M) \approx c_s\psi(s,\ell) + c_R \times P(s,\ell)$ is suggested, where P(s,ℓ)* is the probability of a sub-store with a notional stock level s ordering in the central store lead time L.

11.4 Consideration of Maldistribution Costs

From the results of Experiment 5, the decision to assume expected costs of Maldistribution Type 2 as negligible was supported in the case of control with near-optimal (M,z) operating couples. The author is not too bothered that it is not a good assumption when (M,z) couples are not near optimal, since a commericial complex would never be operated with such a couple anyway.

Equation (1) of 11.2 thus is translated into:-

$C_A(s,F) \;=\; c_R + c_s\psi(s,\ell)$ + Expected Cost of Maldistribution Type 1,

and attention is now focussed on the method of computing the expected costs of Maldistribution Type 1 for an (s,F) couple.

11.4.1 Maldistribution Costs of Type 1

Consider the expected stock-time function for a complex when maldistribution of type 1 occurs. Figure 7 illustrates this.

If the complex reorders with a total stock level exceeding M by Δ, then it is clear from the Figure that the extra area under the total stock ∿ time graph is $Q.\ \Delta/\lambda_T$.

The corresponding cost of extra inventory holding is then $hP\Delta Q/\lambda_T$.

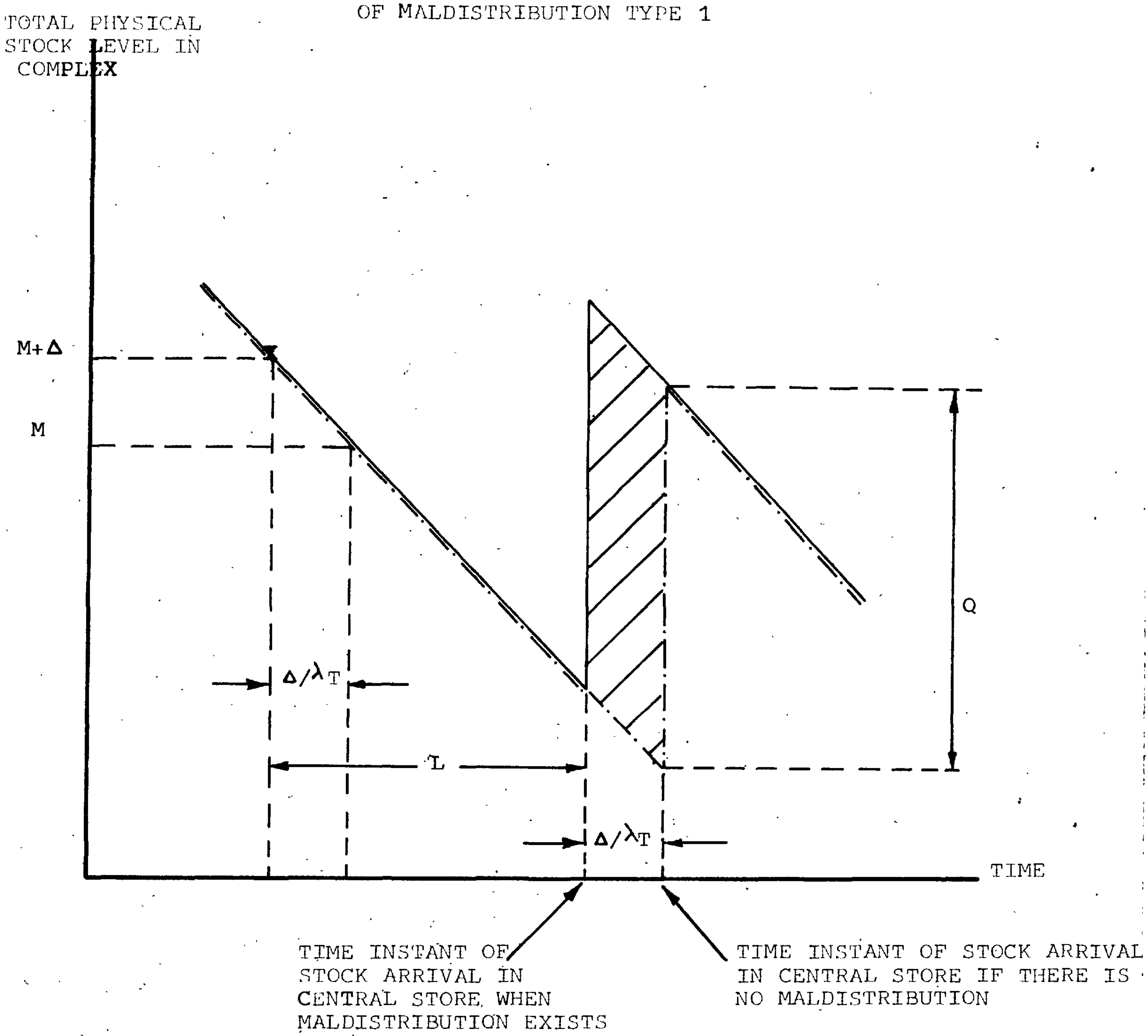
The problem now is to obtain the probabilities of occurrence of Δ. If we decide to replenish a sub-store, then the allocation 'A' is a function of F,M,z; for any given control then where the (M,z) couple is fixed, we have A = A(F).

---

* The method of computation is given in Appendix 7A.

FIG. 7

GRAPH SHOWING THE EFFECT IN TERMS OF EXTRA STOCKHOLDING OF MALDISTRIBUTION TYPE 1



LEGEND: EXPECTED GRAPH IF:

————————	MALDISTRIBUTION
-.-.-.-.-.-.-.-	NO MALDISTRIBUTION
X	REORDER POINT FOR COMPLEX

The ways in which this allocation can result in maldistribution of type 1 are now considered. At the time of reorder for the complex, we will have, assuming that maldistribution (if any) occurs at only the one sub-store under consideration in the coverage time* (this is a fairly reasonable occurrence; it is seen from the simulations that for any given time between procurement arrivals, the likelihood of maldistribution of type 1 occurring as a result of two 'over-distributions' is very small indeed), a stock level of (A-i), where i=amount of demand on the sub-store in question between the time of its allocation and time of reorder for the complex.

Provided that $(A-i) > .2M$, in the case of 5 sub-stores, then maldistribution has occurred.

Remembering assumption (a), we have:-

$$\Delta = (A-i) -.2M, \text{ and}$$

the cost of maldistribution (type 1) is given by C_{M_1} where;

$$C_{M_1} = \frac{hPQ}{\lambda_T} \sum_{i=0}^{A-.2M} (A-i-.2M) \cdot p(i)$$

$$= \frac{hPQ}{\lambda_T} \sum_{i=0}^{A-.2M-1} (A-i-.2M) \cdot p(i)$$

where p(i) denotes the probability of the occurrence of i.

Now immediately after the allocation A to the sub-store, the sum of the notional stocks at other sub-stores and the central store stock is (F-A). At the time of complex reorder, the combined notional stock of the other four sub-stores and the central store must equal 0.8M-1. (Note that it is not 0.8M, since with this value and with the remaining sub-store having a stock exceeding 0.2M - we are taking this store to be the one with maldistribution - the Free Stock of the complex is M+1, and the complex is not yet at reorder point.) Thus the four stores in total have sustained a demand of (F-A) - (.8M-1), i.e., (F-A-0.8M+1); hence p(i) is the probability of a demand i at one sub-store when the combined demand on the other (four) sub-stores is this quantity (F-A-0.8M+1).

Now the probability density function of the time in which a quantity K is demanded over four sub-stores is

$$\frac{(\lambda' t)^{K-1}}{(K-1)!} e^{-\lambda' t} \lambda' dt , \quad \text{where } \lambda' \equiv 4\lambda_i .$$

---

* Assumption (a).

The probability generating function (p.g.f) of the demand in this time on the sub-stores in question is:-

$$\frac{\lambda'^K}{\{(\lambda'+\lambda_i)-\lambda_i z\}^K}$$

$$= \{\lambda'/(\lambda'+\lambda_i)\}^K \;/\; \{1-\lambda_i z/(\lambda'+\lambda_i)\}^K$$

If $\lambda_i/(\lambda'+\lambda_i)$ is denoted by q, $\lambda'/(\lambda'+\lambda_i)$ by $p = 1-q$, then this p.g.f. is $p^K/(1-qz)^K$, which is the p.g.f. of the Negative Binomial Distribution.

Hence $p(i) = (K+i-1)!\, p^K q^i/(K-1)!\, i!$

and $$C_{M_1} = \frac{hPQ}{\lambda_T} \sum_{i=0}^{A-.2M-1} (A-i-.2M)(K+i-1)!\, p^K q^i/(K-1)!\, i! \qquad \text{(A)}$$

where $K = F-A-.8M+1$, and $p = .8$, $q = .2$ applicable for $K \geq 1$, and for $A \geq (.2M+1)$.

Obviously, if $A \leq 0.2M$, then C_{M_1} is zero. If A is sufficiently high (due to a high z-value) as to set off a procurement order immediately, i.e. if $K = 0$, then $C_{M_1} = (\frac{hPQ}{\lambda_T})(F-M)$.*

Further, note that in the case that $A = 0.2M+1$, the expression (A) for C_{M_1} yields:-

$$C_{M_1} = (hPQ/\lambda_T).p^K .$$

A note on the method of computation and a table of the solutions for the dynamic programming approach for reorder levels of sub-stores whilst the procurement is not on order are given in Appendices Eight and Nine, respectively.

11.5 Chapter Summary and Introduction to Chapter Twelve

This chapter has presented a Dynamic Programming Method for obtaining the reorder level at sub-stores as a function of the Free Stock level. The ideas of this chapter and of Chapter Ten are tested by simulation in the next chapter.

---

* for the case of Model Three, where $hP = .1$, $Q = 60$, $\lambda_T = 50$, $C_{M_1} = .12(F-M)$.

CHAPTER TWELVE

TESTING THE IDEAS OF CHAPTERS 10 AND 11

12.1 Introduction to Chapter Twelve

The merits of the ideas presented in Chapters 10 and 11, that is (a) the new Rationing Rule "Share Mk III" to establish the reorder quantities of sub-stores whilst a procurement is outstanding, and (b) the Dynamic Programming method to establish the reorder level of sub-stores when the procurement is not on order, are evaluated in this chapter, in Experiment Seven, for the case of non-zero sub-store lead time.

12.2. Summary of Experiment Seven

10 Year Simulation of Model III with Control 6A Using IBM 1130 System with "Simon" Simulation Language

Full Description of Model Given in 8.5.

Control Decisions

Central store reorder quantity equals 60, and the reorder level for the complex is the parameter "M".

Control 6A is defined thus:

Case 1: Procurement on Order: Ration Rule "Share Mk III" (see 10.2).
Case 2: Procurement not on Order: "Allocation Rule".

Criterion of Reorder Level for Complex: "Free Stock".

Reorder Level for Sub-stores:

Case 1: Procurement on Order: Function of "T" from 1st D.P. Model.
Case 2: Procurement not on Order: Function of "F" from 2nd D.P. Model.

12.2.1 Table 12-1

Control 6A Applied to Model III

Total Costs for Different Stream Numbers

M	z	1115	1729	1147	1921	Mean
45	0.6	82.03	83.95	79.25	78.62	80.96
	0.7	82.71	81.02	78.56	77.92	80.05
	0.8	78.75	77.31	78.25	78.03	78.08
	0.9	80.04	80.29	-	-	-

Breakdown of Costs at Optimum (Average over 4 Streams)

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Average Stock	T.C
45	0.8	7.75	57.67	1.25	55.76	78.08

Optimal Mean Total Cost = 78.08

12.2.2 Note that for the employment of this Second Dynamic Programming Model for Model III, the expected Free Stock level at time of procurement arrival (viz. 85 for M=45) is such that at this time the expected reorder level for sub-stores is about 8 (remember that this reorder level is a function of F, M, and z, and thus for a given (M,z) will be defined as a function of F) for near-optimal z. The expected notional sub-store stocks at this time are 5 or less each (the expected F-value immediately before procurement arrival will be 25); and thus in general sub-stores will be replenished immediately following the central store receiving the procurement. It is with this in mind that the simulation starting conditions were made thus:

Each sub-store has A(M+40)*

equal to $.2(40) + 5 + z\sqrt{.24(40)+5}$

i.e. $13 + z\sqrt{14.6}$ for Model III

and the central store has M+40 - Sum of sub-store stocks.

12.3 Comments on Experiment Seven

When compared to the previous best performance for Model III (obtained in 9.6.1 with Control 4A) a definite improvement is noted in respect of inventory holding costs. Run-out and replenishment cost alter insignificantly. Overall, the improvement did not appear to yield results commensurate with the sophistication of the ideas proposed for controlling the complex.

Detailed study of the simulation showed that the new rationing rule employed was still over-generous in its general operation; indeed, in the great majority of instances, its operation was identical to the "Share" formula itself.

12.4 Illustration of the Working of the Latest Ration Rule, "Share Mk III"

Reference is made to the stock configuration for the complex given in 10.2.

Central Store Stock	Notional Sub-store Stocks				
3	3.	4.	6.	6.	6.

The Free Stock is 28, and hence the 'Share' formula proposes a ration quantity equal to I(28/5), viz. 5, resulting in:-

* The expected F-value at time of procurement arrival is M+40.

Central Store Stock	Notional Sub-store Stocks					
1	5.	4.	6.	6.	6.	Configuration "A"

The choice for a ration quantity greater than the fair share value computed as 28/5 (=5.6), say a choice of 6, results in an actual configuration as below:-

Central Store Stock	Notional Sub-store Stocks				
0	6.	4.	6.	6.	6.

The resulting Free Stock is reduced to 24, and the store to receive the replenishment has a notional stock level exceeding its "fair share" level (which will be 4.8). Only sub-store No 2 is the one for which shortages are considered, and, naturally enough, the cost associated with one store with a stock of 4 to last (T+ℓ) days is considerably higher than the considered run-out costs for the next lower ration quantity. (These shortage costs will be those associated with configuration "A" resulting from a single store with a stock of 10 to last (T+ℓ) days, with an expected demand rate of double λ_i).

This effect of a sudden increase in shortage cost (which nearly always is so important that the combined expected cost of shortage and extra replenishment is greater than for the previous-considered ration quantity) is seen to occur for the first ration quantity greater than the "fair share" value of the stock configuration existing in the complex before the considered replenishment.

The ration quantity adopted is therefore that stock level equal to the value of the "Share" ration quantity itself. This result occurs for the majority of cases in the simulation and so the run-out costs due to Maldistribution Type 2B are not removed.

The improvement in performance over that from the control suggested in Experiment Five must therefore be attributed to that afforded by the Second Dynamic Programming Method (which establishes the reorder levels of sub-stores whilst a procurement is not on order).

12.5 Chapter Summary and Introduction to Chapter Thirteen

This chapter has invoked the ideas of Chapters Ten and Eleven in controlling the complex. The results indicate that the ideas for the new Rationing Rule are not useful, since the "overdistributing effect" of the "Share" Ration Rule is often repeated. However an improvement in overall cost indicates that the ideas for the Second Dynamic Programming

Method for obtaining the reorder level at sub-stores whilst a procurement is not on order may be useful.

In the next chapter, this Second Dynamic Programming Method is retained and a more sophisticated approach for rationing sub-stores whilst the complex is awaiting a procurement is considered.

CHAPTER THIRTEEN

AN ALTERNATIVE TO "SHARE MK III"; THE NEW RATIONING RULE FOR SUB-STORE DELIVERY QUANTITIES WHILST THE PROCUREMENT ORDER IS OUTSTANDING

13.1 Discussion of the Ration Rule "Share Mk III" and Introduction to the New Ration Rule "Share Mk IV"

The last-considered Ration Rule "Share Mk III" was based on the assumption that whatever the ration quantity considered, the sub-store under consideration would only receive one extra delivery, at most, before arrival of the procurement in the complex. This assumption is now relaxed.

Further, this Ration Rule assumes that the costs of Maldistribution Type 2, occurring as a result of any tendency in the Ration Rule to "over-distribute", were negligible for all future sub-store deliveries. The same assumption will apply to the new Ration Rule, since it is expected that if the latter is sufficiently useful, such maldistribution will be minimal.

Maldistribution Type 3 (where shortage is experienced in the sub-store lead time) was not considered for either Share Mk III or Mk IV since the quantity sent out has little direct effect (unless it is very low) on this maldistribution.

With "Share Mk III" the costs of maldistribution were assumed confined to those (K) sub-stores with notional stock levels less than the value of "Fair Share" (equal to F/5) resulting from the considered ration quantity. This cost was equated to the expected cost of shortage for these K stores taken as a single store with demand rate $K\lambda_i$, stock level equal to the sum of the sub-store* notional stocks plus central store stock, and with supply not available until the coverage time hence.

13.2 The New Ration Rule, "Share Mk IV"

13.2.1 General ideas

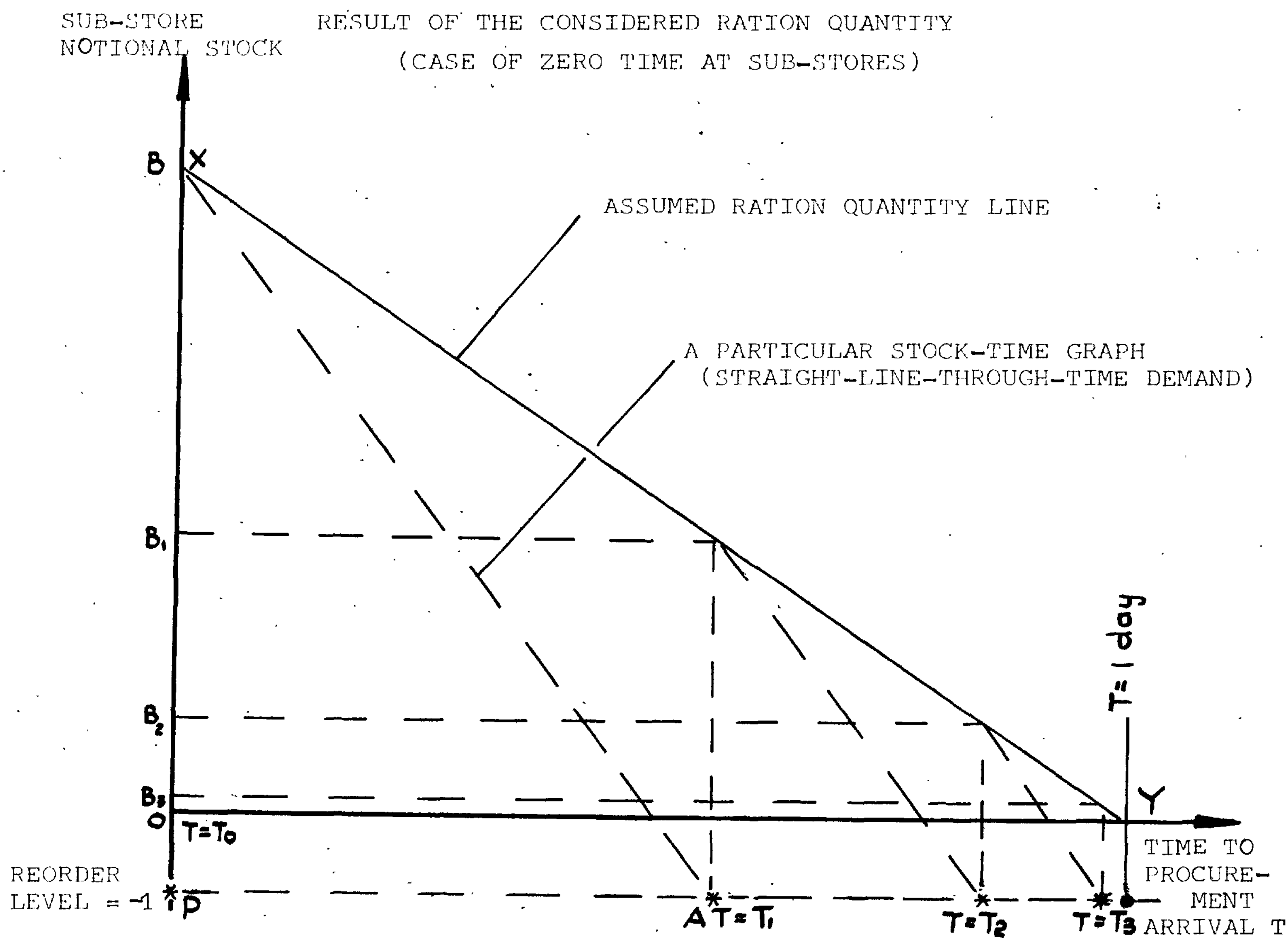
The reorder level has been met some time T_o before procurement arrival ($L \geq T_o > 0$) as shown in Figure 8. Suppose B is chosen as the ration quantity. We would like to know as a result of this ration quantity what the ration quantity is likely to be for other values of $T < T_o$. A reasonable approximation is assumed to be the straight line joining the point X (corresponding to value B at time T_o) to the point Y (corresponding to one unit more than reorder level at T=1). To view the consequences of the decision to allow a quantity of B to the ordering sub-store different demands in future time (T_o-1) will be considered. Attention is restricted to linear through time demand rates. These demands will have the random variable d from the Poisson distribution with mean $\lambda_i(T_o-1)$.

---

* i.e. K sub-stores.

FIG. 8

ILLUSTRATION OF EXPECTED EXTRA REPLENISHMENTS POSSIBLE AS A RESULT OF THE CONSIDERED RATION QUANTITY (CASE OF ZERO TIME AT SUB-STORES)



LEGEND * EXPECTED REORDER POINTS

13.2.2 Consideration of Further Reorder Points

The line XA of Fig.8 illustrates a particular demand d of XP/PA units/day. As a result of the (d,B) combinations further replenishments to this sub-store are expected at times T_1, T_2 and T_3. If the number of such further replenishments is, generally, denoted by K(d,B) and the quantity of stock delivered in such extra deliveries are denoted by b_i, (i = 1,2,... K(d,B)) we have a total of

$$S_T - B - \sum_{i=1}^{K(d,B)} b_i$$ items* to last the other (N-1) stores for a time period equal to $T_o + \ell$ (at which time stock can first get to them from the procurement now on order).

The cost of shortages associated with this stock position is estimated at:-

$$\overline{C}_s(d,B) = c_s\psi(\overline{s}, \overline{L}, \overline{\lambda}) \quad **$$

where
$$\overline{s} = S_T - B - \sum_{i=1}^{K(d,B)} b_i$$
$$\overline{L} = T_o + \ell$$
$$\overline{\lambda} = (N-1)\lambda_i = 4\lambda_i \text{ for 5 sub-stores.}$$

13.2.3 Cost considerations

The combined cost of extra deliveries and shortage cost of the complex until the coverage time is reached is thus estimated at:-

$$C(B) = \sum_d p(d) \left(\overline{C}_s(d,B) + K(d,B) \times c_R\right)$$

where p(d) denotes the probability of the random variable d.

Putting $\lambda_i T_o$ equal to $\overline{\mu}$ we have

$$p(d) = e^{-\overline{\mu}}\, \overline{\mu}^d / d!$$

It is assumed that C(B) is convex.*** Optimal $\hat{B}$ is thus given by $\hat{B}$ where $\hat{B} = \left(\text{Min } B: \; C(B+1)-C(B) > 0\right)$ (1)

---

* S_T is the total virtual stock in the complex at time $T=T_o$.

** The function $\psi(\overline{s}, \overline{L}, \overline{\lambda})$ is given in Appendix 5.

*** This was seen to be true from the results of Experiment 8.

13.2.4 Assumptions of this analysis

This analysis assumes the sub-store under replenishment consideration itself does not incur shortage. In the event of high d however, this assumption does not hold true and we wish to consider the effect of B on the combined cost of shortage and extra deliveries when shortage is experienced at the ordering sub-store, say for $d > d_o$.

Equation (1) is transformed thus:- $\hat{B}$ = Min B such that

$$\sum_{d=0}^{d_o} p(d) \left(\overline{C}_s(d,B+1) - \overline{C}_s(d,B) + \{K(d,B+1)-K(d,B)\} \times c_R\right)$$

$$+ \sum_{d=d_o+1}^{\infty} p(d) \left(\text{Combined Shortage Cost and Delivery Cost for ration quantity B+1 - Combined Shortage Cost and Delivery Cost if ration quantity is B}\right) > 0$$

If the term within this latter bracket is referred to as γ, then the latter summation is $\sum_{d=d_o+1}^{\infty} p(d)\gamma$ and for d values greater than d_o, it is clear that p(d) itself is small, and the γ term is also small (since if shortages do occur at the considered sub-store it is because the complex itself is short of stock and so the combined shortage over all stores is likely to change minimally whether $\hat{B}$ or $\hat{B}+1$ be employed as the ration quantity.). Clearly the difference in replenishment cost is minimal.

The product $p(d)\gamma$ is therefore minimal and the second summation is thus neglected for d_o for which the cumulative probability

$$\sum_{i=d_o}^{\infty} p(i) < .05.$$

13.2.5 Consideration of detail in Fig. 8

This detail is shown in Fig.9 and allows a formula for the number of deliveries to be established for the zero sub-store lead time case.

1 or more extra deliveries corresponds to a demand $d \geq B+1$ (1)

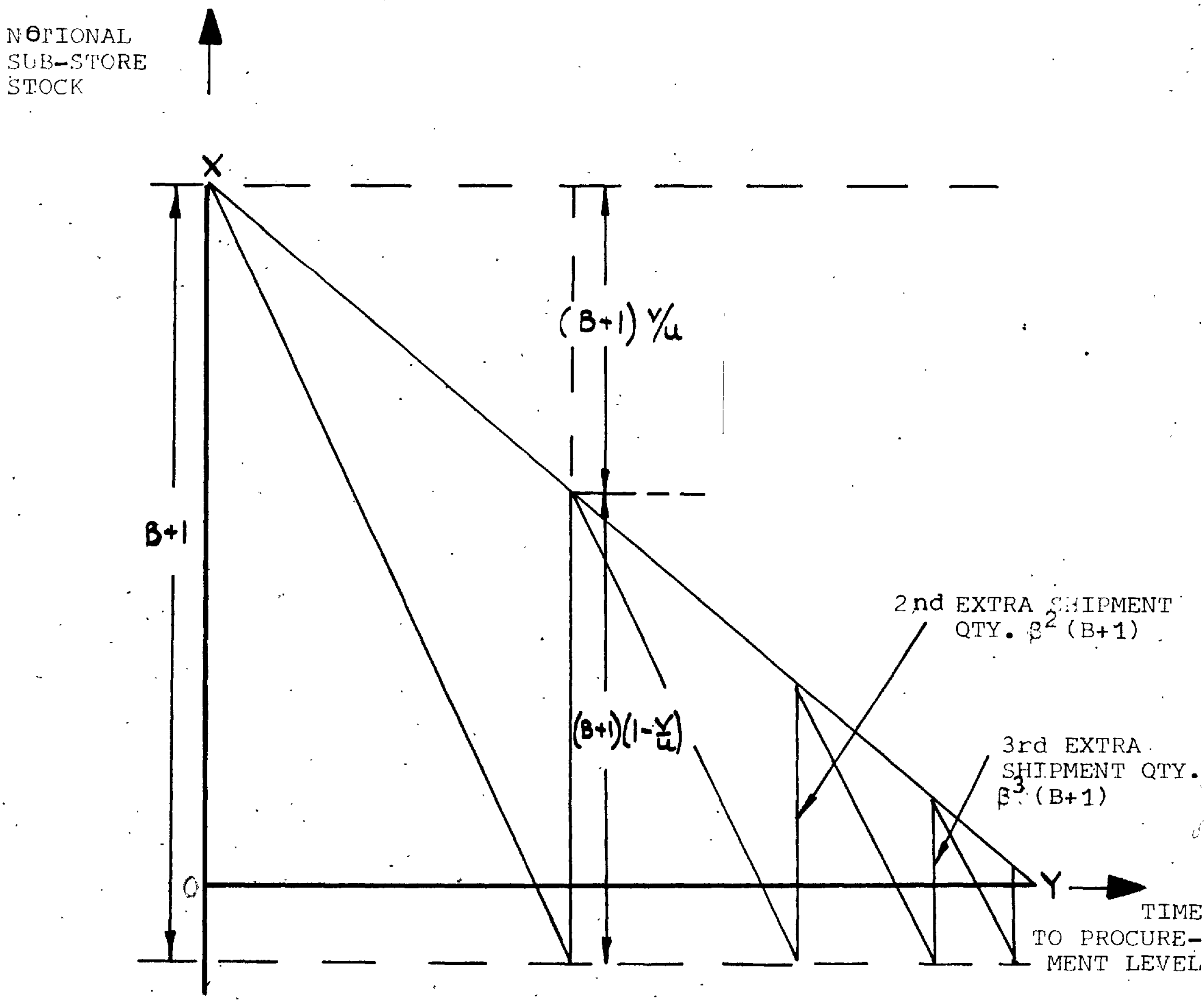
2 or more extra deliveries corresponds to $d \geq (B+1)+(1-B/d)(B+1)$ (2)

and so for K or more deliveries, we have:-

$$d \geq (B+1)\left(1+(1-B/d)+(1-B/d)^2 + \dots (1-B/d)^{K-1}\right)$$

FIG. 9

DETAIL FROM FIGURE 8



N.B. SLOPE OF DEMAND LINE = u
SLOPE OF XY = v
$\beta = 1 - v/u$

i.e. $d \geq (B+1)(1+\beta+\beta^2+ \dots +\beta^{K-1})$ where $\beta = B/d$

i.e. $d \geq (B+1) \{\frac{1-\beta^{K-1}}{1-\beta}\}$

Utilization of (1) and (2) leads to:-

$(1-\beta) + 1 > d/(B+1) \geq 1;$ $K(d,B) = 1$

$(1-\beta)^2 + (1-\beta) + 1 > d/(B+1) \geq 1 + (1-\beta);$ $K(d,B) = 2.$

In general:-

$$1 + \sum_{j=1}^{K(d,B)} (1-\beta)^j > d/(B+1) \geq 1 + \sum_{j=1}^{K(d,B)-1} (1-\beta)^j$$

corresponds to K(d,B) deliveries.

13.3 Summary of Experiment Eight

4 Year Simulation of Models I, II with Control 7A Using IBM 1130 System with "Simon" Simulation Language

Full Description of Model I Given in 6.8.
Model II Given in 7.2.

Control Decisions

Central store reorder quantity equals 60 for Model I and 85 for Model II. Complex reorder level is a parameter "M" in both cases.

Control 7A

Sub-Store Reorder Quantity

Case 1: Procurement on Order: Ration Rule "Share Mk IV" (see 13.2.)

Case 2: Procurement not on Order: "Allocation Rule"

Criterion of Reorder Level for Complex: "Free Stock"

Reorder Level for Sub-stores

Case 1: Procurement on Order)
Case 2: Procurement not on Order) Reorder Level = -1 .

13.3.1 Table 13-1

Control 7A Applied to Model I

Results for Test Stream No. 1729

M	z	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{RO}$	Area	T.C.
30	-0.3	3	33	2.0	39402	29.1608
	0.0	3	29	4.8	39589	30.8536
	+0.3	3	25	52.0	39767	76.9067
35	0.0	3	28	0.0	44397	27.6587
	0.3	3	24	0.0	44397	26.4586
	0.6	3	22	16.0	44857	42.0427

Total Costs for Different Stream Nos.

M	z	1729	1921	1147	Mean
35	0.0	27.6587	27.0452	27.0196	29.2412
	0.3	26.4586	26.1452	25.5196	26.0411

Breakdown of Costs at Optimum

M	z	Stream	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{RO}$	Area	T.C.
35	0.3	1729	3	24	0.0	44397	26.4586
		1921	3	22	0.0	45113	26.1452
		1147	3	22	0.0	43549	25.5196
		Mean	3	22.7	0.0	44353	26.0411

Mean Total Cost 26.04

13.3.2 Table 13-2

Control 7A Applied to Model II

Results for Test Stream No. 1729

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
65	-0.3	6	59	0.0	54767	26.6767
	+0.0	6	53	0.0	54767	26.4967
	+0.3	6	48	1.8	54785	28.1539

Total Costs for Different Stream Numbers

M	z	1729	1921	1147	Mean
65	-0.3	26.6767	26.9631	26.1275	28.5891
	+0.0	26.4967	26.7831	26.1079	26.3469
	∓0.3	28.1539	26.2831	26.2995	26.9122

Breakdown of Costs at Optimum

M	z	Stream	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
65	0.0	1729	6	53	0.0	54767	26.4967
		1921	6	53	0.0	55483	26.7831
		1147	6	56	0.1	53320	26.1079
		Mean	6	54	0.03	54523	26.3469

Mean Total Cost 26.35

13.4 Significance Testing the Improvement Obtained by Use of the "Share Mk. IV" Ration Rule Instead of "Share Mk I" in the Application to Models I, II

13.4.1 Application of "t" significance test for Control on Model I

Share Mk I Result (Control 1A)	Share Mk IV Result (Control 7A)	d_i	Stream No.
26.4586	26.4586	0	1729
26.4452	26.1452	.3	1921
25.8196	25.5196	.3	1147

t is given by $\bar{d}/\sqrt{\{\Sigma(d_i-\bar{d})^2/\bar{N}(\bar{N}-1)\}}$, dof = 2

$\therefore$ t = 2.0.

The improvement in cost is judged significant at the 10% level.

13.4.2 Application of "t" significance for Control on Model II

Share Mk I Result (Control 1A)	Share Mk IV Result (Control 7A)	d_i	Stream No.
26.6480	26.4967	.1513	1729
27.0649	26.7831	.2818	1921
25.9879	26.1079	-.1200	1147

t = 0.889, dof = 2.

The improvement is insignificant at 5% or 10% levels.

13.5 Comments on Experiment Eight

Application of "Share Mk IV" to Model I (for Results, see Table 13-1 at 13.3.1) represents a 2% improvement in total cost result over the previous best control for Model I (viz. that in 6.8.1). However most of the costs comprising the total cost are irreducible anyway. (A substantial inventory holding cost and fair-sized replenishment and procurement cost cannot be eliminated whatever the control type.) As was expected with "Share Mk IV", delivery numbers are increased, but with the benefit of reducing backup substantially, leading to lower overall costs of control, with a higher z-value.

The same type of result is noted in the application of "Share Mk IV" to Model II. Replenishment costs are increased, but backup is reduced, and an overall cost saving is noted.

The "t" significance test shows insignificant improvement in savings due to the use of "Share Mk IV" rather than "Share" in the case of Model II yet a significant improvement at the 10% level for Model I.

13.6 Chapter Summary and Introduction to Chapter Fourteen

This chapter has established a new Ration Rule "Share Mk IV" considered to be superior to the last ration rule "Share Mark III".

For different considered ration quantities, various future demands (the probabilities of occurrence of which are known) are considered and a model proposed which estimates the expected cost of extra replenishments required for the sub-store in question. (Clearly, this will be non-zero unless the quantity distributed is high.) The cost of shortage in the complex is obtained by taking several factors into account to obtain a stock value (say S_g) and computing the shortage cost as that for a single store with stock S_g, demand rate $(N-1)\lambda_i$, at which stock will arrive in the coverage time hence.

The flow diagram used for the Ration Rule is phrased only for the context of zero lead time at sub-stores, and the ideas are tested on Models I, II. An improvement is noted over control with "Share(Mk. I)", insignificant for Model II, but passing the "t" significance test at the 10% level for Model I.

CHAPTER FOURTEEN

EXTENSION OF THE NEW RATIONING RULE "SHARE MARK IV" TO COVER THE CASE OF NON-ZERO SUB-STORE LEAD TIME

14.1 Introduction to Chapter Fourteen

Following the success of the rationing rule of "Share Mk IV" (developed in Chapter Thirteen for deciding on the amount to issue a sub-store which has reached its reorder level whilst the complex is awaiting the procurement arrival) in the case of zero sub-store lead time, this chapter is devoted to the extension of the rule to govern the case where the sub-store lead time is non-zero. Model III applies. Basically the ideas are very much the same, with the additional complication of having a reorder level which depends on how long to go before the procurement arrives.

14.2.1 The ration rule "Share Mk IV^B"

This is identical to the ration rule of the previous chapter except that it is applicable to non-zero sub-store lead time cases.

The situation is as in Figure 10 and as before the ration quantity line is assumed to run from the point (B,T_o) to the point (f+1, 1) where f corresponds to the reorder level at the time T = 1.* The analysis of Chapter Thirteen other than that of 13.2.5 is applicable.

A flow diagram for the computation of the values Σb_i, K(d,B) is given in Fig.17 (at Page 225). The expression "ε(n)" as a function of reorder level "n" gives the T-value at which the reorder level step drops from level "n" to level (n-1) from the 1st Dynamic Programming Model.

14.2.2 Computation of the ration quantity function

The Expected Ration Quantity Line has the equation:-

$$R(T) = (f+1) + \frac{T-1}{T_o - 1}\{B - (f+1)\}$$

14.2.3 Computation of the value $\overline{C}_s$, the expected cost of shortage for the other sub-stores in the complex when a sub-store ration quantity is being decided upon

The procedure uses the expression established in Appendix 5,

$$\psi(\overline{s},\overline{L},\overline{\lambda}) \text{ where } \overline{s} = S_T - \sum_{i=1}^{K(d,B)} b_i - B, \quad \overline{L} = T_o + \ell, \quad \overline{\lambda} = 4\lambda_i .$$

It is better to replace $\overline{\lambda}\,\overline{L}$ by $\overline{\mu}$ in the expression whence:-

$$\overline{C}_S(d,B) = c_s \{\frac{\overline{\mu}\,\overline{L}}{2} \left(1 - \sum_{j=0}^{\overline{s}-1} \frac{\overline{\mu}^j}{j!} e^{-\overline{\mu}}\right)$$

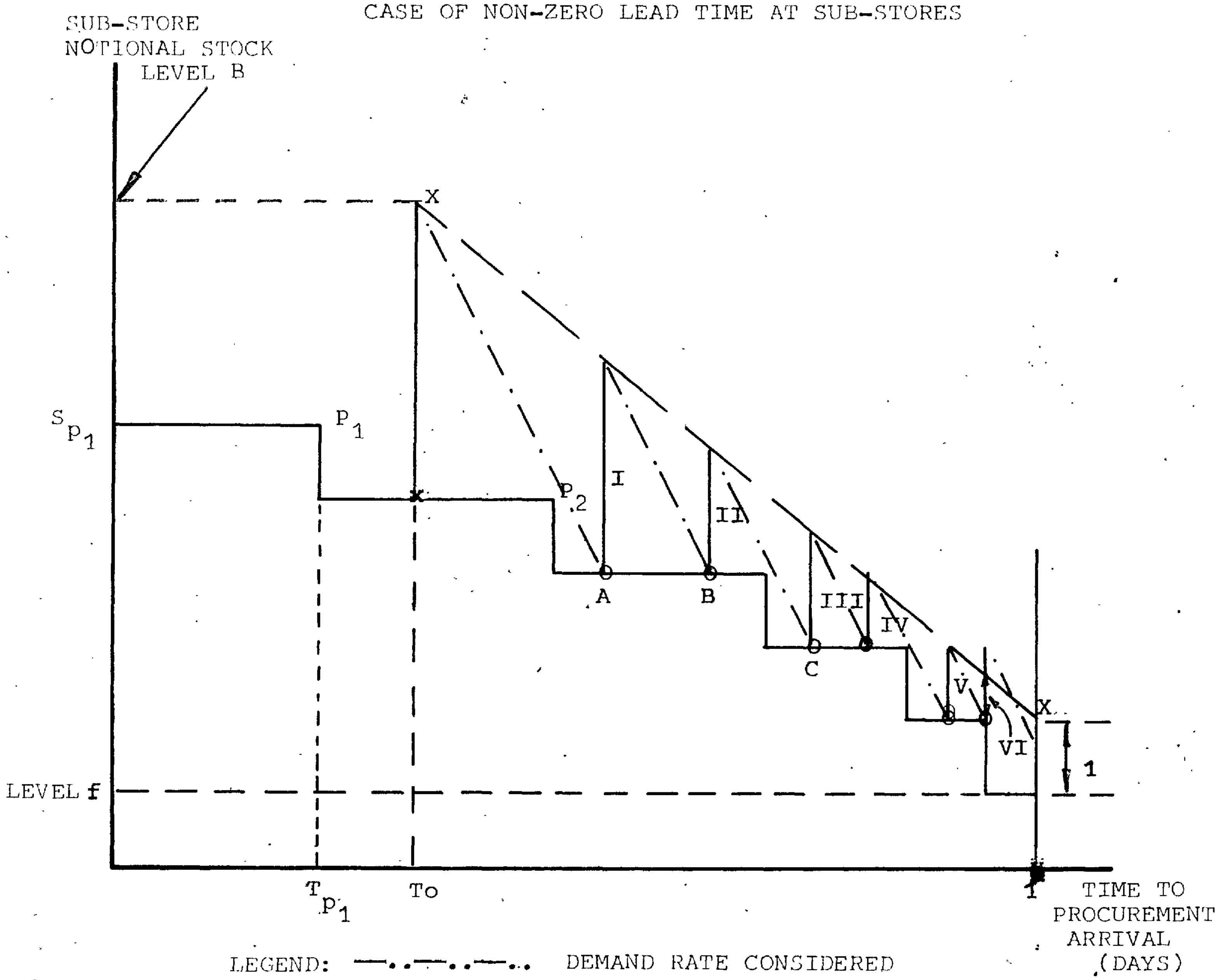
$$- \overline{s}\,\overline{L} \left(1 - \sum_{j=0}^{\overline{s}} \frac{\overline{\mu}^j}{j!} e^{-\overline{\mu}}\right) + \frac{\overline{s}^2 + \overline{s}}{2\overline{\lambda}} \left(1 - \sum_{j=0}^{\overline{s}+1} \frac{\overline{\mu}^j}{j!} e^{-\overline{\mu}}\right)\}$$

---

* If f is not defined, f is taken as 0.

FIG. 10

ILLUSTRATION OF EXPECTED EXTRA REPLENISHMENTS AS A RESULT OF THE CONSIDERED RATION QUANTITY IN RATION RULE "SHARE MK.IVB": CASE OF NON-ZERO LEAD TIME AT SUB-STORES



LEGEND: —..—..—.. DEMAND RATE CONSIDERED

III EXTRA REPLENISHMENTS

— — — RATION QUANTITY LINE

X REORDER POINT NOW

O EXPECTED FUTURE REORDER POINTS IF PRESENT RATION QUANTITY IS B

⊠ STOCK ARRIVAL IN CENTRAL STORE

14.3 Experiment Nine

This experiment is a simulation of Model III of the complex with similar contol to that used in Experiment Seven (see 12.2) where the best results so far have been achieved. Here, however, the new Ration Rule, "Share Mk IV^B", as described in this chapter, replaces the "Share Mk III" Ration Rule.

14.3.1 Summary of Experiment Nine

10 Year Simulation of Model III with Control 8A Using IBM 1130 System with "Simon" Simulation Language

Full Description of Model given in 8.5.

Control Decisions

Central Store Reorder Quantity equals 60, and the reorder level for the complex is the parameter "M".

Control 8A is defined thus:-

Sub-store Reorder Quantity:-

Case 1: Procurement on Order: Ration Rule "Share Mk IV^B" (see 14.2)
Case 2: Procurement not on Order: "Allocation Rule"

Criterion of Reorder Level for Complex: "Free Stock"

Reorder Level for Sub-stores:

Case 1: Procurement on Order: Function of "T" from 1st D.P. Model
Case 2: Procurement not on Order: Function of "F" from 2nd D.P. Model

14.3.2 Table 14-1

Control 8A Applied to Model III

Total Costs for Different Stream Numbers

M	z	1115	1729	1147	1921	Mean
45	0.9	78.35	77.91	76.45	77.03	77.44
	1.0	77.35	77.11	76.16	76.75	76.84
	1.1	77.48	76.67	76.69	76.47	76.83
	1.2	78.20	77.14	-	-	-

Breakdown of Costs at Optimum (Average over 4 Streams)

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Holding Costs Average Stock	T.C.
45	1.1	7.75	55.67	0.4	55.86	76.83

Mean Total Cost = 76.83

14.4 Significance Testing the Improvement Afforded by the Use of the New Ration Quantity Rule

The result must be compared against the result obtained by usage of the identical control but with the previous-best ration rule. This was the control simulated in Experiment Seven at 12.2.

Result: Control 6A	Result: Control 8A	d_i	Stream No.
78.85	77.48	1.27	1115
77.31	76.67	0.64	1729
78.25	76.69	1.56	1147
78.03	76.47	1.56	1921
78.08	76.83	$\overline{d}$ = 1.25	Mean

The value of t is given by $\overline{d}\sqrt{\{\Sigma(\overline{d}-d_i)^2/N(N-1)\}}$

$$t = 6.2, \quad dof = 3.$$

The improvement afforded by "Share Mk IV^B" Ration Rule is judged significant at the 0.5% level.

14.5 Comments on Experiment Nine

The previous best operation of the complex was achieved by use of "Share Mk III" coupled with the two Dynamic Programming Models for sub-store reorder level (for the two respective cases; procurement on and not on, order) at 12.2.1 in Experiment Seven. This represented nearly a 1% improvement in total cost over the result of Experiment Five (where the Second Dynamic Programming Model was not employed, and where the Ration Rule was "Share" (see 5.6.4.2). A further reduction in total cost from 78.08 (Experiment 7) to 76.83 is achieved by employment of the rationing rule "Share Mk IV^B" along with the Second Dynamic Programming Model (sub-store reorder level as a function of Free Stock) in 14.3.1 of Experiment Nine.

In general, as with Experiment Eight for models of the complex with zero lead time to sub-stores, replenishment totals are increased with control using Share Mk IV^B but savings in backup are achieved. The control responds to this by allowing a higher z (which tends to a reduction in delivery totals) whilst performing very well in reducing backup. The optimum turns up at z = 1.1, where the average backup cost in a ten-year simulation is equivalent to only one day of shortage. At higher z-values, the increase in backup cost is greater than the decrease in replenishment costs.

In the next section a suggestion to improve the present Ration Rule (Share Mk IV for zero sub-store lead time, and Share Mk IV^B for non-zero lead times) is given.

14.6 Ideas for Modifying the Ration Rules "Share Mk IV, IV^B"

14.6.1 The disadvantage of computing the shortage costs of the remaining (N-1) sub-stores as the expression in 13.2.2 and 14.2.3 is that due to combining or "lumping" the stock levels of the individual stores together and considering the shortage of a hypothetical store with the lumped stock and (N-1) times the individual demand rate.

Consider the following stock configurations for the four* remaining sub-stores and central store

T = 100	3,	3,	3,	3,	(20)**	No.1
ℓ = 25						
λ = .04	6,	6,	6,	6,	(8)	No.2
Mean demand is 5 in coverage time for each sub-store	8,	8,	8,	8,	(0)	No.3

For Case 3, "lumping" the stocks together and approximating the shortage as that for a store with demand rate .16 with a mean demand in the coverage time of 20 and with a stock level of 32 will give a lower cost of shortage than summing the expected costs of shortage for four stores each with stock levels 8 and mean demand 5 in the coverage time. In this case the correct estimate is naturally the latter.

For Case 2, the stock of 8 at the central store is available to replenish the needy sub-stores and so lumping all the four sub-stores' stocks together (and in effect stating that complete interdistribution without cost is possible) is not such a poor assumption.

For Case 1, it is probably a closer approximation to reality to consider the stores "lumped together into one store" rather than take the sum of the expected costs of shortage for four stores each with stock 8.

---

* Equal to (N-1) for complexes in this thesis.

** The bracketed value is central store stock less the value $\sum_{i=1}^{k(d,B)}$ less the shipment quantity to the considered sub-store. This stock is known as the "left stock".

14.6.2 Where to "Draw the Line" between the "Lumping" and "Non-Lumping" of Stocks

The problem of "where to draw the line" between lumping the stores together and considering the run-out costs of the separate stores now arises. By "lumping the sub-store notional stocks together" this makes a configuration like No.3 appear falsely safe. By considering the stores separately a configuration like No.1 appears falsely unsafe.

A suggested "line" is where the value of "left stock" exceeds the sum of the four sub-store notional stocks. Thus only configuration No.1 would be considered as a single store.

14.6.3 How to compute shortage costs when stores are considered individually

The problem now arises as to how to estimate the costs of run-out for a configuration such as:-

8, 7, 9, 5, (5)*

The stock in brackets will be assumed distributed amongst the sub-stores "in such way as to equalise stocks".

Thus we have:-

8, 9, 9, 8, from which the individual stores' run-out costs are computable.

The Ration Rule "Share Mk IV^B" thus modified is termed "Share Mk V".

14.7 Summary of Chapter and Introduction to Chapter Fifteen

This chapter has considered in Experiment Nine the application of the latest ration rule for the decision of how much to ship an ordering sub-store while the complex has stock on order. The analysis follows along the same lines as Chapter Thirteen, but the computations for expected extra replenishment numbers and expected future shipment quantities are made somewhat more lengthy owing to the step-function nature of the reorder level of sub-stores in the future.(the latter is a function of time to go until procurement arrival).

Application of control using the resulting Ration Rule "Share Mk IV^B" leads to a significant improvement ("t" test yields this result for a 0.5% significance level) over the previous best control (with Share Mk III in Experiment 7). An improvement (Share Mk V) which suggests a criterion for either "lumping" sub-store stocks together or considering sub-stores in-

* Again, the bracketed stock level refers to "left stock".

dividually for the purpose of estimating future shortage costs until the coverage time is finally suggested.

CHAPTER FIFTEEN

INTRODUCTION OF NEW MODELS, TO TEST THE WORKINGS OF THE AUTHOR'S CONTROL OVER A MORE GENERAL FIELD

15.1 Introduction to the New Models IV, V

Further models, with variations in data were introduced in order that the performance of these with control as suggested by the author may be determined.

In Chapter Fourteen we have been dealing with Model III in which the data is as below:-

Replenishment Cost: 0.3

Demand Rate: 10 per year per sub-store, 5 sub-stores,

i.e. λ_T = 50 units/yr

Procurement Cost: 0.5

Holding Cost: 10% item value per year

Shortage Cost: 0.4 per item-day of shortage

Procurement Lead Time: 100 days (.4 year)

Sub-store Lead Time: 25 days (.1 year)

The data for Model IV is identical except that the cost of replenishment is reduced to 0.03.

It is felt that this means that more replenishments per procurement are likely and this is reflected in the calculation of Q*, the procurement quantity, which is 45 for this model.**

The data for Model V is identical to that for Model III with the exception of the lead time being increased to 50 days.*** It is expected that shortage costs will necessarily increase and this is reflected in the computation of Q*, taken as 75.

15.2 Summary of Experiment Ten

4 Year Simulation of Model IV with Control 9 Using IBM 1130 System with "Simon" Simulation Language

Full Description of Model given in 15.1

Control 9 identical to Control 8A (see last Experiment No.9 at 14.3.1) with following exceptions:

---

* See Appendix 10

** The value (hPQ/λ_T) figuring in the calculation of C_{M_1} for the Second D.P. Model now has values .09, .15 respectively for Models IV, V.

*** In which mean sub-store demand is 2.

Sub-store Reorder Quantity:-

Case 1: Procurement on Order: Ration Rule "Share Mk V" (see 14.6) in Exhibit 'B' followed by "Share Mk VA" in Exhibit 'C'.

Complex Reorder Quantity:

Q = 45.

15.3 Analysis of Simulation Details

Analysis of the operation of the Ration Rule "Share Mk V" shows that the rule is too liberal when the sum of the sub-store stocks exceeds the central warehouse stock. This is clearly seen in Exhibit 'B'. This simulation shows that at procurement arrival, the sub-store stocks are, respectively, 3, 7, 1, 2, 3, a clear case of a poor distribution of stocks over the sub-stores. The high level of stock at sub-store 2 can be traced to too high a distribution from the "Share Mk V" rule at a time of 125 (see Exhibit 'B').

15.4 Introduction of Share Mk VA

Following the observance of this phenomenon, the Ration Rule "Share Mk VA" was introduced. In this rule, the shortage costs at the other sub-stores are always computed by hypothetically sharing out the "left stock" from the central store to them in as equalising a way as possible, and then taking the sum of the expected costs of shortages for each sub-store with the resulting stock levels.

Exhibit 'C' records the simulation using "Share Mk VA" and it is seen that the problem discussed above (with "Share Mk V") is alleviated.

15.5 Observance of Inefficient Use of Buffer Stock

In Exhibit 'C' one can see that with stock levels of 2, 5, 2, 3, 4 for the sub-stores at time of procurement arrival, we have an unsatisfactory situation with regard to shortages for sub-stores 1 and 3. This is seen to be the case from the result that a cost of 4.4 is sustained as a result of one backorder existing for 11 days at sub-store 1. Sub-store 3 does not acquire backup.

What is happening is that Share Mk VA is not efficiently using the buffer stock. At a clock-time of 169 in the simulation (lines "a", Exhibit "C") the Ration Rule depletes the central store and there exist 6 units for each sub-store to last until a simulation time 250 (£ after procurement arrival).

15.6 Results of Experiment Ten (incorporating Ration Rule Share Mk VA")

15.6.1 Case of above average demand in coverage time

Exhibit 'C' using Share Mk VA, had the following information extracted from it.

Clock Time	Stock Configuration (Notional Sub-store Stocks)					Central Store Stock
161	6	6	6	6	6	5
161	6	5	6	6	6	5
	6	6	6	6	6	4
163	5	6	6	6	6	4
	6	6	6	6	6	3
168	6	5	6	6	6	3
	6	6	6	6	6	2
168	6	6	6	6	5	2
	6	6	6	6	6	1
169	6	6	5	6	6	1
	6	6	6	6	6	0
Procurement Arrival						
225	2*	5	2	3	4	0

15.7 Reasons for Inadequacy of Share Mk VA

The Ration Rule "Share Mk VA" is still not equipped to deal with the problem of distinguishing between the desirability of having a stock configuration of (a) as opposed to (b) below:

Stock configurations	(a)	3,	3,	3,	3,	3,	(10)**
	(b)	5,	5,	5,	5,	5,	(0)

It has been shown (see reference to Exhibit 'B' on last page) that "lumping" the stocks to compute the expected cost of shortage for a configuration like (a) is incorrect. Yet by doing the other alternative (viz. considering stock to be distributed from the central store in an equalising manner and computing shortage costs correspondingly) we are saying that the above two configurations are equal in terms of expected shortage costs. Clearly for times before procurement arrival, (a) is better than (b) since stock is available in the central store to

---

* This store receives a backup of cost 4.4 units in next ℓ days.

** Central store stock.

supply to those stores really in need as time of real shortage in the complex approaches. It should of course be recognised that we cannot afford to keep too much stock back at the central store since individual stores will then be holding too little stock with consequent risk of shortage in the lead time.

15.8 How Can the Ration Rule be Modified to More Efficiently Use the Buffer Stock?

The Ration Rule has to be modified so as to more efficiently distribute the last few units of stock from the central store.

Clearly it is only for the last few* units of stock in the central store that the problem is really acute, and some "stronger rationing" is required. If we were to incorporate this "stronger rationing" at an earlier stage, the risk is run of incurring extra replenishments and maintaining dangerously low levels of stock at sub-stores over fairly long periods of time. It is seen from the simulation details that for most cases, the present ration rule "Share Mk VA" is adequate when central store stock is not very low.

Clearly, any ideas proposed must not be solely suitable for specialised cases, but must be such as to reflect the relative costs of shortage and replenishment.

15.9 Ideas Adopted to Modify "Share Mk VA" to "Share Mk VI"

After many hours of search for an analytical solution, the task was abandoned as being too difficult; a heuristic approach was adopted. To be acceptable, the Ration Rule must be able to strike the correct balance between overall shortage and replenishment costs for different combinations of unit shortage and unit replenishment costs.

15.10 Heuristic Determination of the "Stronger Rationing" Ideas for "Share Mk VI"

We do not want to let the stock at sub-stores get dangerously low, but we do not want other sub-stores to incur shortage as a result of maldistribution. This applies irrespective of unit replenishment cost. It is solely concerned with efficiently using the buffer stock.

At any time we can compute what stock will be expected to exist in total in the complex when the procurement is just about to arrive at the central store. If an estimate of this figure is 19 say, we can do no

---

*Specifically, this "stronger rationing" is invoked if central store stock equals mean total demand in the sub-store lead time +1, or less.

better than expect to have a configuration such as 3, 4, 4, 4, 4 at the sub-stores at this time. This means that the buffer employed is such that we are not expecting to have the sub-store stock level drop below 3 with replenishment due ℓ days hence. Hence it is unreasonable to let any sub-store stock ever drop below 3 before replenishing. Clearly a replenishment of one unit is the only one to be sensible, otherwise maldistribution is very likely.

The rule thus resolves into one of restricting the ration quantity to one more than the value of "Expected Stock at Procurement Arrival ÷ N". For the case cited above, this restriction in stock value will be 1 + 19/5 = 4.8; the integer value is taken (since the value of 5 would be above the restricted value) and so the ration quantity is thus restricted to 4.

15.11 Consideration of the "Stronger Rationing Ideas"

It is hoped that this rule will tend to result in an even distribution of stock when the procurement arrives in the central store. In the period equal to the sub-store lead time after this time instant is the time when shortages in the complex are most likely, so it is important to try to ensure that at this time there is even distribution of stocks amongst the sub-stores.

If overall stocks are low when the "stronger rationing" rule is invoked, we expect shortages anyway, but we are minimising the overall cost of shortage. Where shortages do not matter very much, the 1st Dynamic Programming Model reflects this in making the sub-store reorder level low anyway.

15.12 Is the Stronger Rationing Rule Excessively Liberal in Replenishment?

An argument may be voiced that by restricting the quantity rationed out to an ordering sub-store, another replenishment at a later date will be necessitated. This question will now be considered for the cases where (i) expected buffer is high and (ii) expected buffer is low.

The following comments are felt to apply equally well whether unit shortage cost is high or low.

Case (i) Buffer High.

The replenishment consists of several units, and the possibility of further replenishment is minimised.

Case (ii) Buffer Low

The replenishment is correspondingly small in terms of units of stock shipped, but another replenishment to the sub-store is not really likely, since with buffer low, the other sub-stores are likely very soon to be at their reorder level, and the remaining few units in the central store can, for most cases, be considered "reserved" for these sub-stores.

15.13 Incorporating the New Ration Rule

The New Ration Rule, known as "Share Mk VI", is used for the simulation of Models IV and V and was seen to be beneficial, although it cannot always be expected to result in an even distribution of stocks. The simulation of Model IV with the new Ration Rule "Share Mk VI" will be known as Experiment 10A and the control known as Control 10A.

15.14 Summary of Experiment 10A

4 Year Simulation of Model IV with Control 10A Using IBM 1130 System with "Simon" Simulation Language

Full Description of Model given in 15.1

Control 10A identical to Controls 8A, 9 (see Experiments Nos. 9,10) with following exception:

Sub-store Reorder Quantity:

Case 1 : Procurement on Order: Ration Rule "Share Mk VI" (see this chapter).

15.15 A Note on the Results of Experiment 10A

The results of the application of Share Mk VI to Model IV for the case (M = 50, z = 0), are very illuminating. In comparing it with the results from Exhibit 'C' using Share Mk VA, we note a more even distribution of stock at the critical time, just prior to stock arrival from the new procurement.

Criticism may still be given that at procurement arrival sub-store 1 has a stock level of 2 whilst sub-stores 2 and 6 each have 4. Careful consideration of the simulation will, however, show that to avoid this occurrence, one would have had to have retained a policy of non-replenishment for all sub-stores from clock-time 193 onwards. This would mean waiting for stock levels of 2 before replenishment. This would mean considerably more shortage cost then the present policy (of the new "stronger rationing" rule) leads to, where only once does such a dangerously low stock level occur before replenishment follows.

15.16 An Extraction from the Simulation of Experiment 10A Application of "Share Mk VI to Model IV for the Case M = 50, z = 0

Clock Time	Sub-store Stocks					Central Store Stock	Ration Quantity from Share Mk VI
161	6	5	6	6	6	(5)	5
163	5	5	6	6	6	(5)	5
168	5	5	6	5	6	(5)	5
168	5	4	6	5	6	(5)	4
169	5	4	6	5	5	(5)	
173	5	4	5	5	5	(5)	
175	5	4	4	5	5	(5)	4
177	5	4	4	5	4	(5)	4
179	4	4	4	5	4	(5)	4
185	4	4	4	4	4	(5)	4
193)	3	4	4	4	4	(5)	4
)	4	4	4	4	4	(4)	
199)	4	4	3	4	4	(4)	4
)	4	4	4	4	4	(3)	
206)	4	4	4	4	3	(3)	4
)	4	4	4	4	4	(2)	
207)	4	3	4	4	4	(2)	4
)	4	4	4	4	4	(1)	
212)	4	4	4	3	4	(1)	4
)	4	4	4	4	4	(0)	
212	3	4	4	4	4	(0)	
216	3	4	4	3	4	(0)	
218	2	4	4	3	4	(0)	
223)	2	4	3	3	4	(0)	
225)							

15.17 Results of Experiment 10A

15.17.1 Table 15-1

Results for Test Stream No. 1729

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
45	-.6	5	122	13.6	48926	39.3303
	-.3	5	103	13.6	48926	38.7603
	0	5	84	13.6	48926	38.1903
	+.3	5	69	13.6	48926	37.7404
	+.9	5	37	13.6	49511	37.0143
	+1.2	5	31	20.4	51193	44.3071
50	.6	5	56	4.4	53903	30.1411
	1.2	5	35	4.4	53903	29.5112 (N.B.)
	1.5	5	27	11.6	55181	36.9823*
55	1.2	5	40	0.0	58892	27.2567 (N.B.)
	1.5	5	31	4.4	60163	31.8951*

Stream No. 1115 Results

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
50	1.2	4	34	0	55431	25.1923
55	1.2	4	34	0	60431	27.1923

Mean of 2 Streams

M	z	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	Area	T.C.
50	1.2	4.5	34.5	2.2	54667	27.3517
55	1.2	4.5	37.0	0.0	59661	27.2245

The latter is the optimum of these results.

15.18 Summary of Experiment 11

4 Year Simulation of Model V with Control 10B Using IBM 1130 System with "Simon" Simulation Language

Full Description of Model given in 15.1

---

* Central store initially depleted; No buffer available from Central Store.

Control 10B identical to that for Experiment 10A (see 15.14) with following exception:

Complex Reorder Quantity Q = 75.

15.19 Results of Experiment 11

Table 15-2

Case	z	Stream No.	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	H.C.	T.C.
M=50	0.3	1115	3	43	0.0	26.40	40.80
	0.6	1115	3	34	0.0	26.85	38.55
		1729	3	36	20.8	27.42	60.52
		Mean	3	35	10.4	27.13	49.53
	1.2	1115	3	19	1.2	29.85	38.25
		1729	3	20	41.6	26.47	79.57
		Mean	3	19.5	21.4	28.16	58.91
M=55	0.6	1115	3	31	0.0	28.40	39.20
		1729	3	38	1.6	28.41	42.91
		Mean	3	34.5	0.8	28.40	41.05
	0.9	1115	3	24	0.0	30.26	38.96
		1729	3	32	1.6	29.40	42.10
		Mean	3	28	0.8	29.83	40.53
	1.2	1115	3	23	1.6	29.51	39.51
		1729	3	24	1.6	32.31	42.61
		Mean	3	23.5	1.6	30.91	41.06
M=60	0.9	1115	3	25	0.0	30.43	39.43
		1729	3	33	0.0	30.40	41.80
		Mean	3	29	0.0	30.42	40.62
	1.2*	1115	3	22	0.0	30.85	38.95
		1729	3	25	0.0	31.39	40.39
		Mean	3	23.5	0.0	31.12	39.67
	1.5	1729	3	24	1.6	32.31	42.61

cont..

* This is the least-cost result.

Table 15-2 (cont.)

Case M=65	z	Stream No.	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{RO}$	H.C.	T.C.
	0.9	1115	3	25	0.0	32.37	41.37
		1729	3	29	0.0	32.40	42.60
		Mean	3	27	0.0	32.39	41.99
	1.2	1115	3	23	0.0	32.40	40.80
		1729	3	27	0.0	32.40	42.00
		Mean	3	25	0.0	32.40	41.40
	1.5	1115	3	22	0.0	33.24	41.34
		1729	3	21	0.0	34.34	43.14
		Mean	3	21.5	0.0	33.79	42.24

15.20 Details of Computing Procedure for Models IV, V

1. Use Program "NNNN" with subroutine "SUBX" to obtain the 1st D.P. Model. Use this information in Programs, "DSJF", "MMMM".

2. Run program "DSJF" to obtain off-line the D.P. Model 2. Store this information in the machine data file.

3. Follow on immediately with the simulation program "MMMM" with its various subroutines.

The complete simulation program for Experiment 11 is given as Program 2. This is the program of control type 10B incorporating the latest-modified Ration Rule "Share Mk VI". This control, apart from this ration rule, and a use of Q = 75 (due to a difference in model data), is identical to that of Control 8A described in 14.3.1. It is applied to Model V (details of which are given at 15.1) and the results are illustrated by means of equi-cost curves in Fig.12. Figure 11 shows the variation of the cost of an (M,z) policy with the parameter M for a close-to-optimal z value.

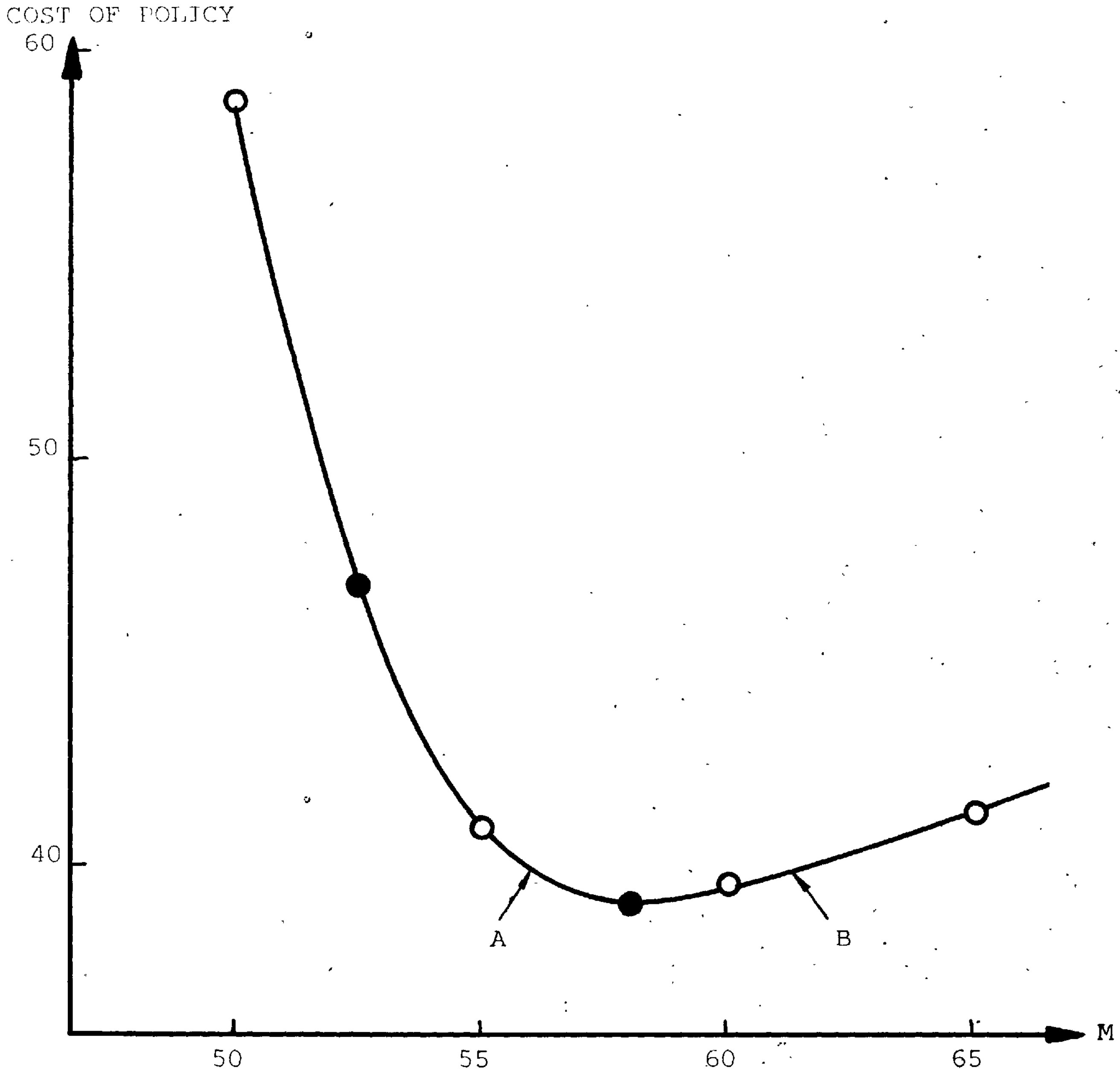
15.21 Details of Program 2

Flow diagrams are given for the main programs "NNNN", and "MMMM" (referred to above in 15.20) and for the more important and interesting subroutine calculations.

The proposal here is to list all the programs and subprograms used in the complete simulation program for Experiment 11, with a brief description of their purpose.

FIG. 11

AVERAGE 4 YEAR COST OF (M,z) POLICY FOR CLOSE-TO-OPTIMAL z (z = 1.2) PARAMETER FOR EXPERIMENT 11



LEGEND

○ POINTS OBTAINED BY SIMULATION

● POINTS OBTAINED BY CUBIC APPROXIMATION USING SIMULATION POINTS

POINTS A, B, GIVE POINTS A, B, FOR FIG. 12

FIG. 12
EQUI-COST CURVES FOR EXPERIMENT 11 (SIMULATION OF MODEL V)

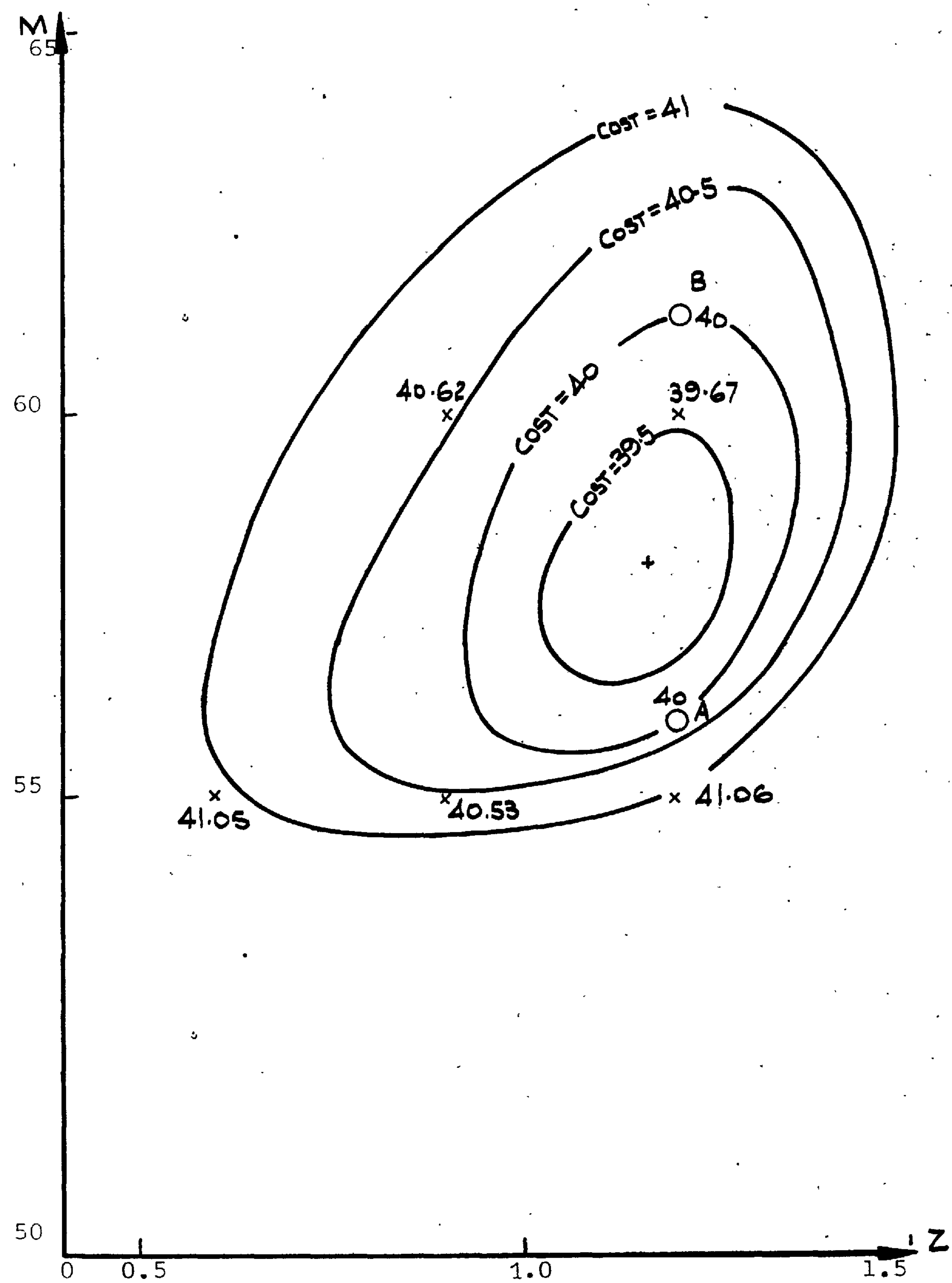
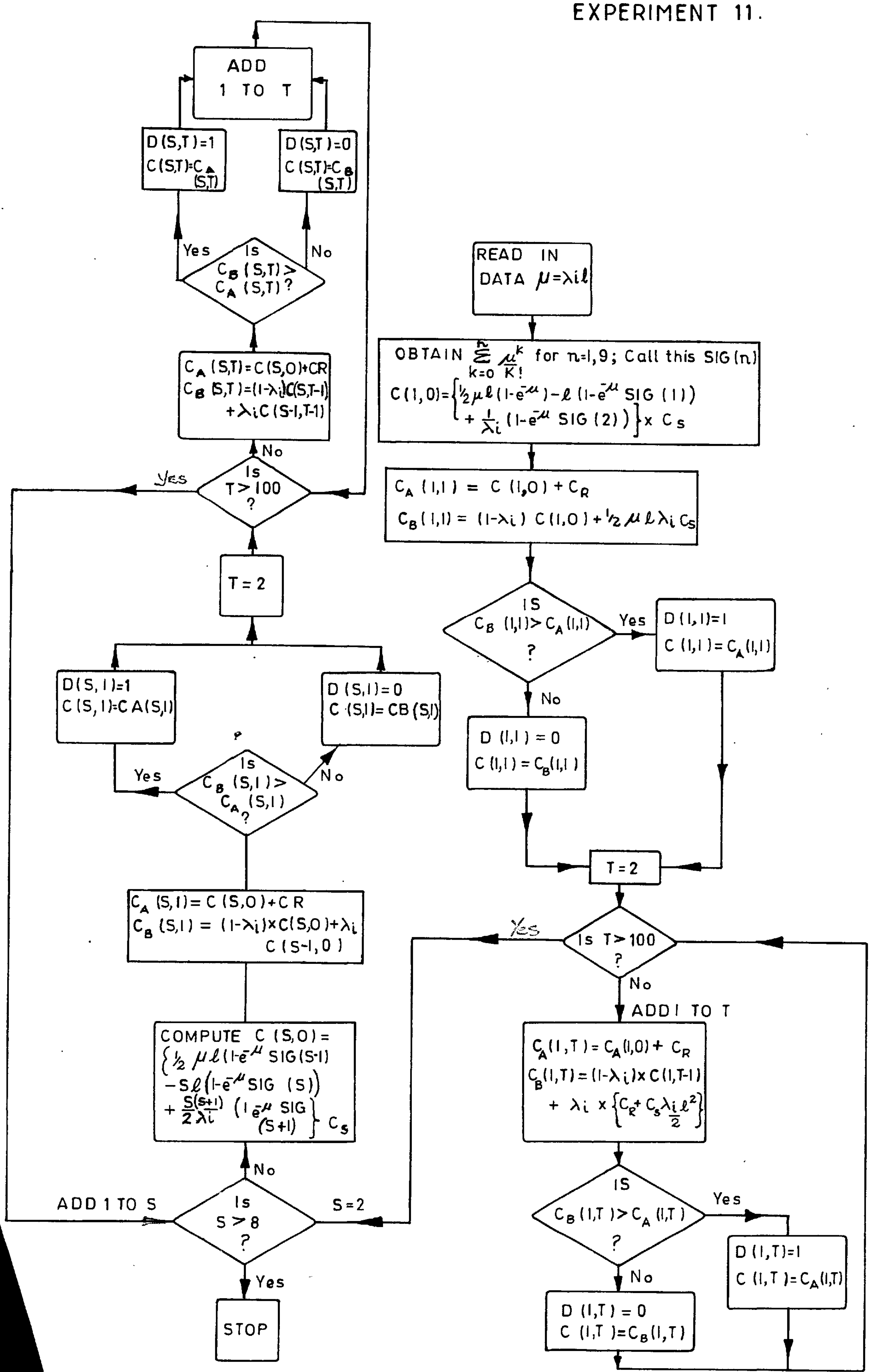


FIG. 13 FLOW DIAGRAM FOR OFF LINE COMPUTATION OF THE FIRST DYNAMIC PROGRAMMING MODEL: "NNNN / SUBX"
EXPERIMENT 11.



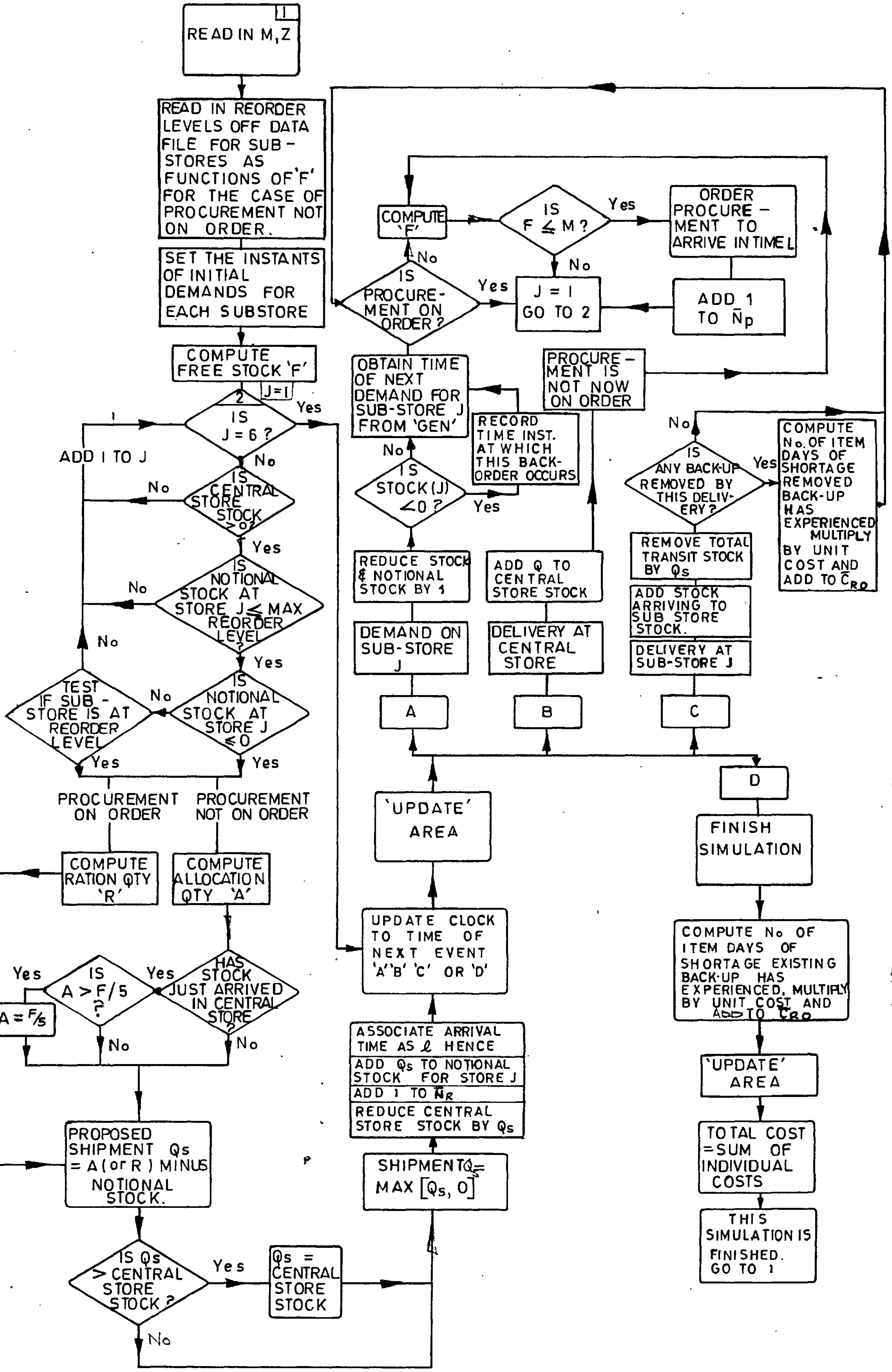


FIG. 14 FLOW CHART FOR SIMULATION IN EXPERIMENT 11.

FIG.15 FLOW DIAGRAM FOR COMPUTATION OF INDIVIDUAL SHORTAGE COST $C_s\psi(\bar{S},\bar{L},\bar{\lambda})$ IN EXPERIMENT 11 "CALCS (ISDA,ILDA,RLAMD)"

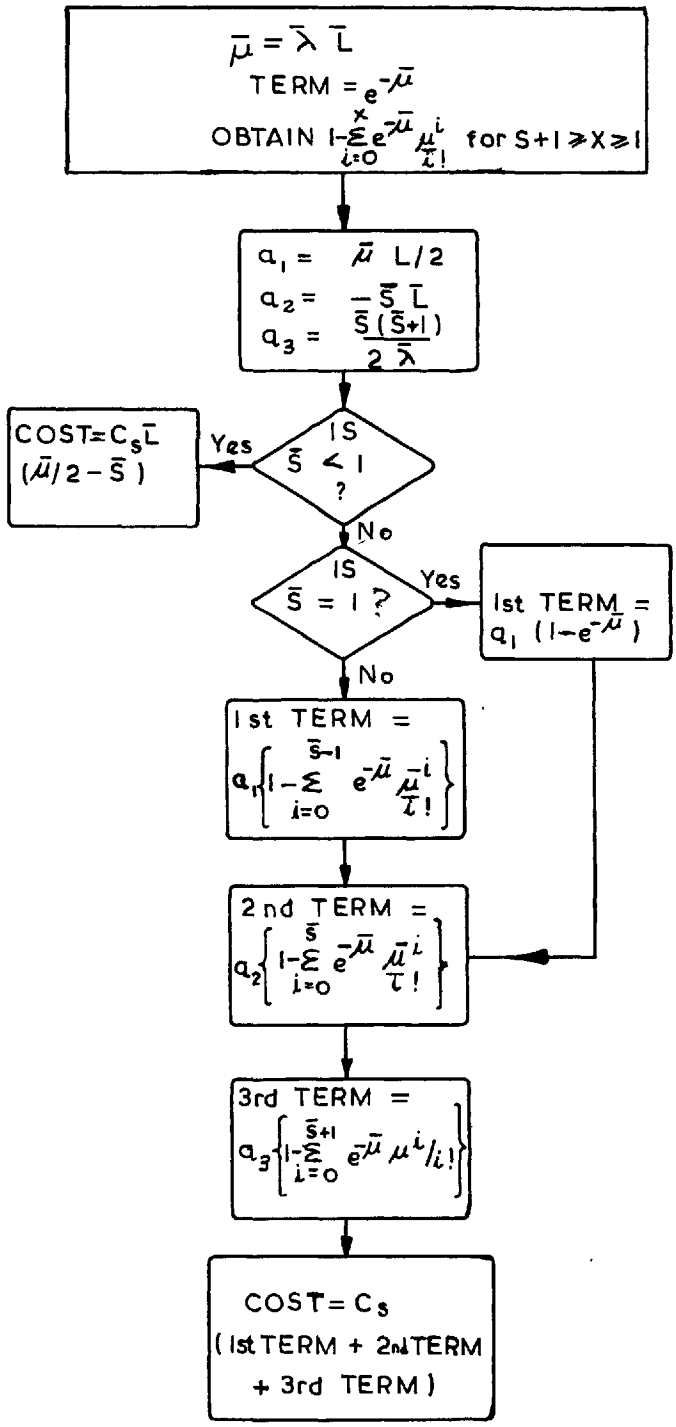


FIG.16 FLOW DIAGRAM FOR UPDATING OF AREA IN EXPERIMENTS 11,12.

[SUBPROGRAMS 'CALG', 'CALGX' ARE FROM STATEMENT 1 TO STATEMENT 2.]

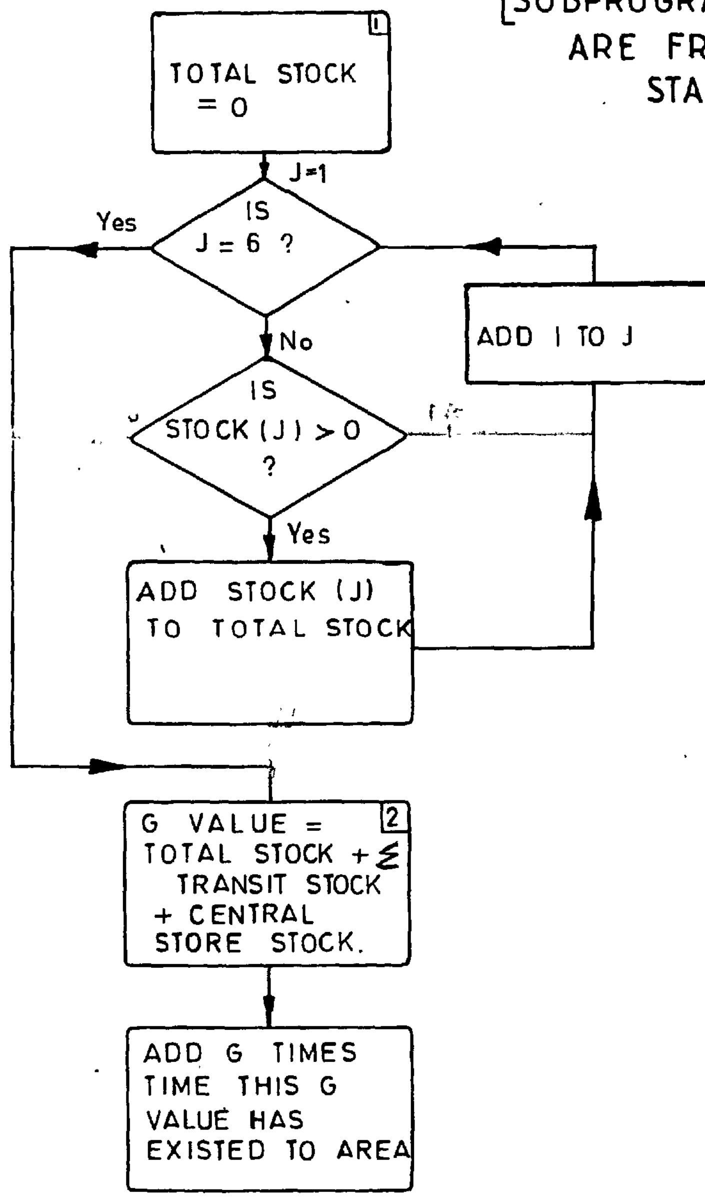


FIG. 17 FLOW CHART FOR COMPUTATION OF K, Σb_i IN EXPERIMENTS 11,12 SUBPROGRAMS "CALKB(D)" AND "CALBX (D)"

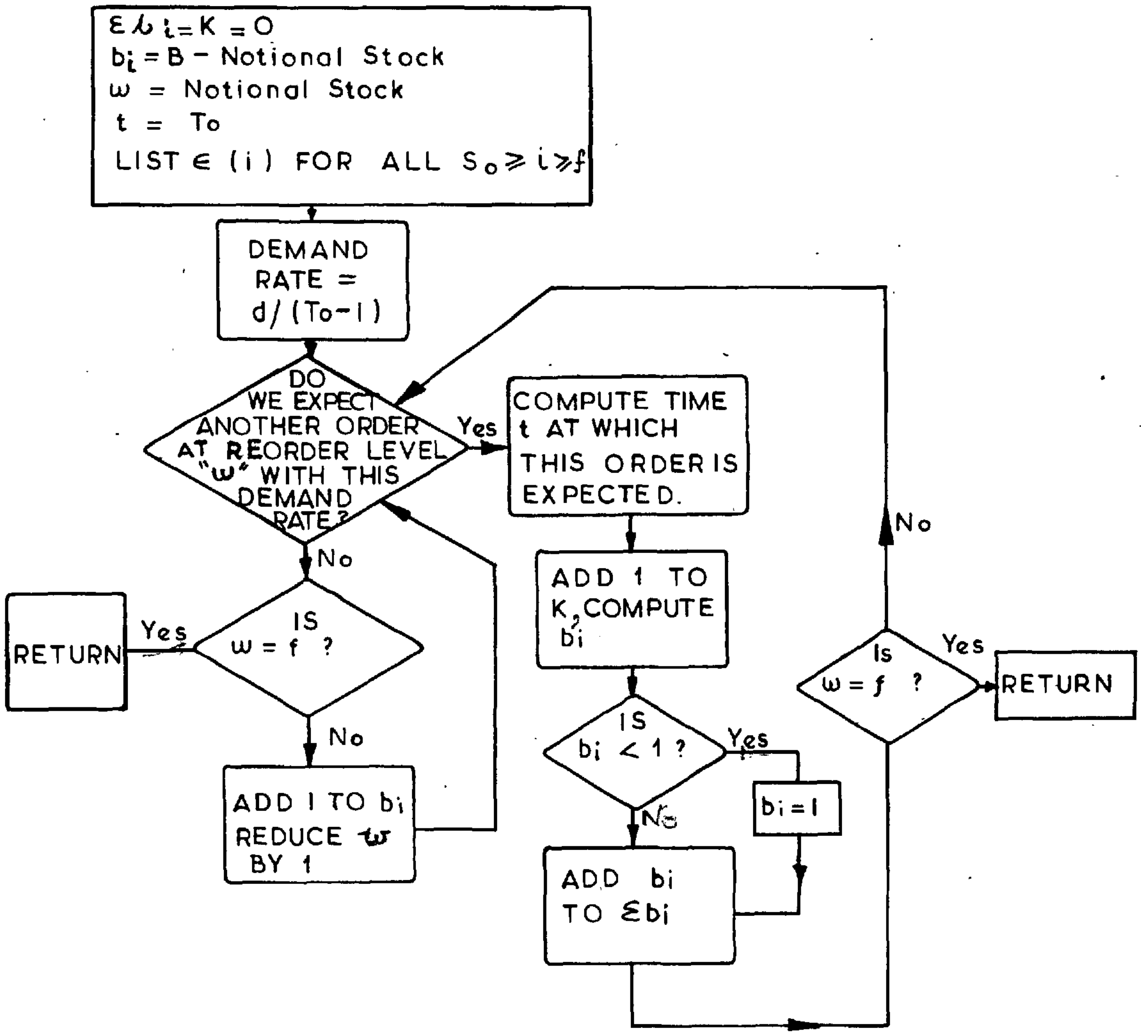


FIG. 18 FLOW CHART FOR COMPUTATION OF RATION QUANTITY IN EXPERIMENTS 11 12 SUBPROGRAMS "Pool", "PooX"

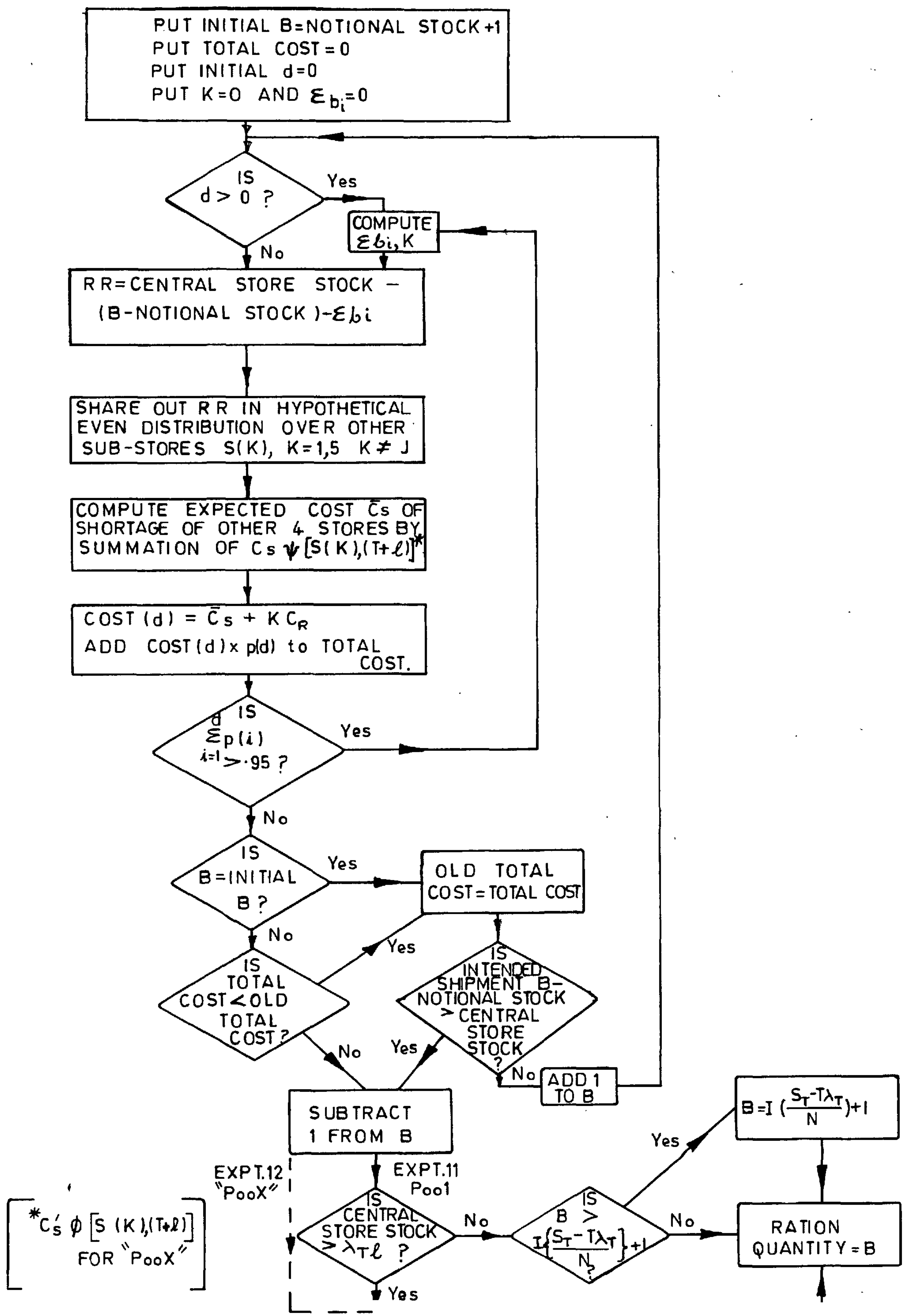
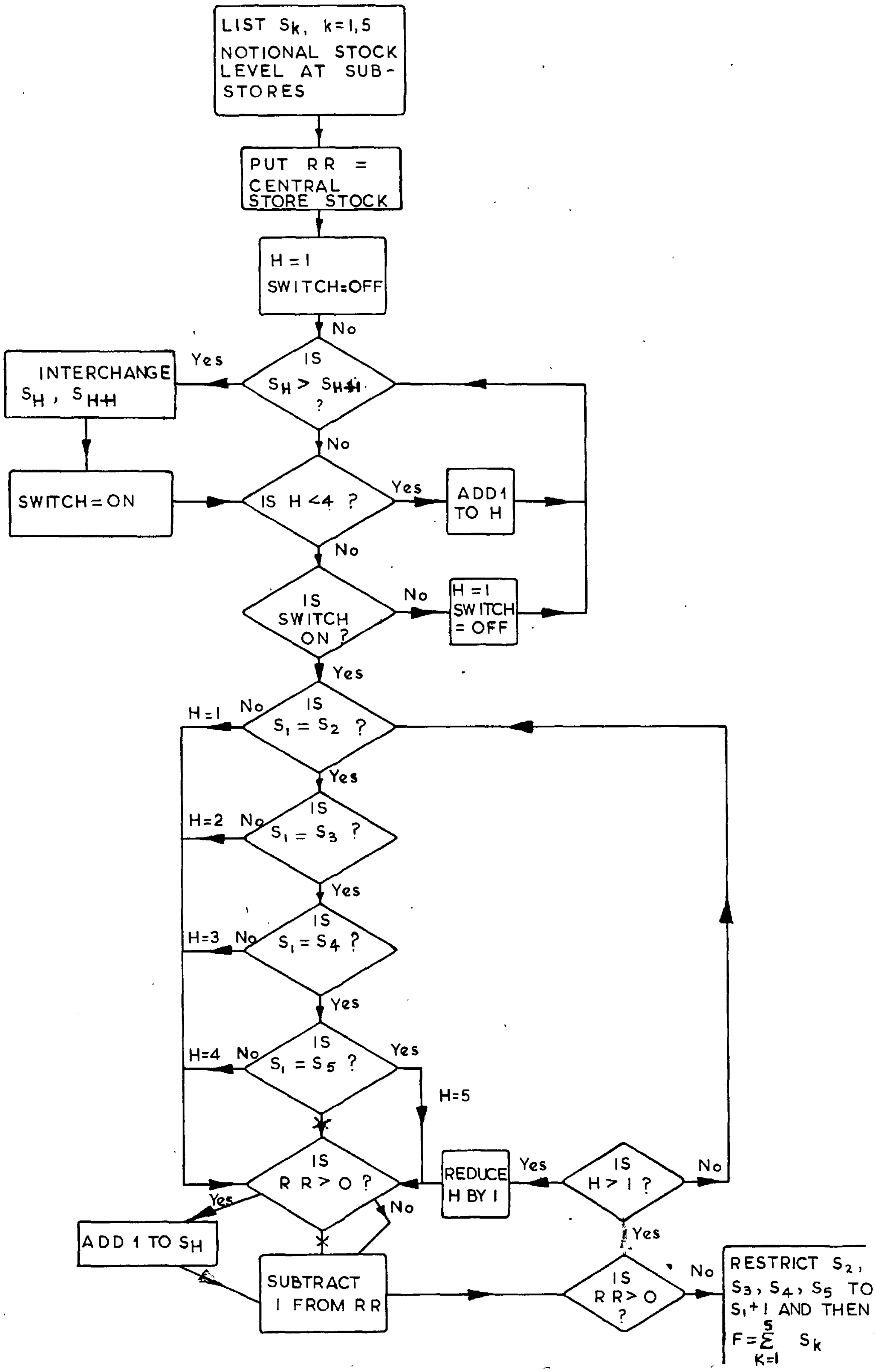


FIG. 19 FLOW DIAGRAM FOR FREE STOCK COMPUTATION FOR SUB PROGRAM "SFRS"



15.21.1 Program "NNNN" + Subprogram "SUBX":

Off-line computation of reorder level at sub-stores as a function of time to arrival of procurement at central store. Actually the decision function for different notional stock level and T combinations are given, but this amounts to the same thing. The flow diagram for this computation is given in Fig. 13.

15.21.2 Program "DSJF" + Subprograms "DSJG", "SUBA", "SUBC"

Off-line computation of reorder level at sub-stores as a function of Free Stock. This information is kept on a data file.

15.21.3 Program "MMMM":

Main simulation program, consisting of the initial conditions plus the three phases of the Simon Simulation Procedure. The flow diagram is given in Fig. 14.

15.21.4 Subprograms called from "MMMM"

(i) "CALCS (ISDA, ILDA, RLAMD)"

This subprogram computes the expected cost of shortage for sub-stores in the coverage time as a result of a possible ration quantity. The three arguments are respectively the values of $\overline{S}$, $\overline{L}$, $\overline{\lambda}$ used in the $\psi(\overline{S}, \overline{L}, \overline{\lambda})$ computation. The flow diagram for this subprogram is given in Fig.15.

(ii) "CALG"

This subprogram computes the G-value of the complex and updates the value of the total stock~time area function "Area". Figure 16 gives the flow diagram for this computation.

(iii) "CALKB(D)"

This computes the expected number of extra replenishment K(d,B) as a result of a ration quantity B when experienced demand until the time T=1 is d, and the total number of shipped items in these extra deliveries,

$\sum_{i=1}^{K(d,B)} b_i$. The flow diagram is given in Figure 17.

(iv) "GEN"

This takes a random variable from the Poisson distribution of sub-store demands and associates the time of the next demand at a sub-store which has just experienced a demand as this time value hence.

(v) "PNOO"

This subprogram calculates the Allocation Quantity A(F,M,z).

(vi) "POO1"

This subprogram calculates the Ration Quantity. It is described by the flow diagram in Fig. 18.

(vii) "SFRS"

This represents the computation of "Free Stock" F. The relevant flow diagram is given by Fig. 19.

(viii) "SSRO"

This subroutine tests whether any sub-store is at its reorder level and calls the relevant subprogram, either "PNOO" or "POO1", if positive.

(ix) "SYRO"

This checks to see if Free Stock is less than the reorder level of the complex and, if so, initiates a procurement order.

(x) "PLAN, ADDL, BEHE, GROU, HEAD, MEMN, REFN, SCA, QUEUE, SETT SIZE, TIMV, SIMO, DELE, ENTI"

are the special purpose subroutines of the "Simon" Simulation procedure.

15.22 Comments on the Results of Experiment Eleven

For the case of stream 1729, in the first cycle, a buffer of 16 at time of procurement arrival* spread over 5 sub-stores means an average of three to each sub-store, and so in the days when the central store stock is below 10 (in the region of half the central store lead time prior to procurement arrival) sub-store reorder level is made by the rationing rule equal to 3. The probability of a demand of greater than 3 in the sub-store lead time (when mean demand is 2) is as much as 0.14! Hence not only does this buffer spell shortage in the sub-store lead time immediately following procurement arrival at the central store, but also in the sub-store lead times prior to procurement arrival.

The "stronger rationing rule" "Share Mk VI" although of some obvious merit, is felt to be unacceptable for future considerations because of its heuristic context, and is dropped from future considerations. The previous best Ration Rule "Share Mk VA" will be employed in the following work.

Note that Chapter 18 gives a summary of the Ration Rules used in this thesis.

* Corresponds to M=50.

15.23 Summary of Chapter Fifteen and Introduction to Chapter Sixteen

Chapter Fifteen introduces the two new models, Models IV, V, the controls of the author are applied to each and the results are presented. The application of "Share Mk V" to Model IV, as shown in Exhibit 'B' results in a poor stock distribution at a critical time. This situation is seen to improve with modification to "Share Mk VA", and an extract from Exhibit 'C' is given which shows this. Further modification for the purpose of improvement of performance is suggested, and Experiment 10A considers the latest suggestion "Share Mk VI" in its application to Model IV. Experiment 11 uses this Ration Rule on Model V. A full program for the simulation for Experiment 11 (control type 10B) is given. A summary of the details of this program is given at 15.21 and flow diagrams for the important operations are given.

The results of Experiments 10A and 11 (given in Tables 15-1, 15-2 respectively) show that a low overall cost with zero expected shortage is possible for use of low buffer stock in the complex. An extraction from the simulation of Experiment 10A is given at 15.16 in which the operation of the all-important ration rules may be viewed.

Chapter Sixteen turns to a more general model of the complex. At the sub-stores the demands are forecast from past data. The performance of the author's control rules is to be tested in their control of this model. An interesting point is that a jump in mean demand rate is applied to one sub-store within the time of control of the simulation.

CHAPTER SIXTEEN

CASE OF CONTROL WHEN DEMANDS COME FROM AN UNKNOWN DISTRIBUTION AND ARE SIGNIFICANTLY DIFFERENT FOR SUB-STORES

16.1 Introductory Remarks

This next phase of the work considers the application of the author's control to a model of the complex in which sub-store demands are forecast from past data. Additionally, the model must cope with the mean demand rates at sub-stores being different. In fact the model will feed in random variables from given distributions, but the control will work in ignorance of the latter, just making forecasts of demand rates from past data.

The model will work with the assumption of shortage cost at sub-stores being proportional to the number of demands not met immediately. The time factor in shortage will not be considered as contributing to cost.

16.2 Model VI

This model considers shortage costs to be independent of time and each shortage occurring will be considered to cost $c_s' = 0.4$. The mean demand rate λ_i vector for the five sub-stores is (.01, .02, .03, .04, .05) units/day, Poisson distributed for sub-stores 1 to 5 respectively.

The lead time to sub-stores is 25 days, and the procurement lead time (as with all other models) is 100 days.

The value of c_R is .3, $c_P = .5$, hP = .1.

Four-tenths of the way in the 4-year simulation, the value of λ_3 changes suddenly to .06/day.

Knowledge of sub-store demand rates is only gained from demand occurrence data. Future sub-store demand rates are forecast for each sub-store.

16.3 Modification of Concepts to Cater for the Change in Model Conditions

16.3.1 Trigger of reorder for the complex

This is still "Free Stock". However, now stock from the central store is considered distributed (hypothetically) to the sub-stores in such a manner as to attempt to equalise the expected time stock will last sub-stores ("stock-lasting time").

When no more stock is available in the central store in this hypothetical distribution, Free Stock is taken as the product of the minimum of the resulting "stock lasting times" and the total demand rate, λ_T, of the complex.

A flow diagram for this computation is given in Fig. 26.

16.3.2 Demand rates

Since for Model VI, no exact knowledge of the distributions from which the sub-store demands are coming is assumed, the demand rates require to be forecast.

It is convenient to consider inter-arrival times of demands, and work a forecasting procedure on these. Sub-store demand rate will then be the inverse of this forecast inter-arrival time. The total demand rate for the complex, λ_T, is found by summing λ_i over all sub-stores.

The forecast used here is that of exponentially smoothing the inter-arrival times. We have for a particular sub-store:-

$$T_{f_n} = \alpha T_{f_{n-1}} + (1 - \alpha) T_p$$

where T_{f_n} is the new forecast

$T_{f_{n-1}}$ is the old forecast

T_p is the last inter-arrival time observed

α is the smoothing constant.

16.3.2.1 Obtaining the initial forecasts

Demands over a period equal to the length of time that the model is being simulated over are fed in for each sub-store, and these are exponentially smoothed. The resulting T_{f_n} and its inverse represent the initial inter-arrival time forecast and demand rate, respectively. This is done for each sub-store.

16.3.3 Reorder level of sub-stores

Case One: Procurement on order

This employs the off-line computation of the reorder level $\sim$ time function for "strategic" values of the demand rate λ_i. (The values of λ_i = .01 to .10 are considered for this model in .01 steps.) This information is stored in the computer by associating the time at which the step 'drops' with the level at which this occurs. Thus, in Fig. 10 (see 14.2.1) the value of the step-end "ε" for level s_{p_1} would be equated to T_{p_1}*. (Note that the value of "ε" for the level immediately above that corresponding to the time T=L is not equated to L; rather the exact value at which this step drop occurs is found in the dynamic programming off-line computation.)

For every value of sub-store notional stock existing the "ε" value is estimated by linear interpolation between the strategic values for which accurate dynamic programming "ε" values of the complex are stored. Reference to Fig. 20 shows this to be a good approximation. This figure

* i.e. the function $\varepsilon(\lambda_i, s_{p_1}) = T_{p_1}$.

FIG. 20

REORDER LEVEL FOR THE (λ_1, T) PLANE FOR THE 3rd D.P. MODEL FOR MODEL VI IN EXPERIMENT 12 (SHORTAGE COSTS TIME-INDEPENDENT)

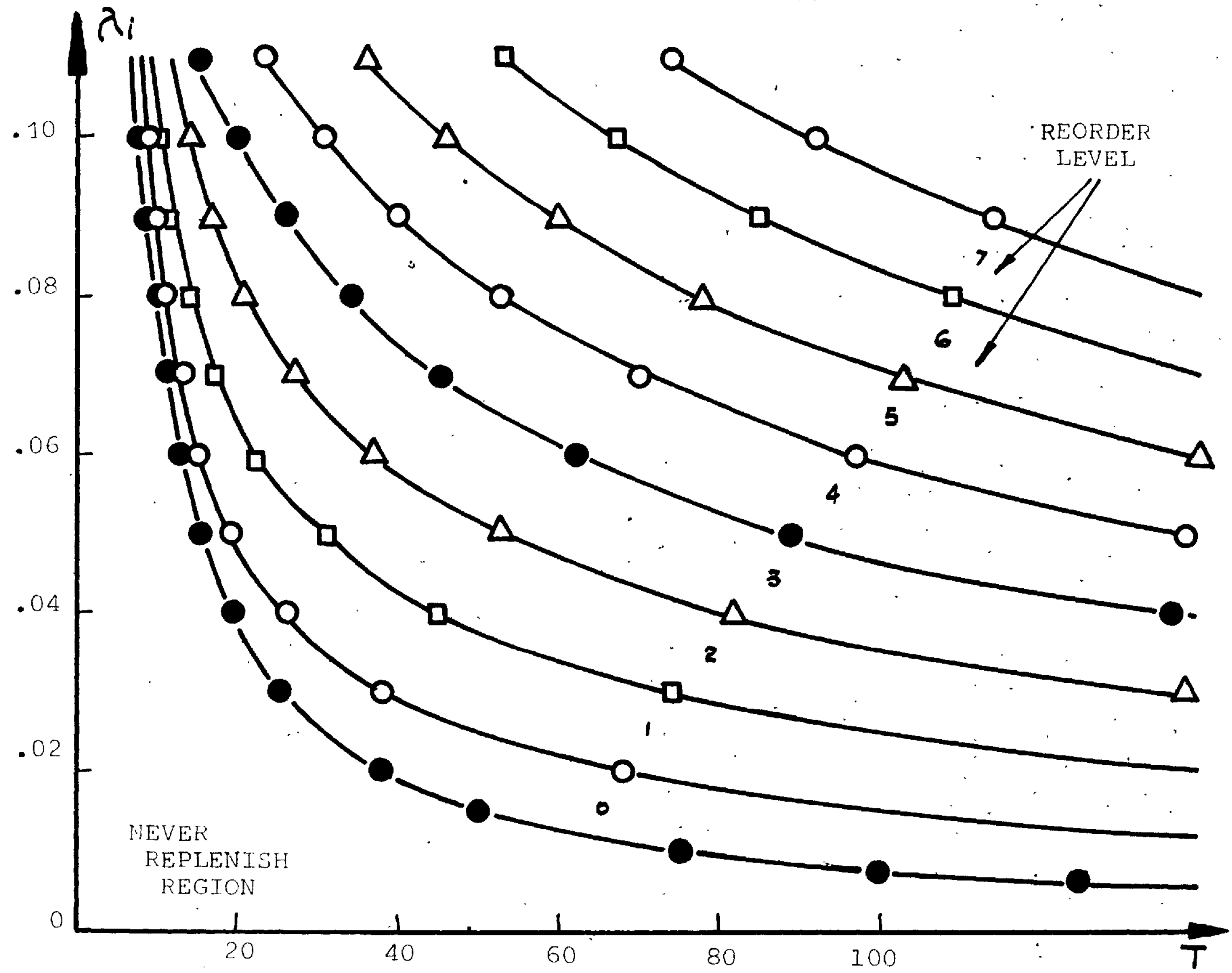
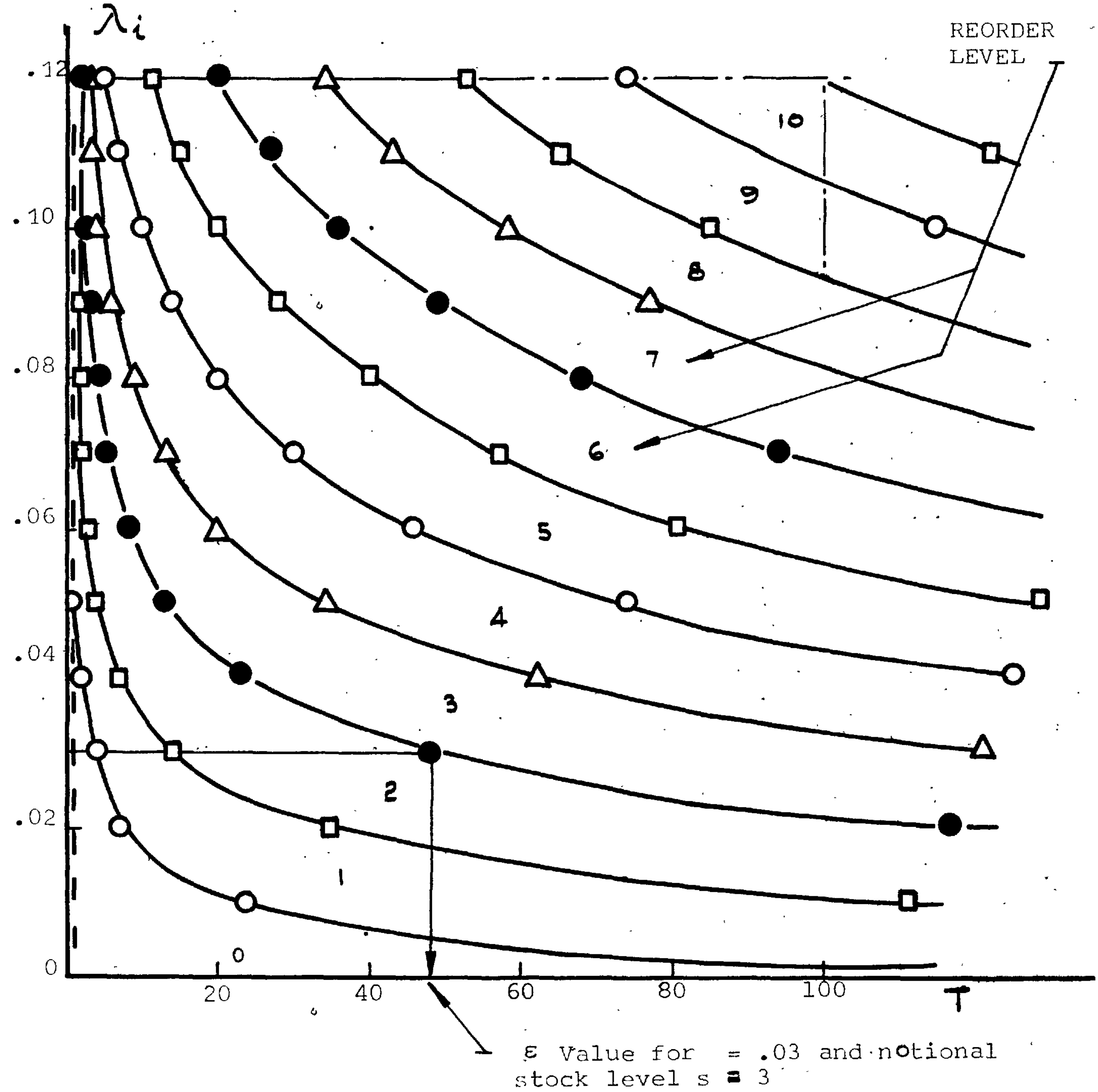


FIG. 21
REORDER LEVEL FOR THE (λ_i,T) PLANE FOR THE NON-SIMULATED CASE OF THE 1st D.P. MODEL WITH C_s = 0.4 APPLIED TO MODEL VI (SHORTAGE COSTS TIME-DEPENDENT)



shows the reorder level regions on the $\lambda_i \sim T$ plane.* ** The "ε" value for any given $\lambda_i \sim$ notional stock level 's' combination is found by obtaining the (s,s-1) boundary for this λ_i and taking the corresponding value of T.

Thus when forecast demand is, say, .07234 and notional stock s is 6, estimated ε is $.234\varepsilon(.07, 6) + .766\varepsilon(.08,6)$. If the actual value of T is such that $T \geqslant$ estimated ε, the sub-store is deemed to be at its reorder level, and its issue quantity is decided by the subprogram "POOX" utilizing the Ration Rule.

A flow diagram for the computation of ε is given in Fig. 24.

The reorder level $\sim$ time function for Case One requires the use of a 3rd Dynamic Programming Model.

16.3.3.1 The Third Dynamic Programming Model

This is very similar to the 1st d.p. Model. The only basic change is that the shortage cost is now independent of time and a cost c_s' occurs only at the instant a backorder is experienced.

The analysis is identical to that given in 9.2, except that the expected cost of shortage of a sub-store with stock s when replenishment will definitely arrive in time ℓ is not the function $c_s\psi(s,\ell, \lambda_i)$ but the function $c_s'\phi(s',\ell, \lambda_i)$ where:-

$$\phi(s',\ell, \lambda_i) = \sum_{x=s'+1}^{\infty} (x-s') \, p(x)$$

where $p(x) = e^{-\lambda_i \ell} \dfrac{(\lambda_i \ell)^x}{x!}$ and s' is Max {s,0}

16.3.3.1.1 Decision functions for negative notional stock

Appendix 6.3 shows that for the 1st d.p. Model $D(-n,1) = 1$ for all $n \geqslant 1$ where $c_R < c_s(n+\lambda_i\ell)$.

Appendix 6.3 shows that if $D(s,1) = 1$ for a general s, then $D(s,T) = 1$ for all $1 \leqslant T \leqslant L$.

In a similar manner to Appendix 6.3, consider the (s,T) state (-n,1) for the 3rd d.p. Model. As before, $C_A(s,T)$, $C_B(s,T)$ refer respectively to the cost of not replenishing and the cost of replenishing a sub-store at the state (s,T).

---

* This graph is determined from the 3rd d.p. Model, details of which are given in 16.3.3.1.

** The reorder level regions which would exist if in fact shortage costs are dependent on time are shown in Fig. 21 (case $c_s = 0.4$).

For all $n \geqslant 0$, we have

$$C_A(-n,1) = c_R + c_s' \phi(0, \ell, \lambda_i)$$

$$C_B(-n,1) = (1-\bar{\lambda}_i)C(-n,0) + \bar{\lambda}_i C(-\overline{n+1}, 0) + \lambda_i c_s'$$

$$= c_s' \phi(0, \ell, \lambda_i) + \lambda_i c_s'$$

Thus $D(-n,1) = 1$ for all $n \geqslant 0$ only where $\lambda_i c_s' > c_R$ (for no cases in Model VI can we ever expect $\lambda_i > c_R/c_s'$).

If we denote $c_s'\phi(k, \ell, \lambda_i)$ by c_{sk}, then for $T = 2$,

$$C_A(-n,2) = c_R + c_{so}$$

$$C_B(-n,2) = (1-\bar{\lambda}_i)C(-n,1) + \bar{\lambda}_i C(-\overline{n+1}, 1) + \lambda_i c_s'$$

$$= (1-\bar{\lambda}_i)(\lambda_i c_s' + c_{so}) + \bar{\lambda}_i(\lambda_i c_s' + c_{so}) + \lambda_i c_s'$$

$$= 2\lambda_i c_s' + c_{so}$$

Similarly,

$$C_B(-n,T) = T\lambda_i c_s' + c_{so}$$

$$C_A(-n,T) = c_R + c_{so}$$

For Model VI in particular, we have

$$D(-n,T) = 0 \quad \text{for } T\lambda_i c_s' < c_R$$

$$D(-n,T) = 1 \quad \text{for } T\lambda_i c_s' \geqslant c_R$$

The decision boundary is T_b where

$$T_b = \frac{c_R}{c_s'\lambda_i}$$

$$= .75/\lambda_i \text{ for Model VI.}$$

For $T \geqslant T_b$, $C(-n,T) = c_R + c_{so}$

$T < T_b$, $C(-n,T) = T\lambda_i c_s' + c_{so}$

16.3.3.1.2 <u>Some other cost functions</u>

$$C_A(1,1) = c_R + c_{s1}$$

$$C_B(1,1) = (1-\bar{\lambda}_i)\, c_{s1} + \bar{\lambda}_i c_{so}$$

$C(n,0) = c_{sn'}$ for all n where $n' = \max(n,0)$.

16.3.4 Reorder level of sub-stores

It is considered satisfactory to make the reorder level in the period when the complex does not have stock on order equal to that corresponding to the time T=L (the time of initiation of the procurement order) from the 3rd Dynamic Programming Model. A dynamic programming model similar to the Second Dynamic Programming Model of this thesis with reorder levels as a function of Free Stock is not available to cope with the modified concept of Free Stock. (Further, at this stage, the value of this Second Dynamic Programming Model is itself considered to be minimal.)

This suggestion for reorder level does work well. It tends to be better at avoiding Maldistribution Type 2 (by postponing replenishment until really considered necessary) than the 2nd d.p. Model. The resulting reorder level versus demand rate function (ROL $\sim \lambda_i$) is the step function illustrated in Figure 28. (Figure 27 illustrates the step function which would result if the cost of backup were proportional to time of shortage for the case $c_s = 0.4$ and otherwise the same data of Model VI.)

16.3.4.1 Reorder level for very low λ_i

The 3rd d.p. Model shows that for Model VI, the decision at T = L is "never to replenish" for λ_i below the level .0075. Clearly, we cannot sensibly establish a reorder level at sub-stores which is less than 0 whilst a procurement is not on order. For those $\lambda_i \leq c_R/c_s'L$ (see the analysis for the negative stock decision boundary in 16.3.3.1.1) the reorder level for the period when procurement is not on order is equated to 0.

16.3.5 Reorder quantity for sub-stores

Case One: Procurement not on order

The Allocation Quantity is the expected demand in the coverage time plus "z" times the standard deviation of demand in this time.

Thus assuming that the mean and variance of demand in the coverage time when working on a Free Stock trigger can be approximated to that when the trigger works on System Stock (as for all previous models considered) we have, with the aid of Appendix 1,

$$\text{Mean} = (F-M)\frac{\lambda_i}{\lambda_T} + \lambda_i L_c$$

$$\text{Allocation} = \text{Mean} + z\sqrt{\text{Var}}$$

where $$\text{Var} = (F-M)\left(\frac{\lambda_i}{\lambda_T}\right)^2 + (F-M)\left(\frac{\lambda_i}{\lambda_T}\right) + L_c\lambda_i$$

If we put $r_i = \lambda_i/\lambda_T$, $R = F-M$, $h_i = Rr_i + \lambda_i L_c$

$$\text{Allocation} = h_i + z\sqrt{\{Rr_i^2 + h_i\}}$$

It is clear that on arrival of the procurement in the central store, whatever the employed z value, the allocation should be restricted to the value of $(\lambda_i/\lambda_T)F$. Otherwise, for those stores not being replenished at this particular instant there will be no stock available to ship to them when they are short of stock (as they generally will be) in this cycle.

16.3.6 Reorder quantity for sub-stores

Case Two: Procurement on order

Again, we must ration sub-stores. The cost function for a ration quantity B if $T = T_o$ is as before:-

$$C(B) = \sum_{d=0}^{d_o} p(d)\{\bar{C}_s(d,B) + k(d,B) \times c_R\}$$

To compute both k(d,B), (i.e. the expected number of extra orders) and the amount b_i for each of these orders for a given d, the following procedure is adopted. The ε-values for all those notional stock levels for which $\varepsilon \lesssim L$ are first established from the subprogram "CAIEX" and used to estimate the future reorder level vs. T function. The procedure follows on as before. The flow diagram for the computation of K(d,B) and Σb_i is given for the subprogram "CALBX(D)" in Fig.17.

To obtain p(d) some forecast of the future distribution of demands has to be made. It is assumed here that from analysis of past demands, a reasonable approximation will be the Poisson distribution, hence p(d) is given as follows:-

$$p(d) = e^{-\lambda_i(T_o-1)} \frac{\{\lambda_i(T_o-1)\}^d}{d!}$$

B is taken as that value minimizing the C(B) function. Convexity of C(B) is assumed.* $\bar{C}_s(d,B)$ is computed as for "Share Mk.VA" (see 15.4).

This 'Ration Rule' is known as "Share Mk VIA".

16.3.7 Reorder quantity for the complex

As before, in previous models of the complex this is taken as $\sqrt{2\lambda_T\{c_P + c_R \times \varepsilon(N_R) + \varepsilon(C_{RO})\}/hP}$ where $\varepsilon(N_R)$, $\varepsilon(C_{RO})$ are estimates.

---

* and seen to be true in practice.

16.3.8 Should sub-stores necessarily be replenished immediately following the procurement arrival in the complex?

The better policy in this respect is assumed to be identical to that for the control of previous models, viz., "Not necessarily replenish" policy.

16.3.9 Reorder level of the complex

As before, we would like to make this a control parameter. However, since the forecast demand will vary (and the actual mean demand may itself vary) the reorder level of the complex "M" would be required to be different at various instants of time.

To overcome this problem, we note M to be closely related to buffer stock held in the complex. For any given control, we want to employ a certain buffer stock so that there is a given (small) probability of the demand in the combined lead time L_c exceeding this buffer. If this probability is ~~the~~ related to a parameter "y", we ~~have~~ may write, since mean demand in time L_c is $\lambda_T L_c$;

$$M = \lambda_T L_c + y \times \sqrt{\text{Variance of Demand in time } L_c}$$

$$\text{i.e.} \quad M = \eta + y\sqrt{\eta} \qquad \text{where } \eta = \lambda_T L_c$$

16.4 Summary of Experiment Twelve

4 Year Simulation of Model VI with Control 11 Using IBM 1130 System with "Simon" Simulation Language

Full Description of Model given in 16.2.

Control Decisions

Central Store Reorder Quantity equals 60.

Control 11 is defined thus:-

Case 1: Procurement on Order: Ration Rule Share Mk VIA (see 16.3.6.)
Case 2: Procurement not on Order: "Allocation Rule"

Criterion of Reorder Level for Complex: "Free Stock"

Reorder Level for Sub-stores:

Case 1: Procurement on Order: Function of "T" from 3rd d.p. Model
Case 2: Procurement not on Order: Made equal to reorder level for instant T = L from 3rd d.p. Model.

16.4.1 Results of Experiment 12

In the following results the breakdown of the costs are given for three values of α, the response factor in the exponential smoothing forecast routine.* The results are illustrated graphically by the equi-total cost curves for the three α-values in Figs. 29, 30, 31.

y	$\bar{N}_P$	$\bar{N}_R$	$\bar{C}_{R_0}$	H.C.	T.C.	
-0.5	3	15	2.6	22.95	31.55	Case
0.0	3.5	17.5	2.4	20.73	30.13	α = .80
0.5	3.5	22	2.2	20.95	31.50	z = 0.0
1.0	3	23	2.2	21.60	32.20	
1.5	3	23	1.4	22.25	32.05	
-0.5	3	15	1.6	22.65	30.25	Case
0.0	3	16	2.2	21.01	29.51	α = .80
0.5	3	16.5	2.0	22.04	30.49	z = 0.3
1.0	3	15.5	2.4	23.41	32.01	
1.5	3	17.5	1.8	23.98	32.53	
-0.5	3	15	1.6	23.52	31.12	Case
0.0	3	15.5	2.6	21.63	30.38	α = .80
0.5	3	15.5	2.4	23.13	31.68	z = 0.6
1.0	3	15	2.2	24.24	32.34	
1.5	3	15.5	1.0	23.63	30.78	
2.0	3	19	1.4	24.50	33.10	
0.0	3	16.5	2.6	22.57	31.87	Case
0.5	3.5	16	1.8	24.08	32.43	α = .80
1.0	3	15	1.8	25.02	32.92	z = +0.8
1.5	3	13	1.0	23.93	30.33	
2.0	3	16.5	1.2	24.14	31.79	
1.0	3.5	13	1.4	26.75	33.80	Case
1.5	3	13.5	0.8	24.97	31.32	α = .80
2.0	3	14	0.8	25.27	31.78	z = +1.0
-1.0	3	22.5	4.4	16.00	28.65	Case
-0.5	3	26.5	3.0	15.74	28.21	α = .90
+0.0	3	33.5	2.4	16.12	30.07	z = -0.6
+0.5	3	38	1.6	16.75	31.25	

---

* The results give the average over two simulation runs (employing streams 1115, 1729).

y	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{R_0}$	H.C.	T.C.	
-0.5	3	19.5	3.0	17.76	28.11	Case
0.0	3	21	2.4	17.35	27.55	α = .90
0.5	3	24	2.2	17.64	28.54	z = -0.3
1.0	3	28.5	2.0	17.78	29.33	
-1.0	3	15	3.4	20.93	30.33	Case
-0.5	3	17.5	3.2	18.63	28.58	α = .90
0.0	3	17.5	2.6	19.49	28.84	z = 0.0
0.5	3	19.5	2.4	19.44	29.19	
1.0	3	21	1.8	20.29	29.89	
+0.0	3	16.5	3.0	19.67	29.12	Case
0.5	3	16.5	2.4	20.14	28.99	α = .90
1.0	3	14	3.2	24.23	33.13	z = +0.3
-0.5	3	11.5	2.4	22.76	30.11	Case
+0.0	3	15	2.4	21.87	30.27	α = .90
0.5	3	16.5	2.4	20.14	28.99	z = +0.6
1.0	3	12.5	1.8	23.97	31.02	
-1.0	3	17.5	5.8	17.61	30.16	Case
-0.5	3	18.5	6.0	17.25	30.30	α = .95
0.0	3	22	5.4	17.29	30.80	z = -0.3
-1.5	3	15.5	5.8	19.06	31.01	Case
-1.0	3	16	5.4	18.70	30.55	α = .95
-0.5	3	16	5.6	18.92	30.82	z = 0.0
+0.5	3	19.5	4.6	18.96	30.91	
-1.0	3	14.5	6.0	19.20	31.05	Case
0.0	3	16	4.8	19.50	30.60	α = .95
0.5	3	19	3.4	19.39	30.49	z = +0.3
1.0	3	20	2.8	20.43	30.73	
1.5	3	21	2.4	21.16	31.37	
-1.0	3	14	5.0	19.84	30.54	Case
-0.5	3	13.5	4.8	19.90	30.25	α = .95
+0.0	3	14	4.6	20.42	30.72	z = +0.6
0.5	3	16	3.8	19.62	29.70	
1.0	3	16.5	4.2	20.86	31.51	

y	$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{R_0}$	H.C.	T.C.	
-0.5	3.5	14	4.0	20.63	31.58	Case
+0.0	3	15.5	4.6	20.60	31.35	α = .95
0.5	3	14.5	4.0	21.40	31.25	z = +0.9
1.0	3	17	3.8	21.63	32.03	
1.5	3	15	3.4	21.51	30.91	
1.0	3	15.5	4.0	21.58	31.73	Case
1.5	3	15.5	4.0	22.84	32.99	α = .95
2.0	3	16.5	3.0	23.02	32.47	z = +1.2

16.5 Details of Program 3

This describes the work for Experiment 12. The Third Dynamic Programming Model is given by Program "NNNY" and subprograms "SUBY", "CAL" (see Flow Diagram in Fig.22). This is computed off-line and the results are used in the main simulation prgram "MMMX" (see Fig. 23 for flow chart) and its subprograms.

16.5.1 Subprograms called from the Main Simulation Program "MMMX"

(i) "CAIEX"

This subprogram computes for sub-stores the value "ε" for the relevant demand rate and notional stock level by linear interpolation between those ε values for the nearest defined λ. The flow chart is given by Fig. 24.

(ii) "CALCX (ISDA)"

This subprogram is analogous to "CALCS" in Program 2. It computes the expected cost of shortages in the coverage time as a result of a possible ration quantity. The flow diagram for this calculation of $c_s'\phi(\overline{S},\overline{L},\overline{\lambda})$ is given by Fig.25.

(iii) "CALGX"

This subprogram computes G and is described by the flow chart in Figure 16.

(iv) "CALBX(D)"

This is analogous to "CALKB(D)" of Program 2 and its flow diagram is given by Fig. 17.

(v) "GENX"

This subprogram is identical to "GEN" and contains the instructions

FIG.22 FLOW DIAGRAM FOR OFF-LINE COMPUTATION OF THE 3rd DYNAMIC PROGRAMMING MODEL: "NNNY/SUBY" EXPERIMENT 12

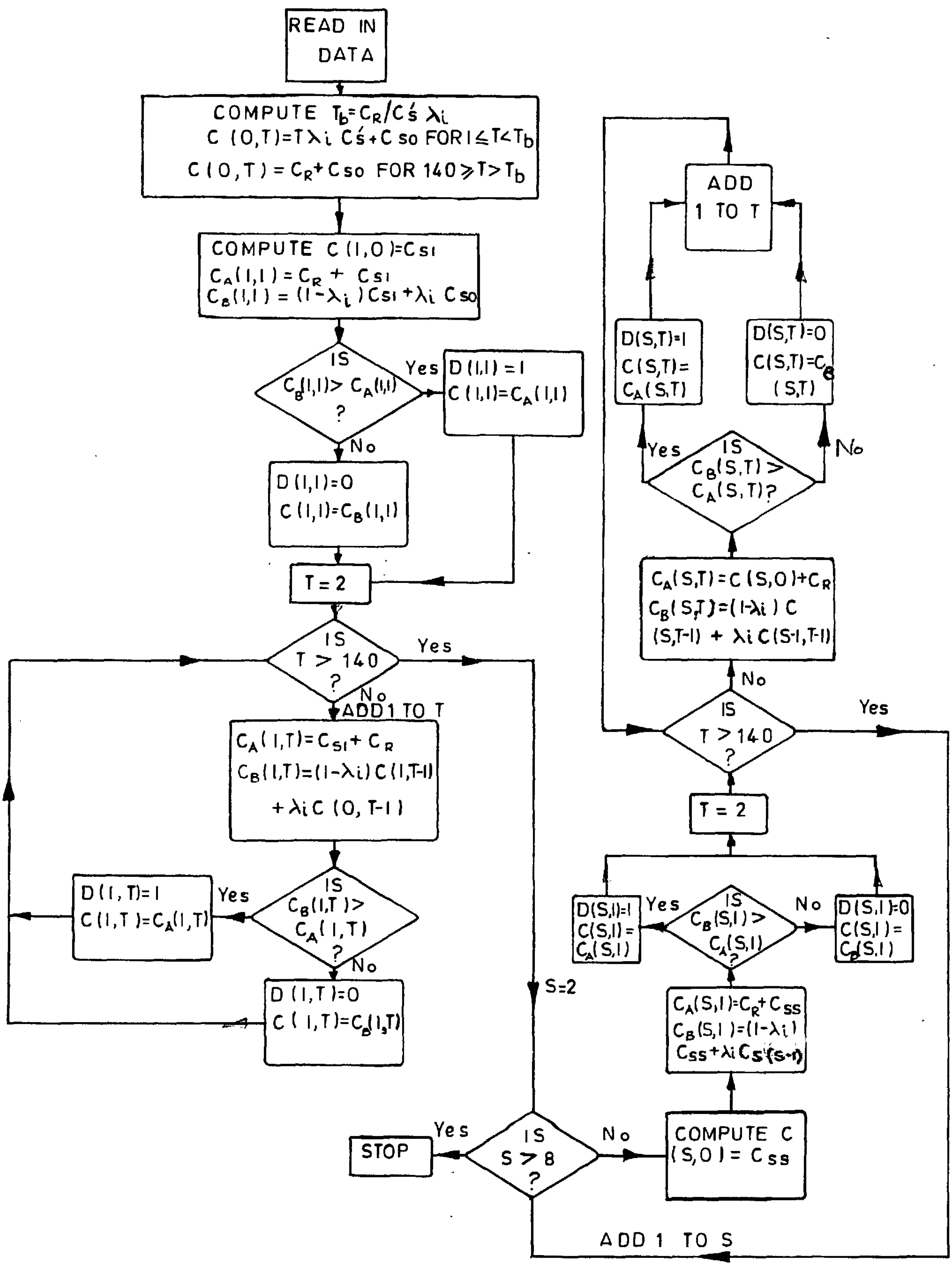


FIG. 23 FLOW CHART FOR SIMULATION IN EXPERIMENT 12

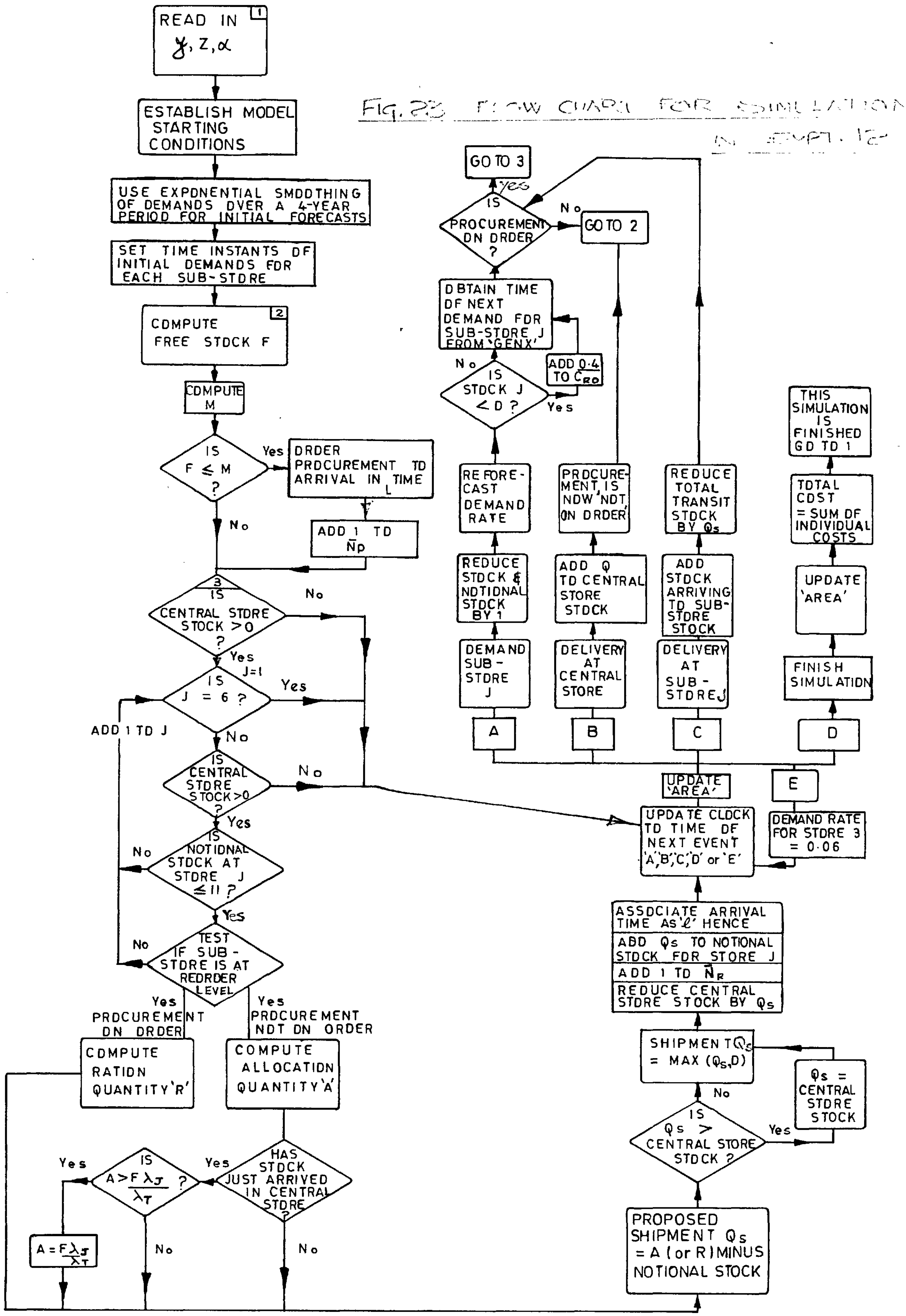


FIG. 24 FLOW CHART FOR COMPUTATION OF THE VALUE ϵ IN EXPERIMENT 12 SUBPROGRAM 'CAIEX'

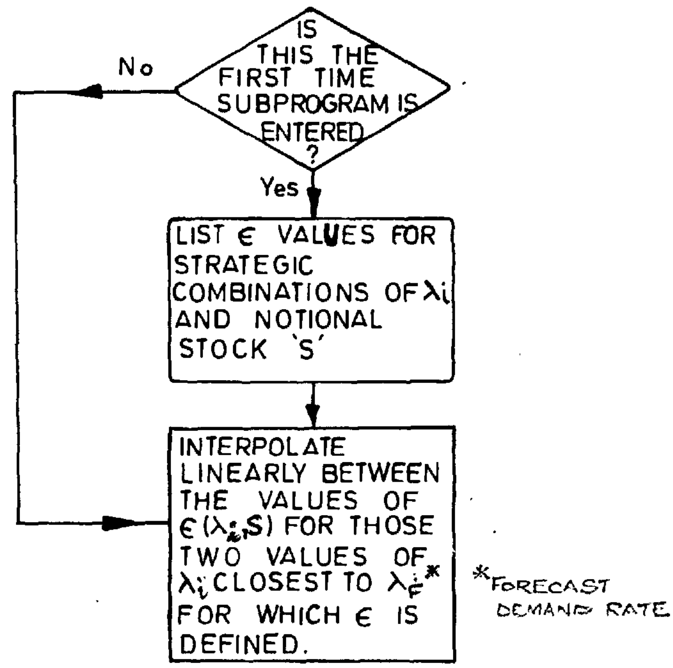
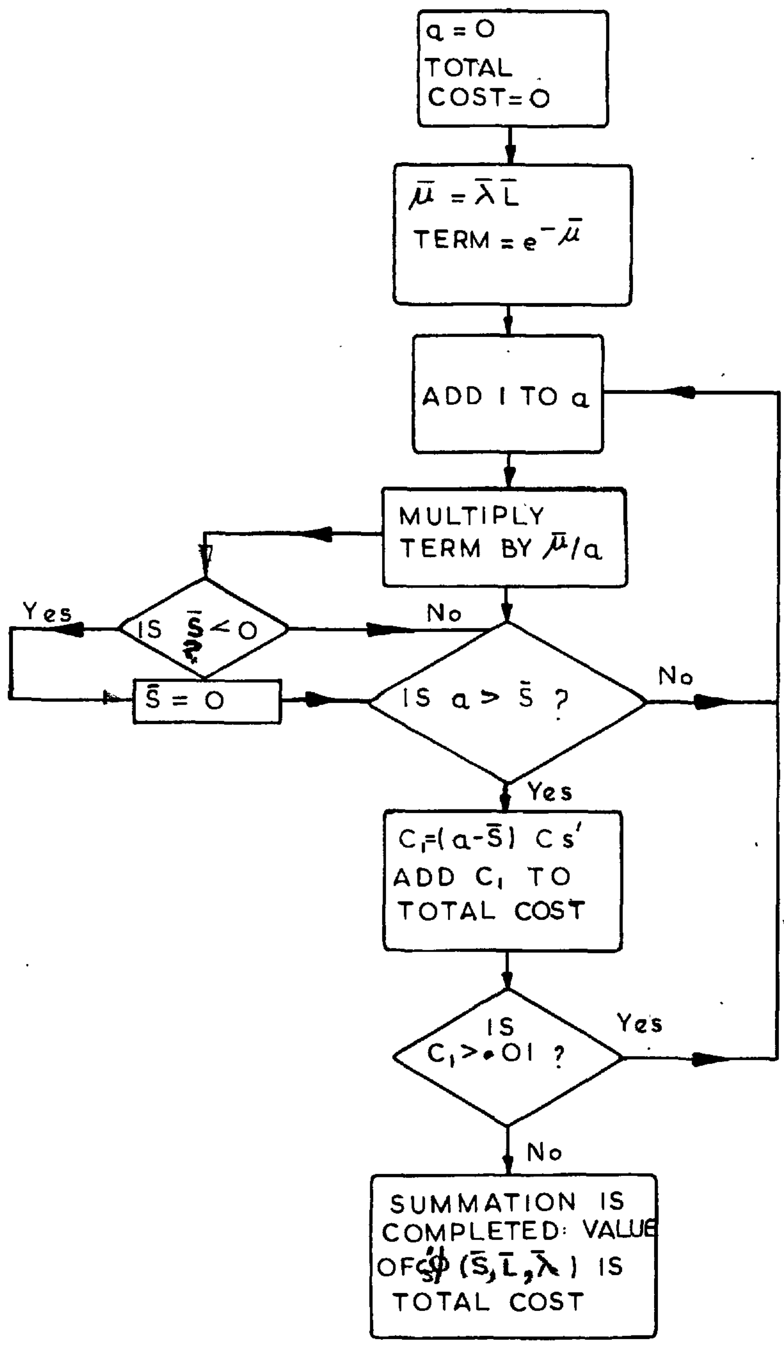


FIG. 25 FLOW DIAGRAM FOR COMPUTATION OF INDIVIDUAL SHORTAGE COST $C'_s \phi(\bar{S}, \bar{L}, \bar{\lambda})$ IN EXPERIMENT 12 "CALCX"(ISDA)



THIS ROUTINE PERFORMS THE SUMMATION:

$$Cs' \sum_{i=\bar{S}+1}^{\infty} (i-\bar{S})\, p(i) \text{ for } \bar{S} \geq 0 \text{ where } p(i) = e^{-\bar{\lambda}\bar{L}} \frac{(\bar{\lambda}\bar{L})^i}{i!}$$

FIG. 26 FLOW CHART FOR COMPUTATION OF FREE STOCK FOR EXPERIMENT 12 SUB PROGRAM "SFRX"

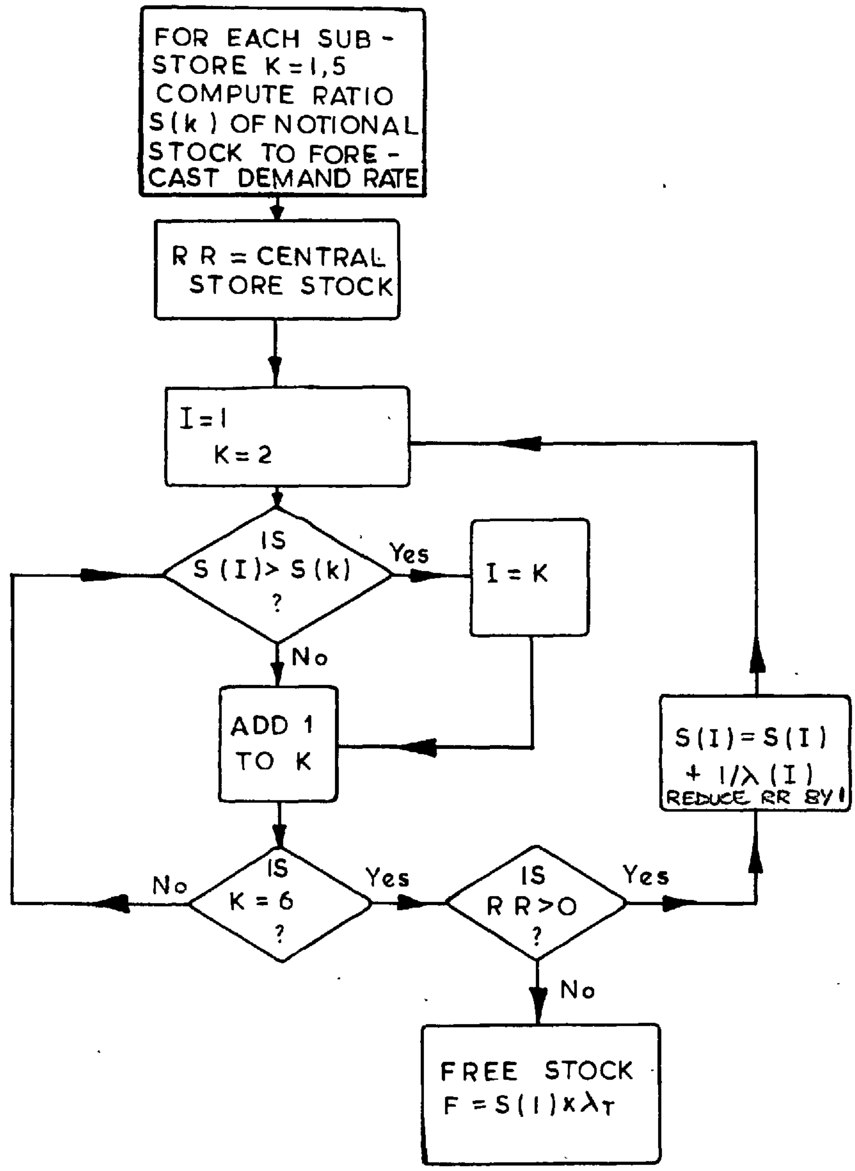


FIG. 27

SUB-STORE REORDER LEVEL FOR THE CASE WHEN AN ORDER FOR A PROCUREMENT IS NOT OUTSTANDING PLOTTED AGAINST FORECAST DEMAND RATE (USES THE 1st D.P. MODEL RESULTS FOR MODEL VI ASSUMING BACKORDER COST TO BE TIME DEPENDENT)

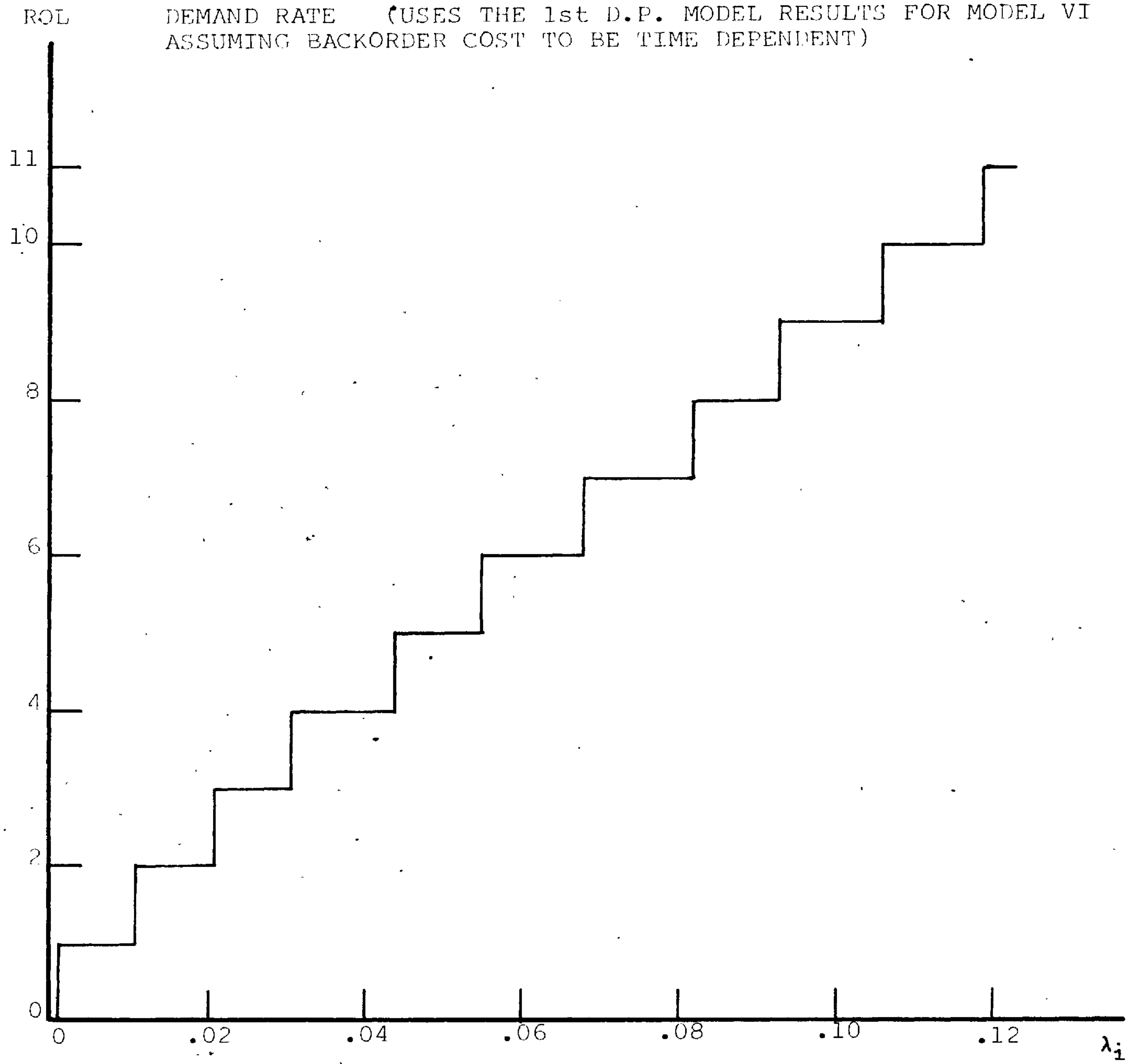


FIG. 28

SUB-STORE REORDER LEVEL WHILST A PROCUREMENT IS NOT OUTSTANDING VERSUS FORECAST DEMAND RATE FOR EXPERIMENT 12: (USES THE 3rd D.P. MODEL RESULTS FOR MODEL VI WITH SHORTAGE COSTS TIME-INDEPENDENT)

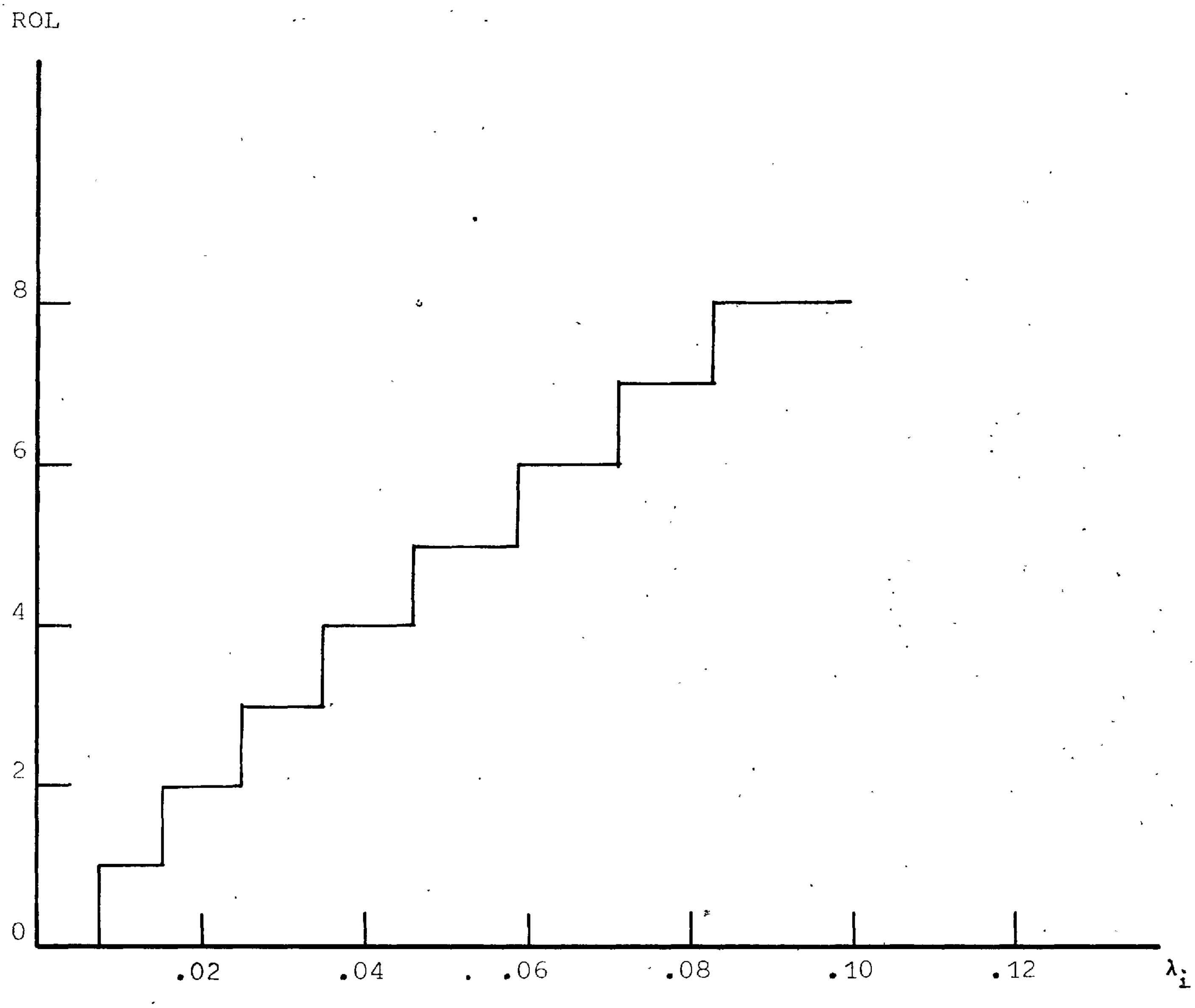
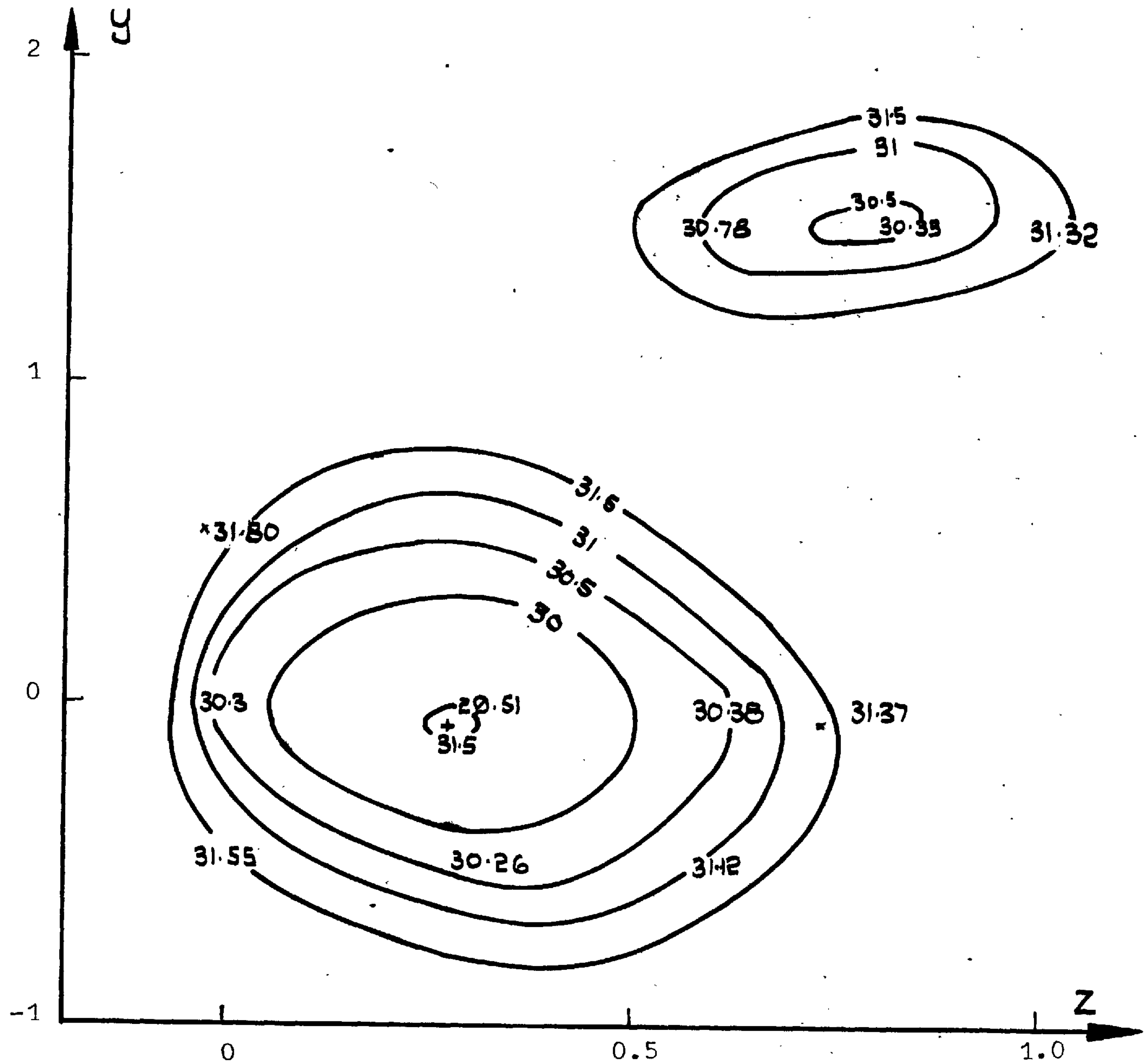


FIG. 29

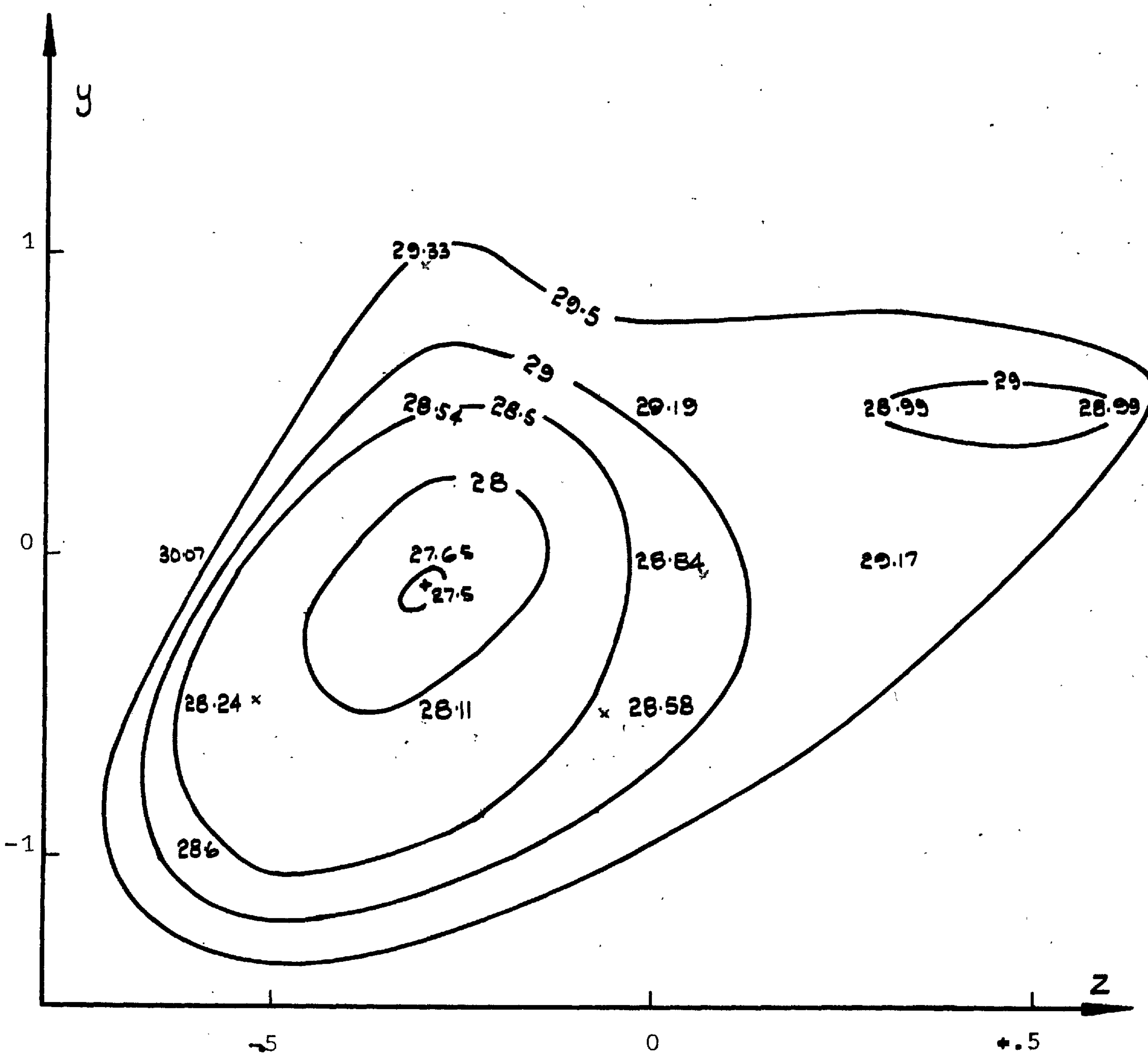
EQUI-COST CURVES FOR CONTROL APPLIED TO MODEL VI
IN EXPT. 12 (CASE $\alpha = 0.8$)



ESTIMATED MIN COST = 29.5

FIG. 30

EQUI-COST CURVES FOR CONTROL 11 APPLIED TO MODEL VI
IN EXPT. No. 12 (CASE $\alpha = 0.9$)

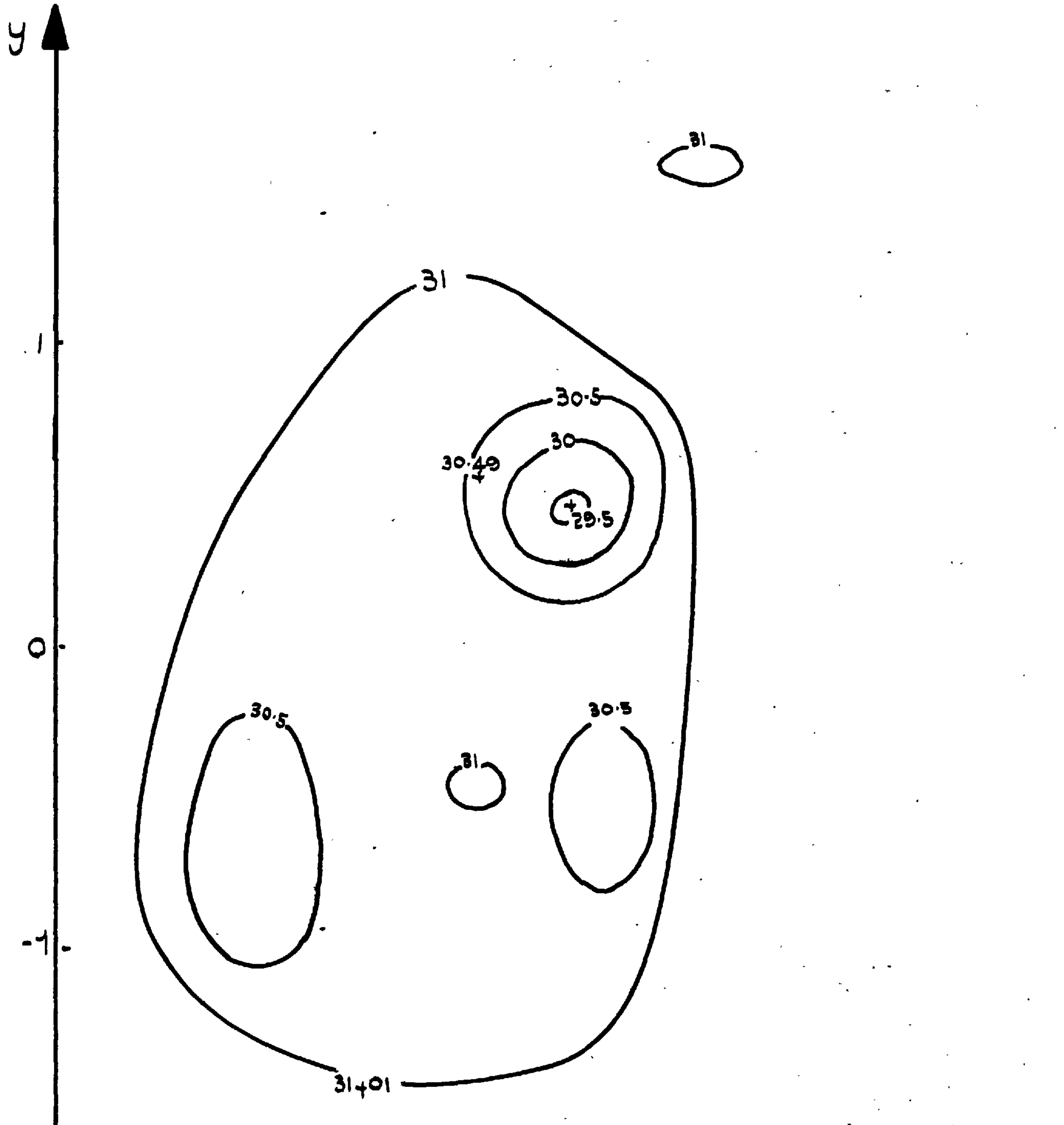


ESTIMATED MIN COST = 27.5

252

FIG. 31

EQUI-COST CURVES FOR CONTROL APPLIED TO MODEL VI
IN EXPT. No. 12 (CASE : α = 0.95)



ESTIMATED MIN COST = 29.5

for establishing the time of the next demand for a sub-store which has just experienced a demand.

(vi) "PNOX"

This is the computation of the allocation quantity, fully described in 16.3.5.

(vii) "POOX"

This is the computation of the Ration Quantity analogous to "POO1" for Experiment 11 (Program 2) and is given as a flow chart in Fig.18.

(viii) "SFRX"

This is the computation of "Free Stock" F. The flow chart is given by Fig. 26.

(ix) "SSRX"

This is analogous to "SSRO" in Program 2 and tests whether any sub-store is at its reorder level or not.

(x) "SYRX"

This is analogous to "SYRO" in Program 2. It computes M from the expression in 16.3.9 and tests whether Free Stock < M. If so, a procurement order is initiated.

(xi) "PLAN, ADDL, BEHE, GROU, HEAD, MEMN, REFN, SCA, QUEUE, SETT, SIZE TIMV, SIMO, DELE, ENTI" are the special-purpose subroutines of the "Simon" Simulation procedure.

16.6 Comments on the Results of Experiment 12

Cost of computer simulation prohibits simulation with further random number generating streams. The two employed for this experiment produce meaningful equi-cost curves and from these one can see that the $\alpha = 0.9$ parameter is able to produce control for a wide area of (y,z) combinations for which total cost is less than the best achieved for both the $\alpha = 0.8$ and $\alpha = 0.95$ cases (see contour lines for Total Cost = 29.5 in Figs. 29, 30, 31).

Good estimates of the best parameter combinations and their expected total costs are obtained from these equi-cost graphs. These are

$$\alpha = 0.8 \quad \begin{pmatrix} y = -0.1 \\ z = +0.3 \end{pmatrix} \quad \text{Cost} = 29.5$$

$$\alpha = 0.9 \quad \begin{pmatrix} y = -0.1 \\ z = -0.3 \end{pmatrix} \quad \text{Cost} = 27.5$$

$$\alpha = 0.95 \quad \begin{pmatrix} y = +0.4 \\ z = +0.5 \end{pmatrix} \quad \text{Cost} = 29.5$$

It is fair to generalise that the results indicate an optimal result in the region of zero y and zero z. This may appear somewhat surprising since, on the surface, zero y seems to mean very small buffer and zero z seems to indicate that replenishment costs will be high.

16.6.1 Implication of the results that the optimal is for zero y, zero z

Let us refer to the least-cost result recorded from the simulation runs. This occurred at $\alpha = 0.9$, $z = -0.3$, $y = \text{zero}$. Raising y for this z reduces the cost of shortage but not to a significant extent. However it clearly results in an increased overall replenishment cost. Lowering y means more shortage and more holding costs.

Keeping y = zero, and raising z reduces the number of replenishments but results in extra holding costs and (to a lesser extent) extra shortage cost.

The result that (for the least-cost case) y is zero, does not really mean that there is little buffer stock; it merely means that the control is responding to the low-level (relative to expected demand) of stock which has occurred in one particular sub-store when initiating the procurement order. Exhibit "D", which records the simulation details for the least cost result admirably demonstrates this point. At a clock time of 798, the third request for stock for the complex is made. Since $y = 0$, $M = \lambda_T L_C$ and since forecast $\lambda_T = .172$, $M = .172 \times 125 = 21.5$. The sub-store stocks (notional) are:-

2, 5, 7, 11, 11.

Respective forecast demand rates λ_i are:-

.010, .019, .064, .034, .044.

The minimum value of notional stock/λ_i is for the 3rd store, 7/.064, whence the hypothetical "equalising distribution" from the central store results in:- the new minimum value of 8/.064, with zero stock at the central store. As usual, the value of F is computed as this value, 8/.064, times λ_T; i.e. $8/.064 \times .172 = 21.5$, whence F = M and the procurement order is initiated. At the least-cost result, the average buffer stock maintained is 13.4. Attempts to reduce this buffer with this control must be made by keeping z down - whence the expected optimum result is for a low z, viz. z = 0.

16.6.2 Consideration of the parameter α

Clearly, making the value of α very low has the undesirable effect of making the forecast demand rates too responsive to the random nature of the individual times between demands. Too high a α (close to unity) means that the control takes an excessively long time to respond to the type of jump in demand that is experienced in the workings of Model VI. The expected best choice for α can be seen (from a cubic approximation from the minimum cost estimates for those values of α employed in Experiment 12) to lie in the region of 0.88, for which an estimate of annual total cost of operation is 6.75 cost units.

16.7 Conclusions from Experiment 12

The ideas of modification to the concepts of the earlier chapters for their application to a model of the complex with demand rates which are forecast and different over the sub-stores produce sensible results. The Exhibit "D" can be used to check the details of the control for a close-to-optimal parameter combination; the ration rule can be seen to decide on the "right sort" of shipment quantity. Further, the idea to make the reorder level for sub-stores in the period when the procurement is not on order equal to that at the time instant when the order is first made is clearly an important contribution. It results in delaying the decision to replenish sub-stores until about the right sort of time. Replenishing earlier (at a higher notional stock level) is likely to lead to both higher cost of Maldistribution Type 2 and extra stock-holding costs (the latter is particularly variable and poor distribution of stock over the sub-stores can result in high maldistribution cost, type 1) whilst resulting in minimal savings of Maldistribution Type 3.

16.8 Experiment 13

This experiment represents the simulation of the previous model for a period of 8 years (2000 days) but here the jump in demand for sub-store No.3 (from a mean rate 0.03 to 0.06) occurs at the half-time instant. Two simulations are run. The first of these uses the stream 1111 for the occurrence of demands in the simulation with stream 1113 for the occurrence of demands to obtain the initial forecast. The second uses streams 1115, 1117 respectively for these two purposes. As with the previous experiment, the initial forecast represents that resulting from the consideration of four years' demands. The parameters ($y = -.1$, $z = .3$, $\alpha = .9$), which are the optima from Experiment 12, are employed.

16.9 Results of Experiment 13

Figure 32 represents the forecast demand rate for sub-store No.3 when plotted against time. Fluctuation about the mean AB is noted for the period in time 0 → 1000 days and the lag in time for the forecast to reach the mean CD is noted from 1000 days onwards.

Figure 33 shows how the virtual stock for sub-store No.3 varies through time. The occurrence for three backorders at time 1130 is to be noted in one of the simulations.

16.10 Conclusions from Experiment 13

The response and stability of the control rules may be judged from the Figures 32 and 33. The forecast demand rate reacts fairly quickly to the demand jump and there is no disastrous onset of backup at the critical period (just after time equals 1000 days).

16.11 Experiment 14

This experiment represents the simulation of Model VI with those parameters which yielded the least-cost result in Experiment 12.* Here the demand jump is not applied. It is a lengthy simulation over 20 years and is primarily designed to show the sort of variation in stocks and number of replenishments per procurement that is likely as a result of the control.

A cycle is defined as the time between procurement arrivals. The cycle for the consideration of maximum and minimum virtual stock at sub-stores is advanced by the sub-store lead time on the defined cycle. That is to say the cycle for this case begins time ℓ after procurement arrival (when sub-store stocks are critically low) and finishes time ℓ after the next procurement arrival.

16.12 The Graphs for Experiment 14

Figure 34 records for the thirteen different cycles System Stock at interesting instants in time, viz.

---

* (viz. $y = -.1$, $z = -.3$, $\alpha = .9$)

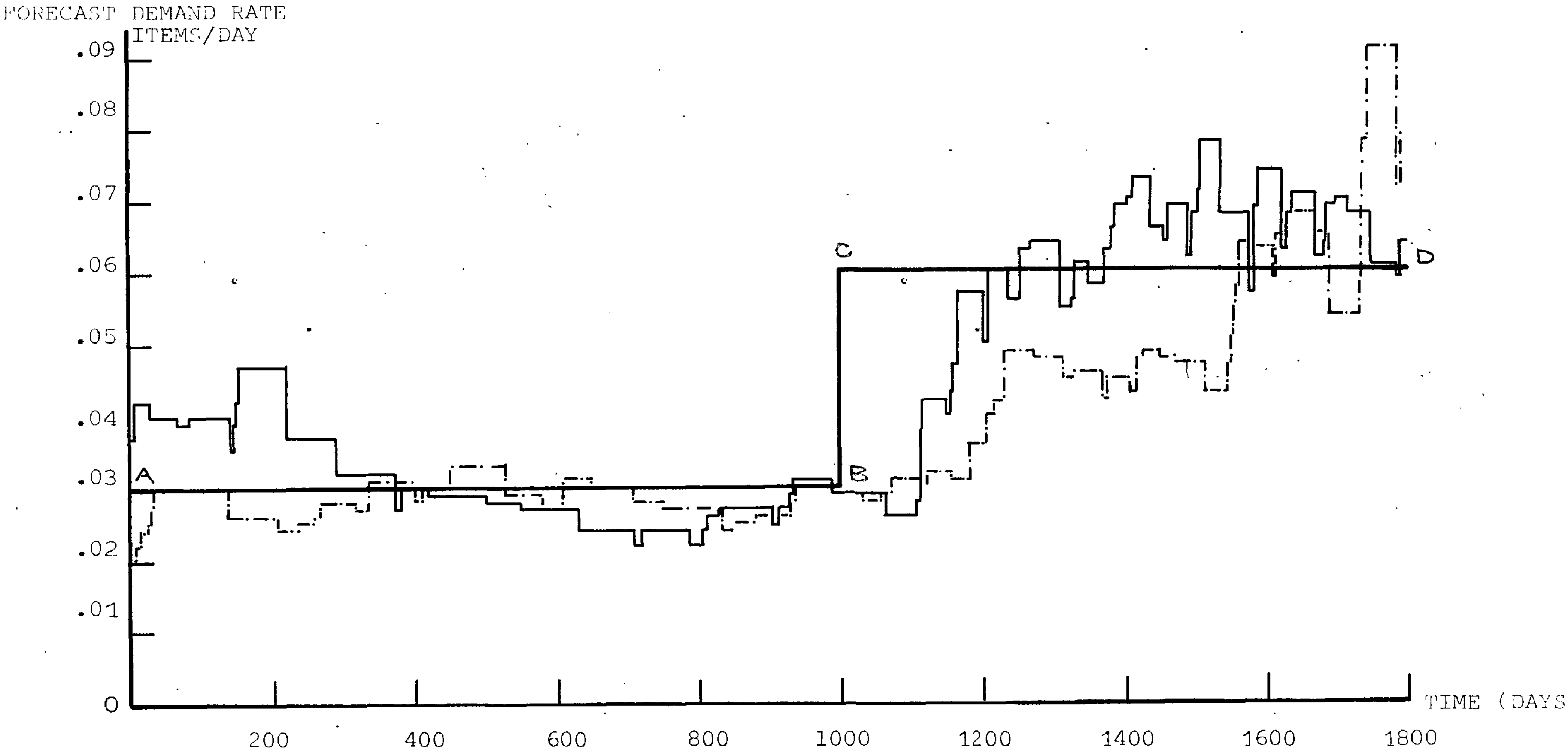


FIG. 32 (EXPERIMENT 13)

FORECAST DEMAND RATE THROUGH TIME FOR SUB-STORE NO. 3 WHEN ACTUAL MEAN-DEMAND RATE IS GIVEN BY ABCD

(CASE y = -0.1
z = -0.3
c = 0.9)

LEGEND: -.-.-.-.- USING STREAMS 1111 FOR SIMULATION, 1113 FOR INITIAL FORECAST
———— USING STREAMS 1115 FOR SIMULATION, 1117 FOR INITIAL FORECAST

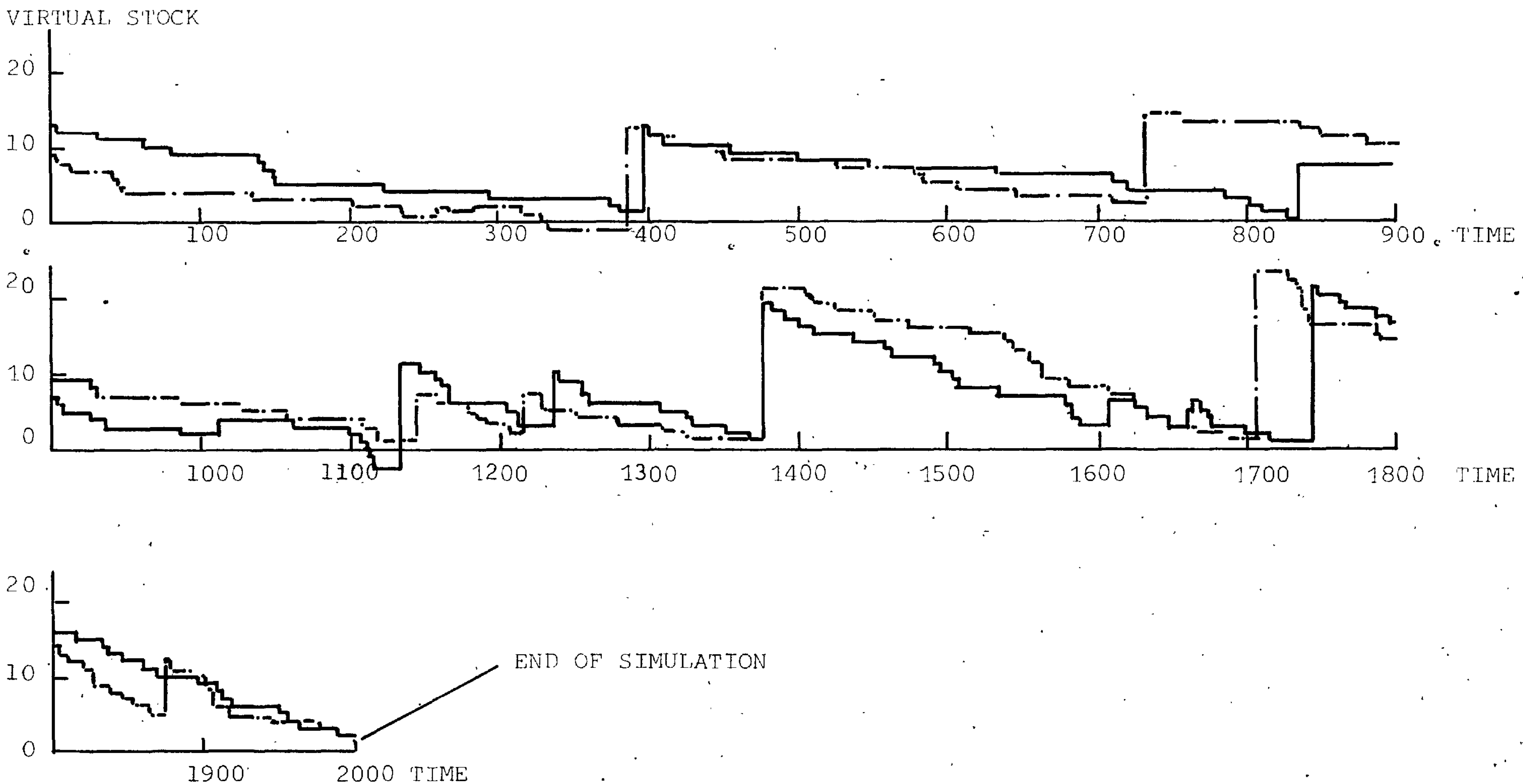


FIG. 33 (Expt. No. 13)

VIRTUAL SUB-STORE STOCK FOR SUB-STORE No. 3 (AT WHICH JUMP FROM MEAN POISSON DEMAND .03 TO .06/day IS APPLIED AT TIME 1000) PLOTTED AGAINST TIME (CASE $y = -0.1$: $Z = -0.3$: $\alpha = 0.9$)

LEGEND -.-.-.- USES STREAMS 1111 FOR SIMULATION, 1113 FOR INITIAL FORECAST

——— USES STREAMS 1115 FOR SIMULATION, 1117 FOR INITIAL FORECAST

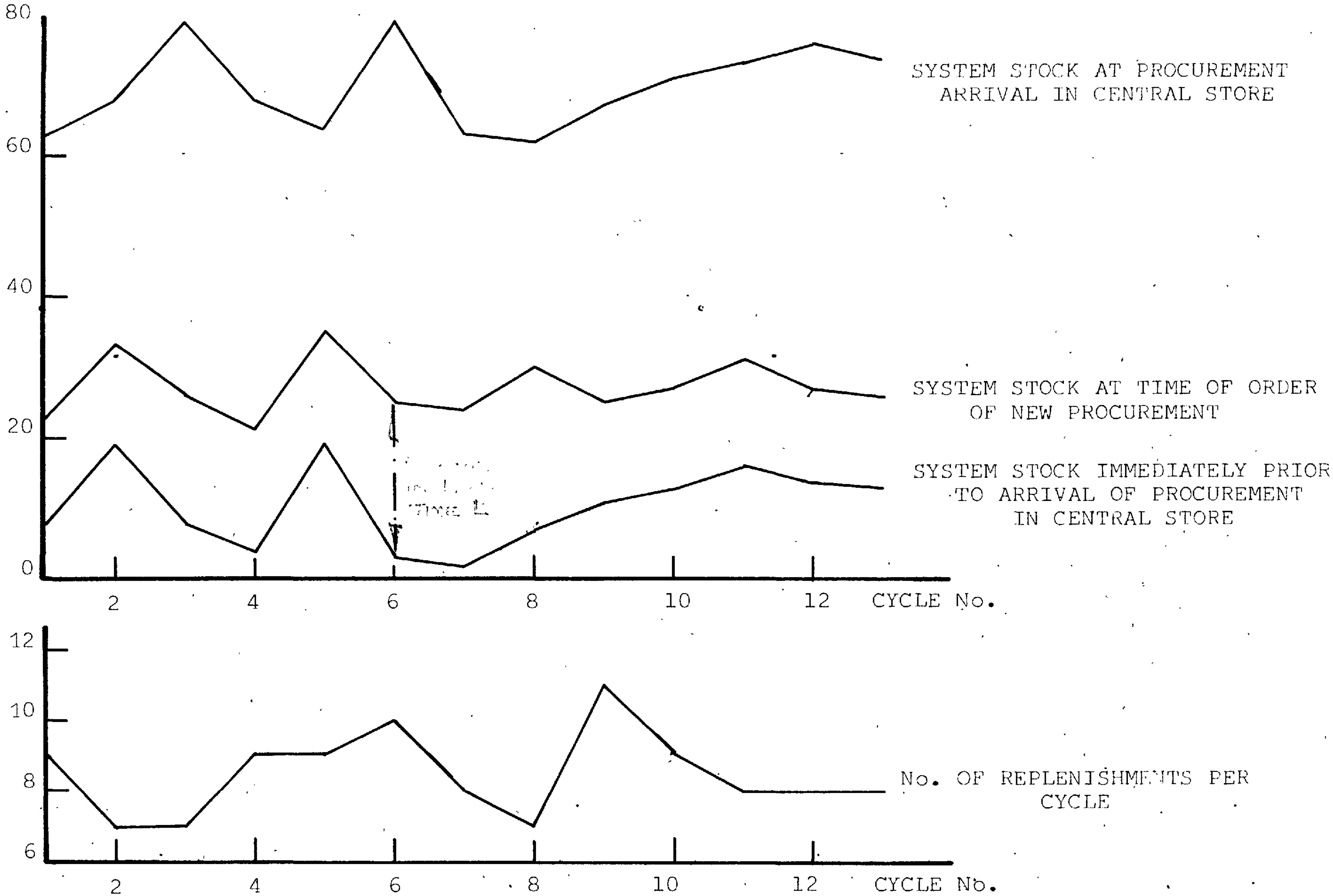


FIG. 34 (Expt. No. 14)

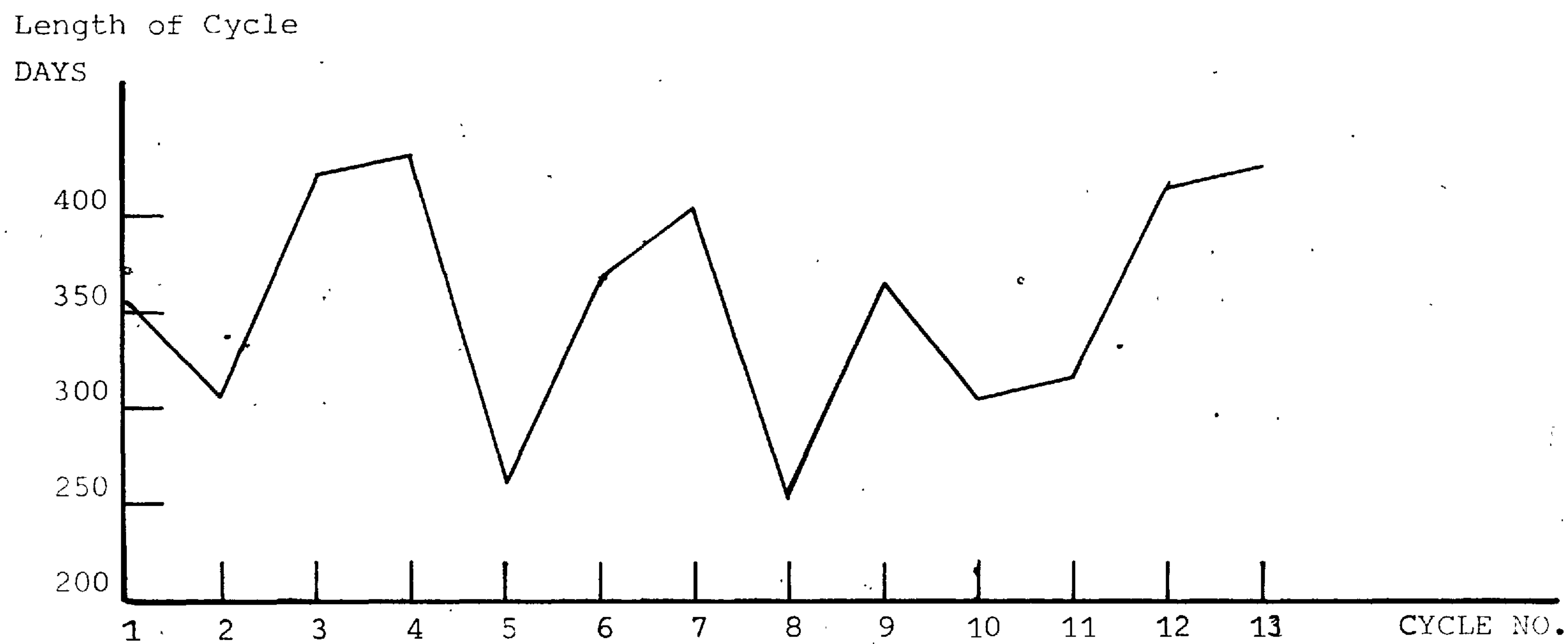


FIG. 35 VARIATION IN CYCLE LENGTH (Expt. No. 14)

FIG. 36. (Expt. No. 14)

MAXIMUM AND MINIMUM VIRTUAL STOCK LEVELS FOR THE VARIOUS CYCLES FOR SUB-STORES NO. 1 to 3 INCLUSIVE

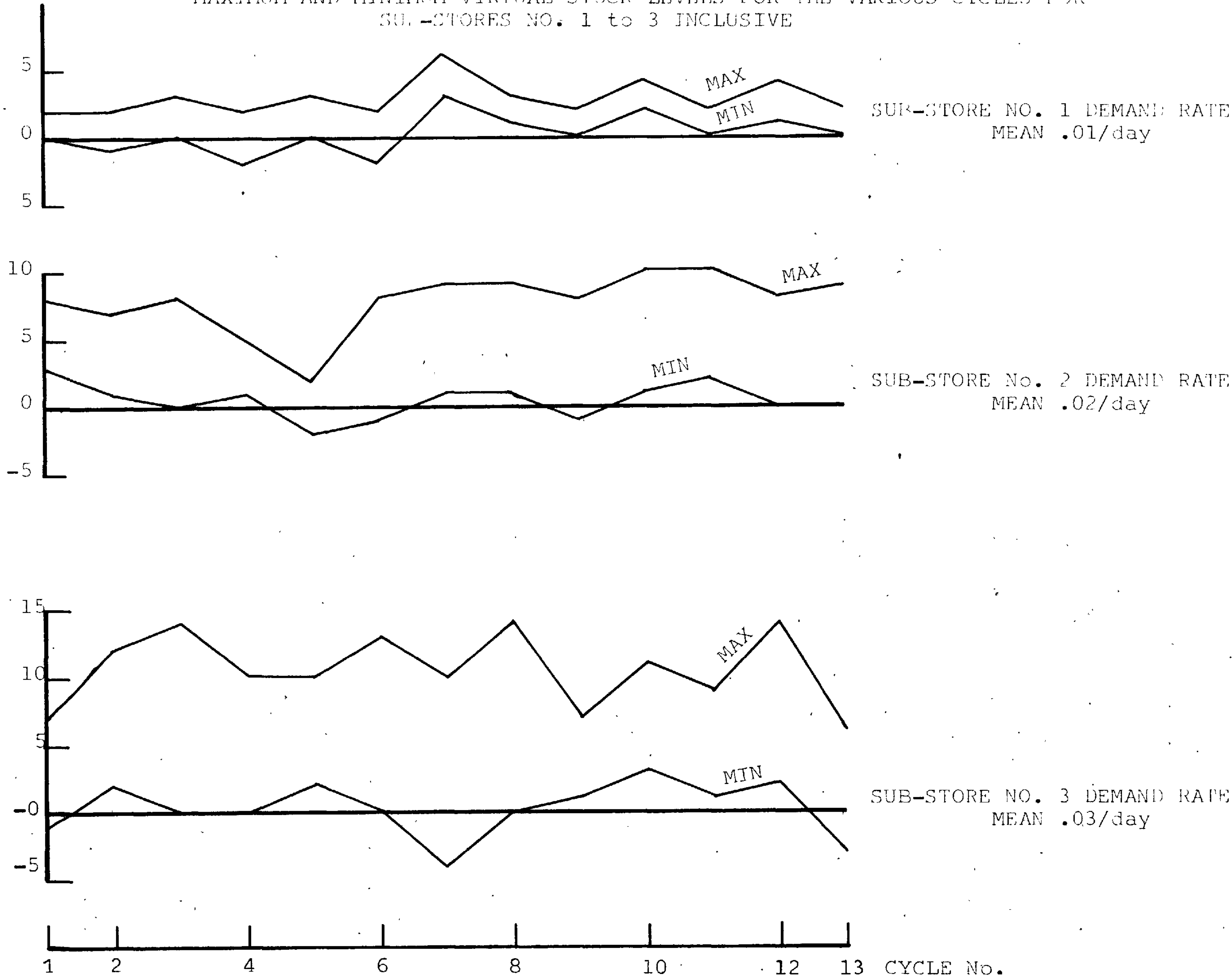
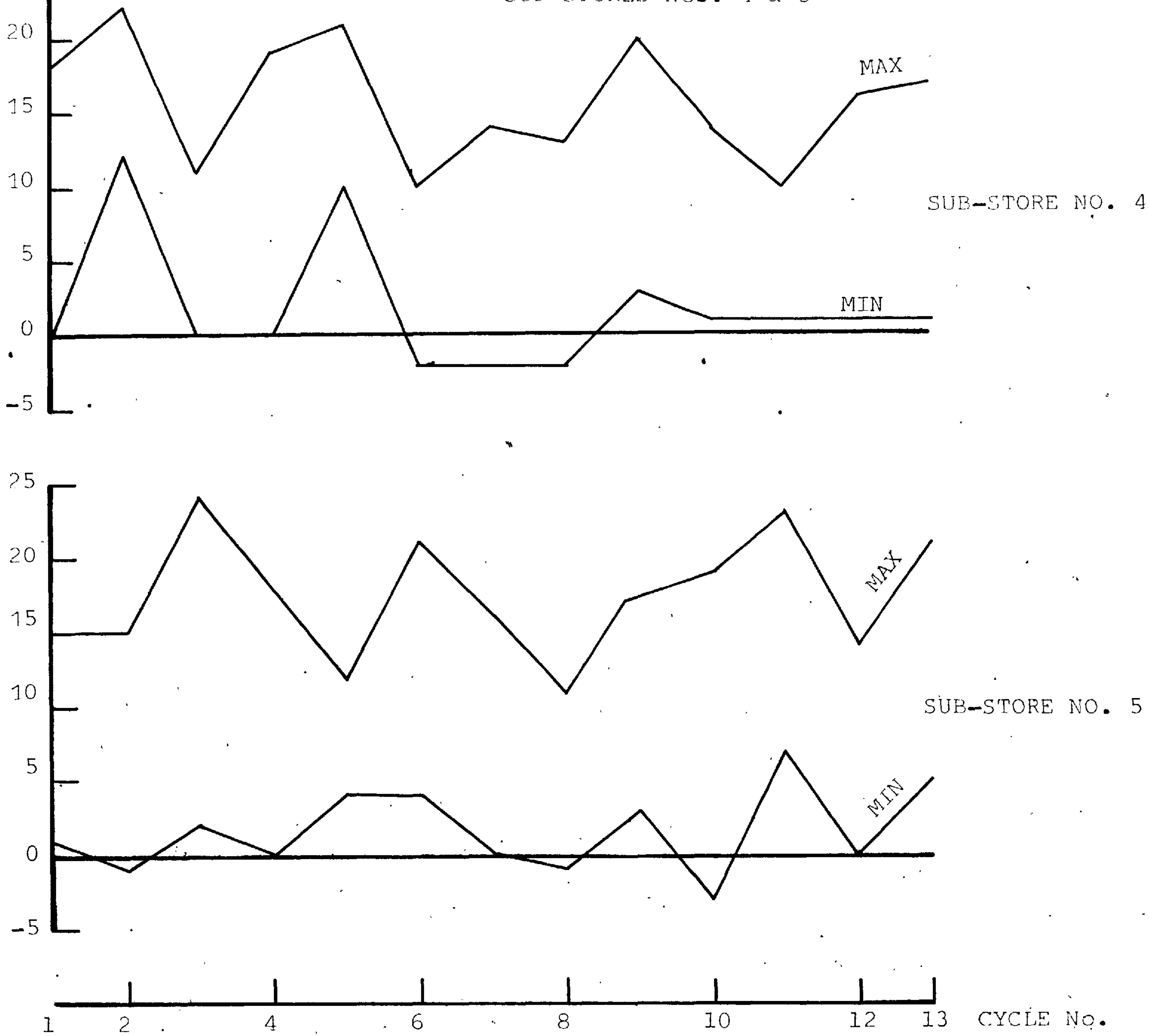


FIG. 37 (Expt. No. 14)

MAXIMUM AND MINIMUM VIRTUAL STOCK LEVELS FOR THE VARIOUS CYCLES FOR SUB-STORES Nos. 4 & 5



(i) time of procurement in central store

(ii) time of order of new procurement

(iii) immediately prior to arrival of new procurement.

It also graphs the number of replenishments for each cycle.

Figure 35 records the length of each cycle.

Figures 36, 37 record the maximum and minimum virtual stock levels for the five sub-stores over each cycle.

16.13 Summary of Chapter Sixteen

This chapter has considered the extension of the control rules as proposed in this thesis fo a model in which demands at sub-stores are different and are forecast. A further innovation is to introduce a sudden change in demand at one sub-store. The response to this may be altered by changing the parameter 'α' in the exponential forecast procedure. Equi-cost graphs for three different values of α are given and a detailed simulation for a close-to-optimal parameter combination is shown in Exhibit "D".

The variation in the forecast demand rate and the virtual stock level for the sub-store experiencing the demand jump are illustrated graphically. Also represented graphically for a similar model of the complex in which no demand jump is experienced are the variation in stocks and number of replenishments per procurement over a lengthy period of time.

CHAPTER SEVENTEEN

AN ANALYTICAL APPROACH TO OBTAINING THE TRIGGER FOR PROCUREMENT ORDER AND COMPARISON OF PROPOSED CONTROL IDEAS WITH THE CONTROL OF CRAN

17.1 Introduction to Chapter Seventeen

In this chapter a comparison is made between the optimal solutions using the author's suggested control with the control that Cran would achieve when following the procedure[14] (involving the drawing of graphs and solving a series of equations by the suggested iterative method) to locate his optimal solution.

17.2 Method of Comparison

The comparison is to be made on a non-zero lead time model in which shortage costs are time-independent. This latter point is important since Cran's procedure is given only for the case of time-independent shortage costs.

There are two alternative comparisons which may be made:

(i) The author's control rules may be applied to the specific model considered by Cran and simulation performed to locate the expected optimal parameters. Use may then be made of Cran's graphs to locate the parameters which will yield his optimal solution. A comparison* is then possible by simulating the model (a) with Cran's control rules and his optimal parameters, and (b) with the author's control rules, using optimized parameters. The same random number streams should be used for (a) and (b).

(ii) The author's control rules may be applied to any suitable model for which Cran's control is applicable and simulation performed to locate the minimum cost solution. The graphs Cran requires which will permit the location of his optimal solution are obtainable by simulating the model and determining the expected number of replenishments (γ_R), and number of backorders (γ_S), occuring when System Stock is allowed to run right down to zero. The same number of simulator runs as suggested by Cran over which to average will be used to obtain the relevant graphs in this case. These graphs from the simulation will be used in the method indicated by Cran[14] to locate those parameters he claims will yield his optimal solution. This is followed by the comparison* in (i) above.

17.3 Factors Affecting Choice of a Model for Comparison

As has been said before when considering the model favouring Cran with zero lead time to sub-stores, Cran's control will work well when shortages and replenishments are not costly. This then includes those complexes for which it is correct to incur shortages. We would expect that for such complexes, the cost of holding stock is high compared with shortage cost. Clearly, low buffer must be maintained for sensible control. Whether Cran has holdback (i.e. whether his holdback factor exceeds zero) will depend on whether the savings in shortage costs as a result of holdback are worth the extra replenishment costs required. (This depends on the relation between the unit costs of each.) If Cran is striking a correct balance between inventory holding and shortage for his control, the author's control, which has been generally orientated towards shortage prevention (but flexible enough in the z-parameter to respond to cases where it is correct to have shortages) is expected to be little better than Cran's. The regions in which possible gains can be made are:-

(i) Ensuring that reorder for the complex occurs at stock configurations of comparable strategic value (unlike Cran's control with a fixed System Stock level where the procurement is likely to be trigger at dissimilar strategic values with regard to shortage).

(ii) Not necessarily replenishing sub-stores following procurement arrival in the central store. As a result of this, there is generally stock "held back" in the central store after procurement arrival. This has the advantage that for some period of time there will be stock available in the central store to buffer any needy sub-store; this benefit is "free", i.e. it is not necessary either to:

(a) incur more than one delivery to sub-stores per procurement on average;

or (b) keep stock in the central store held back at the critical time just prior to next procurement arrival.

If Cran has holdback, either (a) or (b) must occur, and so, in this respect, the author's control has advantage. However, when c_s'/c_R is sufficiently low, it is generally not worth a replenishment to save a possible shortage, and the benefit of a period of "free buffer" for sub-stores in this case does not exist since it is not generally sensible to utilize this stock in further shipments in the same cycle.

17.4 Model VII

The model for comparison will be designed to incorporate data features where Cran's control is expected to work well compared with the author's. Model VII, with the data for Model VI (without demand jump) fits this category. This model is realistic in that sub-stores have differing average demand rates (Poisson distributed) although the distribution is stationary. This latter point is assumed for control purposes, and the real average demand rate and Poisson nature will be assumed obtainable from past demand figures.

The λ_i vector is (.01, .02, .03, .04, .05) units/day and hence λ_T = .15/day or 37.5/year. The shortage penalty c'_s is 0.4/unit and c_R = 0.3, c_P = 0.5, hP = 0.1, ℓ = 25 days, L = 100 days.

17.5 The Author's Control Compared with Model VI

It will be recalled that for Model VI, three parameters of control (viz. α, y, z), were required. Since demand rates are not forecast for Model VII α is not required. Further, reversion to a fixed value of M is now possible since the total demand in the the combined lead time L_c is assumed to have a stationary distribution.

17.6 The Rôle of "Free Stock" as Criterion for Triggering a Procurement Order

It has been shown in this thesis that "Free Stock" is a good criterion for triggering a procurement order for cases where shortage cost is time-dependent. It is adaptable to cases of differing demand rates and applies to cases where shortage costs are time-independent (as for Model VI). However, it is clearly orientated towards shortage prevention. The argument that it is a parameter and can be changed to adapt to the different cost data for different complexes does not detract from the fact that it will not work so well for cases where it is correct to have shortages as it will where it is costly to have them.

Free Stock responds to the stock figure of the store with lowest stock. (If stock is present in the central store, the "lowest stock" means lowest stock after the usual hypothetical "equalising distribution" from the central store.) If the sensible* amount of buffer stock for the complex is generally about 50% of the combined lead time average demand (results show this is of the correct order), this figure is often likely to be reached when Free Stock is lower than the mean combined lead time L_c demand.

Clearly, we would like to establish M in the region of 15-17 for a Free Stock trigger criterion in the case of Model VII. There are times, however, when a sub-store of below average demand has experienced a high demand in a short period of time resulting in its stock level dropping to unity or below. This may set the procurement order trigger off even

---

* balancing shortage costs against holding costs.

if high levels of stock exist in the other sub-stores. To prevent high associated holding costs as a result of this, a higher level of holdback can be retained in the central store; this can be induced by (i) lowering the parameter z.

Alternatively (ii) the parameter M may be reduced (but for typical cycles this policy means numerous shortages occur as a result of ordering the procurement too late). The best policy may be neither of the two cited here; rather it may be: "Do not order a procurement and let this sub-store run down its stock". Since its average demand is low the expected shortage penalty resulting from its run-out is low.

The savings in overall replenishment costs as a result of not adopting policy (i), or in shortage costs as a result of not adopting policy (ii) are likely to be substantial.

Thus "Free Stock" may be improved on for the role of criterion for procurement order trigger. The suggestion made now is not only to ensure that stock configurations at time of procurement order are strategically equivalent, but also to obtain the approximately correct level by analysis.

17.7 Analysis to Establish When to Trigger a Procurement

For any given stock configuration, certain costs are expected to accrue as a result of waiting until another unit is demanded somewhere in the complex.

17.7 1 The addition to shortage costs by waiting

The expected shortage costs at present are in total approximately:-

$$c'_s \sum_{i=1}^{N} \phi(s'_{iH}, L_c, \lambda_i), \qquad s'_{iH} = \text{Max}\,\{s_{iH}, 0\}$$

where s_{iH} represents the hypothetically equalised stock at store i. The increment in expected shortage costs as a result of waiting until the next demand approximates to:-

$$\Delta c_{SH} = c'_s \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_T} \left(\phi(s'_{iH}-1, L_c, \lambda_i) - \phi(s'_{iH}, L_c, \lambda_i)\right) .$$

(If $(s'_{iH}-1) < 0$, the function $\phi(s'_{iH}-1, L_c, \lambda_i)$ is taken as unity.)

This expression follows since the probability of the next demand in the complex reducing the "hypothetically equalised" stock at store i to $(s'_{IH}-1)$ is (λ_i/λ_T).

17.7.2 Savings in holding cost by waiting

The savings in holding cost by this postponement of triggering the procurement are approximately* $hPQt_P$, where t_P is the time of postponement (in years). t_P, which will be the time until the next demand on the complex is exponentially distributed with mean $(1/\lambda_T)$ and therefore the probability of t_P is $\lambda_T\, e^{-\lambda_T t_P}$. The expected savings in holding cost are then:

$$\Delta C_{SAV} = hPQ\lambda_T \int_0^\infty t_P\, e^{-\lambda_T t_P}\, dt_P$$

$$= hPQ/\lambda_T$$

If $\Delta C_{SH} > \Delta C_{SAV}$ (1)

procurement order should be triggered now, and not be postponed.

17.8 Effect of the Suggested Trigger Criterion

M is now redundant as a parameter. This means a really useful saving in computing time since z is the only remaining parameter. The Allocation Rule requires an estimate of the time to reorder level of the complex. Hitherto, this time is taken as $(F-M)/\lambda_T$ where F is the present Free Stock. M, the System Stock level at which procurement order occurs will now vary.

17.9 Use of the Level M

A few trial runs on the computer can give an estimate for the average level of System Stock at which procurement order occurs whilst using the new trigger criterion. This average value is used for M in the Allocation Rule. Furthermore, the test (1) of 17.7.2 is not considered (to save unnecessary computing) until F drops to the level M+10.

It is clear that now F may be less than M without the procurement order having been triggered. If this is the case, then F is made equal to M, and the allocation computed with the usual formula.

17.10 Method of Location of Optimal z for Author's Method of Control

A 14 year period of control for the complex with z varying around the optimal region (which was determined by employing just two random number

---

* More correctly, this expression is $hP\left(Q-\left(\frac{N_{s1}+N_{s2}}{2}\right)\right)t_P$ where N_{s1}, N_{s2} are respectively the numbers of shortages expected if order occurs now or when the next unit is demanded. Generally for $(M-\lambda_T L_c) > 0$, this expression is well approximated by $hPQt_P$.

streams) in 0.3 intervals was simulated until convexity in z was noted for the averaged total cost values (three random number streams). A cubic approximation on these total cost figures indicated an estimate for optimal z.

17.11 Random Number Streams Chosen for Comparison with Cran's Control

These were other than those which were employed in obtaining an estimate of optimal z. They were Nos. 1811, 1831, 1891. The simulations were long enough such that at least twelve procurements would be experienced. 14 years was the length of time the complex was simulated for, for each random number stream. Each control was started off with the expected stock after a procurement arrival, spread over the stores according to the respective control rules.

17.12 Results of Location of Optimal Parameter z for Author's Control

Initial Trial Runs (M guessed as 24 in first instance) yielded M = 24.

Q was estimated at 47 (see Appendix 10).

17.12.1 Location of optimal z

Table 17.1 Total Costs for Different Stream Numbers

z	1711	1731	1751	MEAN
0.0	84.77	96.90	-	-
0.3	88.42	83.58	-	-
0.6	87.20	84.54	88.68	86.81
0.9	81.90	82.34	86.48	83.57
1.2	84.77	85.29	86.27	85.44

A cubic approximation yielded estimate for optimal z of 0.95.

17.13 Cran's Method of Location of the Optimal Parameters ($\bar{A}$, M) for His Control

It is sufficient for Cran's control to know the λ_i values, the lead times and the sub-store reorder levels before the simulations can proceed to establish the γ_R, γ_S characteristic curves.

The reorder level for sub-stores was set as low as possible subject to the condition that the probability of a shortage occurring during the lead time does not exceed 0. 01. For Model VII, these reorder levels were (3,4,4,5,5) for sub-stores 1 to 5, respectively, with demand rates (λ_1, λ_2, λ_3, λ_4, λ_5). This is consistent with Cran's suggestion, although he does not quote a figure for this shortage probability.

17.13.1 The simulator to produce the characteristic curves

At first Monte Carlo Simulation was attempted, but the work was tedious, especially for higher holdback factor $\overline{A}$ values. Minor modifications to the main simulation program allowed the accumulation of shortages and replenishments which would occur if System Stock were allowed to run down beyond zero to be recorded. The value of the initial System Stock was varied between 60 and 87 in steps of 3 over 10 simulations and each simulation employed different random number generating streams. For values of S between 90 and 0 in 5-unit intervals, the averages over the ten simulator runs of the cumulative number of shortages γ_S and replenishments γ_R were plotted against System Stock S. This was done for various holdback factors $\overline{A}$. The plot of γ_S against S is given in Fig. 38.

The expected number of shortages and number of replenishments are obtained from the γ_S, γ_R curves by the summations $\varepsilon(N_S) = \sum_S \gamma_S(S)f(M-S)$ and $\varepsilon(N_R) = \sum_S \gamma_R(S)f(M-S)$, where f(x) is the probability of demand x in the combined lead time L_c. Cran states that this was performed by approximate summation. The method used here was to obtain the product $\gamma(S)f(M-S)$ for S values over the range of S and the values of $\varepsilon(N_S)$, $\varepsilon(N_R)$ were obtained by the areas under the respective graphs of $\gamma_S f(M-S)$ and $\gamma_R(M-S)$ versus S. The $\varepsilon(N_S)$, $\varepsilon(N_R)$ curves are given by Figures 40, 41.

17.13.2 Use of the characteristic curves to locate the optimal solution

The extra information relating to the costs is utilized here. Following the method suggested by Cran, the procedure adopted is as follows:

(i) Guess M = 24, $\overline{A}$ = 0:
Figs. 40, 41 give N_S =2.5, N_R = 5.
Equation (c)* gives Q = 47;
Equation (d)* gives

$$\frac{\partial}{\partial M}\{\varepsilon(N_S)\} = -0.37:$$

Graph 40 gives M = 24:

Iteration

(ii) M = 24, $\overline{A}$ = 0:

Equation (e)* gives $-0.337 \frac{\partial}{\partial \overline{A}}\{\varepsilon(N_S)\} = c_R \frac{\partial}{\partial \overline{A}}\{\varepsilon(N_R)\}$

---

* see the review of Cran's work, 3.15.10.

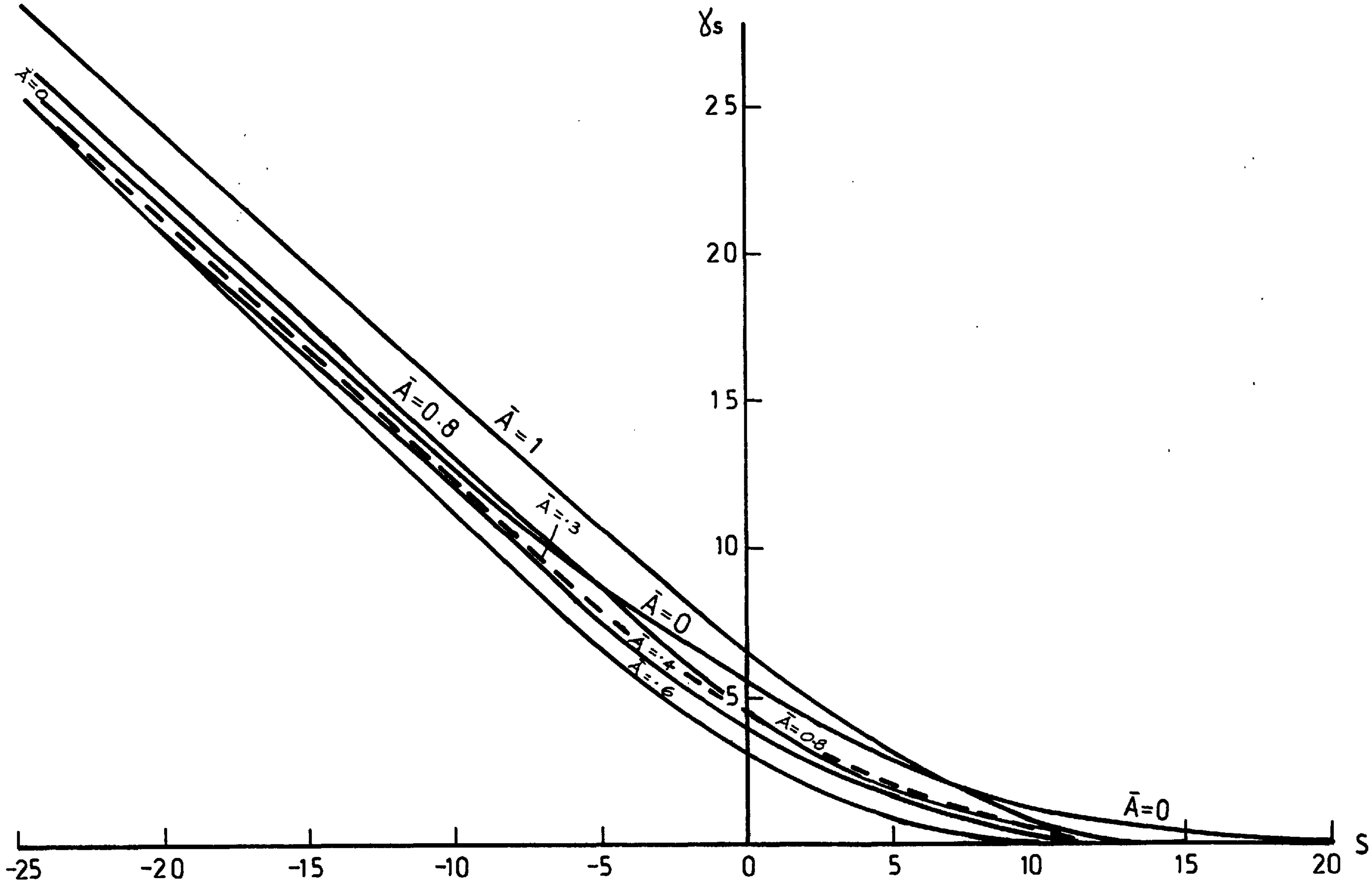


FIG. 38 NUMBER OF SHORTAGES VERSUS SYSTEM STOCK FOR SEVERAL Ā VALUES FOR CRAN CONTROL.

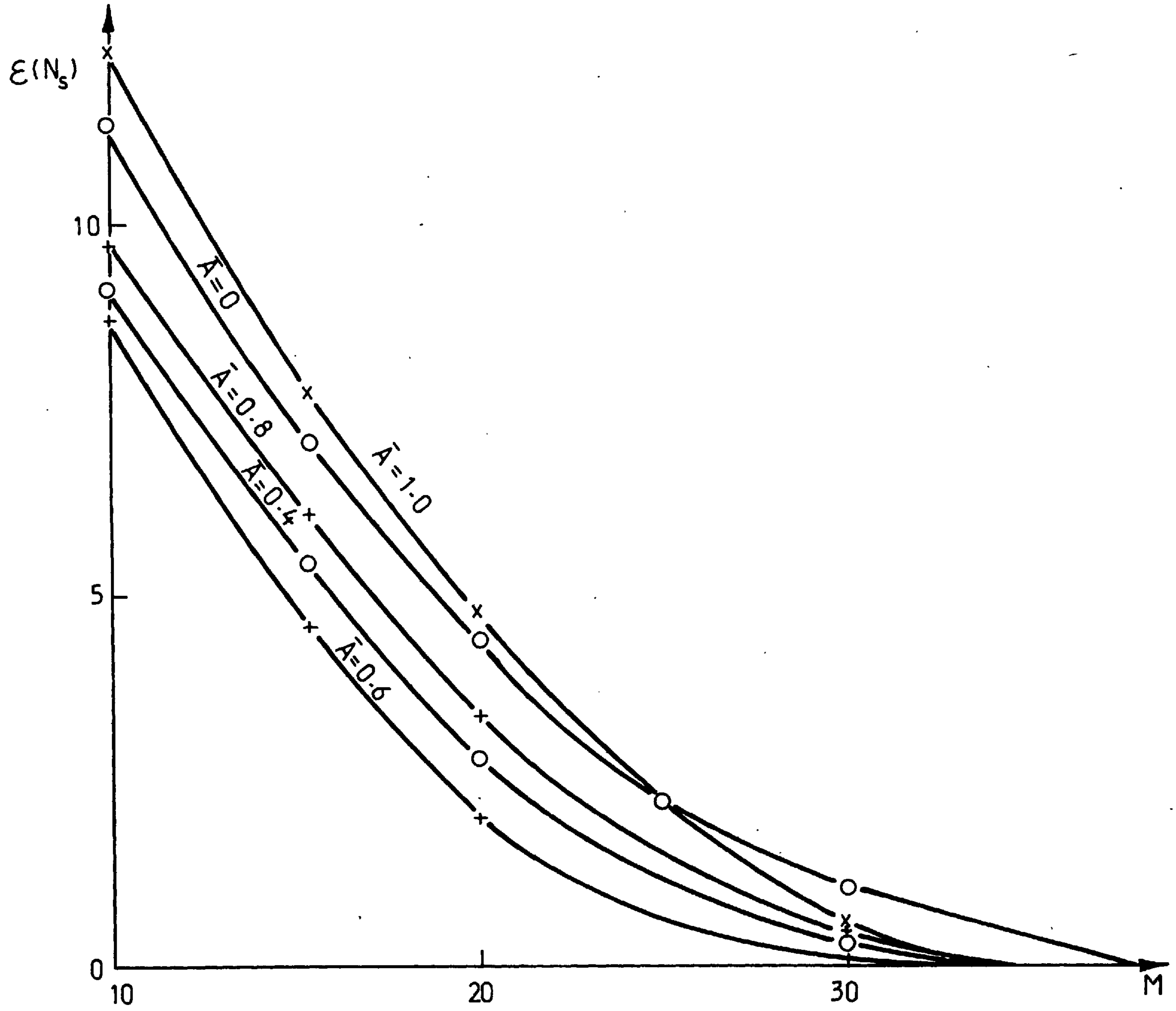


FIG. 40 EXPECTED No. OF SHORTAGES PER PROCUREMENT VERSUS SYSTEM R.O.L. FOR CRAN CONTROL.

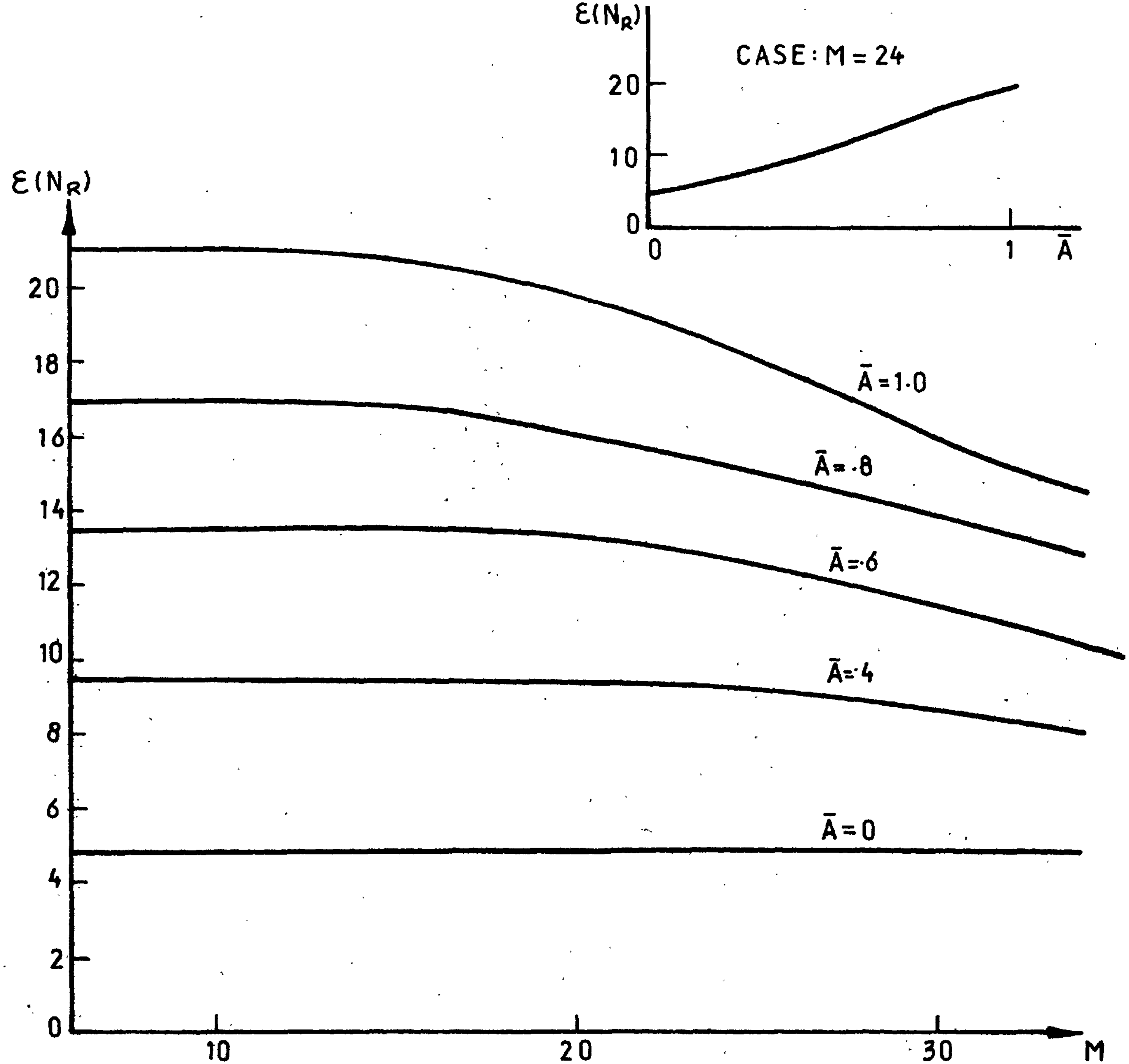


FIG. 41 EXPECTED No. OF REPLENISHMENTS PER PROCUREMENT PLOTTED AGAINST SYSTEM REORDER LEVEL M. FOR CRAN CONTROL.

Figs. 40, 41 give $\overline{A} = 0$

Hence, final solution is ($\overline{A} = 0$, $M = 24$, $Q = 47$).

17.14 Results of Experiment 15: The Comparative Performance with Cran's Control of the Author's Control applied to Model VII

17.14.1 Summary of Experiment 15*

14-year Simulation of Model VII with Controls 12A, 12C Using IBM 1130 System with "Simon" Simulation Language

Model VII Description

Item Value = 1

Cost of Procurement = 0.5

Cost of Supplying Sub-store = 0.3

Cost of Sub-store Shortage = 0.4

Cost of Stock Holding = 10% Value of Average Stock per year held.

Number of Sub-stores = 5

Lead time for Complex = 0.4 year

Lead time for Sub-stores = 0.1 year

Sub-store Demand Rates (0.01, 0.02, 0.03, 0.04, 0.05) items/day.

17.14.2 Author's suggestions (Control 12A)

Sub-store Reorder Quantity:

Case 1: Procurement on Order: "Share Mk VIA" (see 16.3.6)

Case 2: Procurement not on Order: "Allocation Rule"

Procurement Order Quantity (see Appendix 10 for computation) 47 units

Criterion of Reorder Level for Complex: Analysis (see 17.7)

Reorder Level for Sub-stores:

Case 1: Procurement on Order: Level in 3rd D.P. Model corresponding to T = L

Case 2: Procurement not on Order: Found from 3rd D.P. Model.

---

* The complete computer programs for this experiment are given in Program 4.

Table 17-2 Control 12A Applied to Model VII for Comparison Streams

$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{R_0}$	H.C.	T.C.	Stream No.
12	60	12.8	41.56	84.36	1811
13	70	19.6	44.61	91.71	1831
13	65	17.6	46.62	90.23	1891
12.7	65	16.7	44.93	88.77	MEAN

17.14.3 Control 12C: Cran's control

Sub-store Reorder Quantity: Cran Allocation Rule

Criterion of Reorder Level for Complex: System Stock

Reorder Level for Sub-stores: As low as possible, such that the probability of a shortage in the lead time does not exceed 0.01.

Table 17.3 Control 12C Applied to Model VII for Comparison Streams

$\overline{N}_P$	$\overline{N}_R$	$\overline{C}_{R_0}$	H.C.	T.C.	Stream No.
12	60	17.2	44.16	85.37	1811
13	65	23.6	43.04	92.64	1831
13	60	22.0	44.76	91.27	1891
12.7	61.7	20.9	43.99	89.76	MEAN

17.15 Lead in Triggering Instants Over Cran's Control

Since procurement quantities are identical for both controls (this is a chance occurrence), the instants of triggering procurement order are comparable. The following are the period in days by which the suggested trigger precedes Cran's trigger. Note that the average demand on the complex is 1/6.7 days.

Table 17-4

	Procurement Numbers											
Stream No.	1	2	3	4	5	6	7	8	9	10	11	12
1811	18	12	0	0	13	34	20	6	34	25	5	11
1831	20	-10	6	-27	2	27	1	7	4	40	20	4
1891	42	39	19	6	0	2	-16	23	17	-19	4	0

17.16 Significance Testing the Results: "t" Test

Control 12A Result	Control 12C Result	d_i	Stream No.
84.36	85.37	1.01	1811
91.71	92.64	0.93	1831
90.23	91.27	1.04	1891

t = 30.4; dof = 3.

The improvement over Cran's control is judged significant at the 0.05% level.

17.17 Conclusions from Chapter 17

The author's optimal result in general has an average reorder level in excess of that of Cran, although there are instances when the procurement order is triggered later than for Cran's control. It is to be expected, then, that holding cost will exceed that for Cran's control. Slightly higher than one sub-store delivery per cycle for the author's control indicates that on rare occasions, it is considered worthwhile to effect a second distribution to a sub-store. (This is a result of the "free" benefit of buffering sub-stores by "not necessarily replenishing" all sub-stores following procurement arrival.) The fact that higher z leads to an expected worse result indicates that too much of the stock "held back" (in the author's control) in the central store would be shipped to the first sub-store (i.e. first after the initial shipments to needy sub-stores on arrival of the procurement) which runs down to its reorder level.

The ability to achieve a significantly better result than Cran for this model which is specifically designed to favour Cran's control is felt to significantly strengthen support for the ideas in the suggested control policy. The suggestion to remove the reorder level M as a control parameter appears valuable since much computing effort to locate the optimal control solution is saved. This idea can be applied to the author's control generally. (This implies that for real complexes where demand rates are forecast, only the optimal parameters (α,z) need to be located before controlling the complex with these values.)

CHAPTER EIGHTEEN

SUMMARY

18.1 Summary of Experiment Details

Experiment No.	Model No.	Reference	Controls Considered
1	I	6.8	IA, 1B, 1C.
2	II	7.2	1A, 1C
3	III	8.5	2A, 2B, 2C
4	III	9.4	3A, 3B
5	III	9.6	4A
6	III	9.7	5A, 5B
7	III	12.2	6A
8	I, II	13.3.1	7A
9	III	14.3.1	8A
10	IV	15.2	9
10A	IV	15.14	10A
11	V	15.18	10B
12	VI	16.4	11
13	VI	16.8	11
14	VI	16.11	11
15	VII	17.14	12A, 12C

18.2 Model Descriptions

Model No.	Description	Reference
I	Zero Lead Time for Sub-stores	6.8
II	Zero Lead Time Costs favouring Cran Control	7.2
III	Non-zero Lead Time for Sub-stores	8.5
IV	"	15.1
V	"	15.1
VI	Different Demand Rates/Non-zero Lead Time	16.2
VII	"	17.4

18.2.1 Summary of Control Rules

Control Ref Page	No	Sub-Store Control: Reorder Level: Procurement on Order	Reorder Level: Procurement not on Order	Quantity Allocated: Procurement on Order	Quantity Allocated: Procurement not on order	Control for Complex: Reorder Level	Control for Complex: Reorder Qty.
129	1A	- 1 only		Share [1]	Allocation Quantity [2]	Free Stock [3] Parameter M	Q using an Expression for optimal Q see p. 115 (5.9)
129	1B	-1 or upon Procurement Arrival		Share [1]	Allocation Quantity [2]	Free Stock [3] Parameter M	- Ditto -
130	1C	- 1 or upon Procurement Arrival		Cran Allocation Qty. [6]		System Stock Parameter M	-- Ditto -
147	2A	- 1 only		Share [1]	Allocation Quantity [2]	Free Stock [3] Parameter M	- Ditto -
147	2B	- 1 or upon Procurement Arrival		Share [1]	Allocation Quantity [2]	Free Stock [3] Parameter M	- Ditto -
147	2C	- 1 or upon Procurement Arrival		Cran Allocation Qty. [6]		System Stock Parameter M	- Ditto -
160	3A	1st d p model variable		Share [1]	Allocation Quantity [2]	Free Stock [3] Parameter M	- Ditto -
161	3B	1st d p model [4] variable or on Proc. Arrival		Share [1]	Allocation Quantity [2]	Free Stock [3] Parameter M	- Ditto -
164	4A	1st d.p [4] Model variable	RoL when Proc. ordered (from 1st d.p Model)	Share [1]	Allocation Quantity [2]	Free Stock [3] Parameter M	- Ditto -

Control Ref. Page	No.	Sub-Store Control: Reorder Level: Procurement on order	Sub-Store Control: Reorder Level: Procurement not on Order	Sub-Store Control: Quantity Allocated: Procurement on order	Sub-Store Control: Quantity Allocated: Procurement not on Order	Control for Complex: Reorder Level	Control for Complex: Reorder Qty.
166	5A	Parameter		Share MK II (similar to 'share': see P163, 9.5.2)	Allocation[2] Quantity	Free Stock[3] Parameter M	Q using an Expression for opt[illegible] Q see p.115 (5.9)
167	5B	Parameter or on Proc. Arrival		Share MK II (similar to 'share': see P163, 9.5.2)	Allocation[2] Quantity	Free Stock[3] Parameter M	- Ditto -
185	6A	1st d.p.[4] Model vble.	2nd d.p.[5] Model vble.	Analytical Ration rule Share MK III (P175,10,2,3)	Allocation[2] Quantity	Free Stock[3] Parameter M	- Ditto -
195	7A	- 1 only		Analytical Ration rule Share MK IV (p190, 13.2)	Allocation[2] Quantity	Free Stock[3] Parameter M	- Ditto -
203	8A	1st d.p.[4] Model vble.	2nd d.p[5] Model vble.	Share MK IV B[11] see p. 201	Allocation[2] Quantity	Free Stock[3] Parameter M	- Ditto -
209	9	1st d.p.[4] Model vble.	2nd d.p.[5] Model vble.	Share MK V A[10]	Allocation[2] Quantity	Free Stock[3] Parameter M	- Ditto -
214 216	10A 10B	1st d.p.[4] Model vble.	2nd d.p.[5] Model vble.	Share MK VI[9]	Allocation[2] Quantity	Free Stock[3] Parameter M	- Ditto -

Control		Sub-Store Control				Control for Complex	
		Reorder Level		Quantity Allocated			
Ref Page	No	Procurement on order	Procurement not on order	Procurement on Order	Procurement not on order	Reorder Level	Reorder Qty.
240	11	3rd d.p.[7] Model vble.	RoL when Proc. is Ordered, from 3rd d.p. Model	Share MK VI A[8]	Allocation[2] Quantity	Parameter y relates[12] to M	Q using an Expression for optimal Q see p. 115 (5.9)
276	12A	3rd d.p.[7] Model vble.	RoL when Proc. is Ordered, from 3rd d.p. Model	Share MK VI A[8]	Allocation[2] Quantity	Analysis (see 17.7, p. 269)	- Ditto -
277	12C	As low as possible subject to "probability of shortage in lead time ≤.01" and Proc. Arrival		Cran Allocation Rule		System Stock Parameter M	Parameter Q

1. Share = Truncated value of Free Stock ÷ No. of Sub-Stores.
2. A = Mean Demand + Z x Std. Deviation of Demand in coverage time.
3. Free Stock F is computed by a hypothetical stock distribution from central store to sub-stores to minimize shortage cost. See pages 108-109 for further details.
4. See p. 159, (9.2.3.)
5. See p. 178, (11.2)
6. See p. 86, (3.15.3)
7. See p. 236, (16.3.3.1)
8. See p. 239, (16.3.6.)
9. See p. 212, (15.10)
10. see p. 210, (15.4)
11. See p. 201, (14.2)
12. See p. 240, (16,3.9)

18.3 Summary of Experimental Results

Expt.1: Significant Improvement over Cran's Result for "t" test at 0.5% level.

Expt.2: " " " " " " " " 5% level.

Expt.3: " " " " " " " " 1% level.

Expt.4: Use of 1st d.p. Model: Results worse than for Experiment 3: Fault in d.p. Model: Improvement suggested.

Expt.5: Use of amended d.p. Model: Improvement on Result for Unamended d.p. Model significant: Improvement in result over Control 3A (fixed r.o.l. at sub-stores) but insignificant.

Expt.6: Suggestion for a new Ration Rule "Share Mk II" without d.p. Model not a good idea.

Expt.7: Use of the new Ration Rule "Share Mk III" plus use of two d.p. Models for sub-store reorder level gives small improvement over Control 4A (Expt.5) but Ration Rule still needs improvement.

Expt.8: A new Ration Rule "Share Mk IV" a good idea ("t" test indicates significant improvement compared with "Share" for zero lead time models I, II at 5%, 10% levels, respectively).

Expt.9: New Ration Rule "Share Mk IVB" a good idea for non-zero lead time Model III ("t" significance test positive compared with previous best result (Control 6A) at 0.5% level). Further improvement on Ration Rule suggested.

Expt.10: Two suggested modifications ("Share Mk V") tested both improve performance but a further improvement is possible.

Expts. 10A,11: A new Ration Rule "Share Mk VI" applied to Models IV, V produced improved performance. However, "Share Mk VA" deemed to be best so far ("Share Mk VI" is strictly to be discounted because of its heuristic context).

Expts. 12,13,14: Modifications to concepts to allow for different and forecast demand rates over sub-stores (Movel VI) with shortage costs time-independent and where jump in demand is applied to a particular sub-store. Demonstrates the stability of the Control Rules.

Expt.15: (Comparison with Cran for a non-zero lead time, shortage cost time-independent model for different demand rates. Author's model incorporates analysis for trigger criterion and this model favours Cran control.) An improvement over Cran's control is judged by "t" test to be significant at the 0.05% level.

18.4 Ration Rule Summary

"Share" = I (Free Stock/N)

"Share Mk II" = Min{Share, Allocation} as in 9.5.2.

"Share Mk III" = Analytical Ration Rule as in 10.2.

"Share Mk IV" = Analytical Ration Rule, distinct improvement over last Ration Rule, specified in 13.2.

"Share Mk IVB" identical to "Share Mk IV", except for modification to take into account the step-function reorder level for sub-stores (case $T \leq L$) - see 14.2.

"Share Mk V" as "Share Mk IVB" except shortage cost computation considers a criterion which decides whether to consider individual stores' shortages or lump their stocks together and compute shortage accordingly - see 14.6).

"Share Mk VA" always computes the shortage costs of the sub-stores individually (see 15.4).

"Share Mk VI" incorporates a stronger rationing idea, limiting the ration quantity when central store stock drops below a "critical level" (see 15.8).

18.5 General Summary and Conclusions

This thesis was written in the belief that it fills to some extent a large gap in the literature in the field of controlling inventory shipments and storage quotas for a stores complex consisting of a central store feeding a number of stores (sub-stores).

Chapter One introduces the reader to the nature of the complex and indicates the problem of control. Chapters Two and Three review the literature pertinent to the subject. In Chapter Four the author presents the assumptions of the models of the complex to be used in this work and Chapter Five puts forward the author's first alternative method for controlling the complex. Chapter Six discusses the simulation techniques employed for comparing methods of control and details the simulation of control on the first model (Experiment One). Chapters Seven and Eight consider the author's second and third models and give the results of their simulation, comparisons being made (as in Experiment One) with Cran's control.

Chapters Nine, Ten, and Eleven present analysis consisting of two dynamic programming methods to establish the reorder level of sub-stores in two types of situation and put forward an alternative to the "Share" Ration Rule for sub-stores formulated in Chapter Five. At the end of Chapter Nine appears a short summary of the conclusions reached at that stage.

Chapter Twelve considers the simulation of the author's third model with the control rules governed by the analysis of the past three chapters, and concludes that the Rationing Rule to sub-stores can be further improved. Chapters Thirteen and Fourteen develop and test the new Ration Rule and present the results of the simulation of the complex when governed by this rule for each of the first three models.

This work is followed by the testing of suggested improvements to the Ration Rule. Chapter Fifteen introduces two further non-zero sub-store lead time models on which the author's control is tested. In the following chapter, a more generalized model in which the sub-store demand rates are forecast and different is introduced. For this model, the response of the control to a sudden increase in demand at a sub-store is observed.

In Chapter Seventeen, some analysis is formulated to establish the trigger for the procurement order. (This is felt to be particularly valuable.) The comparative performance of the author's suggestions and Cran's control is then observed on a representative model of the complex, in which the data is specifically designed to favour Cran's control ideas.

In conclusion to this thesis, the author puts forward the view that the potential cost savings inherent in applying sophisticated control ideas such as proposed in this work to a commerical complex are well worth the effort involved.

If all the information prerequisite for this type of control can be held centrally, the control decisions can be made at this point. The required control couple (z,α) can be obtained by simulation of a model of the commercial complex as demonstrated in this thesis, and then used in the control rules for on-line control of the complex.

The extent of programming necessary to include all the control rules is not too substantial for a medium-sized electronic computer.

Appendix Details

1. The Distribution of Sub-store Coverage Time Demand and a Note on Cran's hold-back factor. . . . . p. 285
2. Method by which an estimate of the variance of demand on a sub-store in the coverage time was sought.(when free stock is the procurement order trigger). . . . . p. 289
3. Method of calculation of the variance∿freestock relation constants of Appendix 2. . . . . p. 292
4. References. . . . . p. 294
5. The nature of $\psi(\bar{S},\bar{L},\bar{\lambda})$ and $\psi(s,\ell)$. . . . . p. 297
6. Detail of the first dynamic programming model. . . . . p. 300
7. The probability functions $P(s,\ell)$, $P'(s,\ell)$. . . . . p. 305
8. A note on the sub-store reorder level computations for the second dynamic programming model. . . . . p. 311
9. Further notes on the second dynamic programming model. . . . . p. 313
10. Calculation of reorder quantities of the complex for the various models. . . . . p. 316
11. List of symbols. . . . . p. 318
12. Glossary. . . . . p. 323
13. Exhibit details. . . . . p. 327
14. Program details. . . . . p. 345

APPENDIX ONE

THE MEAN AND VARIANCE OF SUB-STORE COVERAGE TIME DEMAND WHEN SYSTEM STOCK IS THE PROCUREMENT TRIGGER, AND A NOTE ON THE CRAN HOLD-BACK FACTOR

1.1 Introduction

This appendix establishes the mean and varience of the demand at a sub-store in the coverage time* when System Stock is t the procurement trigger.

1.2 Computation of Mean. and Variance of Sub-Store Demand in the Coverage Time

The complex reorder is triggered by System Stock at a level M, and the present System Stock is S. The sub-store demand affects the time until the complex reaches the trigger point. Let this time be t, so that there are R = S-M demands during t. Let r be the demand at sub-store i during this time.

The probability of R and r during time t is

$$\frac{e^{-\lambda_i t}(\lambda_i t)^r}{r!} \quad \frac{e^{-\lambda' t}(\lambda' t)^{R-r}}{(R-r)!} \qquad \text{where } \lambda' = (N-1)\lambda_i$$

Hence the probability of r gi en K is

$$R!\ \lambda_i^{\ r}\ \lambda'^{\ R-r} / \ r!\ (R-r)!\ \lambda_T^{\ R}$$

which has mean $R\lambda_i/\lambda_T$ and varience $R\lambda_i(\lambda_T - \lambda_i)/\lambda_T^2$,

The sub-store demand during the combined lead time L_c is Poisson and independent of demand prior to this time. Hence mean and variance during L_c are both $\lambda_i L_c$. Since demand in the coverage time is the sum of two independent parts (viz, during time to trigger and during the combined lead time) its mean is $(S-M)\ \lambda_i/\lambda_T + \lambda_i L_c$. The variance is $(S-M)\lambda_i(\lambda_T - \lambda_i)/\lambda_T^2 + \lambda_i L_c$

For the case $\lambda_T = N\lambda_i$, this expression for the variance becomes $(S-M)\ (1/N - 1/N^2) + \lambda_i L_c$. As a check, when N=1, variance is due only to demand during the lead time, and the variance should be independent of N.

The given expression satisfies this check.

For the particular case of 5 sub-stores,

Mean = $0.2\ (S-M) + \lambda_i L_c$

Variance = $0.16\ (S-M) + \lambda_i L_c$

*Time until Stock from the next Procurement can arrive at the Sub-Store.

APPENDIX TWO

METHOD OF SEEKING THE DISTRIBUTION OF DEMAND ON A SUB-STORE IN THE COVERAGE TIME WHEN FREE STOCK IS THE PROCUREMENT ORDER TRIGGER

2.1 Introduction

This appendix describes the method by which the distribution of the demand on a sub-store in the coverage time was sought when "Free Stock" is employed as the procurement order trigger. Although this method was found impractical in this case, the ideas appeared interesting enough to warrant insertion in an appendix.

2.2 The Problem of Variance of Sub-store Demand in the Coverage Time

The variance of this distribution is difficult to obtain mathematically. For a given λ_T, λ_i, L_c, we have from Appendix One the variance of coverage time demand with a trigger of System Stock to be a function of (S,M). Now the latter variance is known to be less than that of the distribution required here. This is because when System Stock is the trigger, reorder cannot occur for the complex until exactly (S-M) items are demanded over the sub-stores, whereas with Free Stock as the trigger reorder can occur when a total number of units either less than or greater than (S-M) have been demanded.

2.3 Approximations for Variance

A first approximation to the required variance could be to use the variance from Appendix One. If this expression is denoted in general by:

$$a + b(S-M),$$

a better approximation to the required variance is:

$$a + b(F-M),$$

where F is the "Free Stock" level. The mean is assumed to be satisfactorily approximated* by $(F-M)/N+L_c\lambda_i$.

2.4 The Suggested Method to Obtain the Variance of Demand

A more correct value of the required variance could possibly be acquired by the following method. The complex is simulated with an initial estimate of this variance as a+b(F-M). When a sub-store reorder level is reached for the first time (say when Free Stock = F_1) the Allocation Quantity is computed using the constants (a,b). At a time ℓ after the procurement arrival, the actual figure for demand on this sub-store since shipment will be available.

---

* for the case $\lambda_T = N\lambda_i$.

As time advances, so reorder for a sub-store will occur for another $F(F \neq F_1)$, say $F = F_2$, when the deviation from mean will be some value, say $dev_1(F_2)$ which is the difference between d_2 and $\{(F_2-M)/N+L_c\lambda_i\}$.

Now the variance vs.(F-M) relation may be re-estimated by linear regression between the points $[dev_1(F_1), (F_1-M)], [dev_2(F_2),(F_2-M)]$ The resulting relation may be denoted thus:

$$p + q(F-M).$$

As time advances with the Allocation Quantity being decided using the new constants p,q, so more values of deviation from mean for various F values will be available. At any instant, if the experienced deviations from mean for a level F_j are

$$dev_1(F_j),\ dev_2(F_j),\ \ldots\ dev_k(F_j),$$

then the best estimate of variance Var for the point (F_j-M) is

$$\frac{1}{k} \sum_{i=1}^{k} dev_i(F_j).$$

Again, the best estimate of the constants p,q, at any time will be found by linear regression on the $[Var,(F-M)]$ couples which are available.

2.4.1 Difficulties with initial estimates of p,q

Obviously, as the simulation begins, estimates of variance for different (F-M) values are poor and so the p,q constants in the Var∿(F-M) relation are correspondingly poor. It is envisaged that the beginning of the simulation will yield such poor estimates of p,q, that if these were to be used for the Allocation Rule, sensible estimates would only be obtained after a prohibitively long simulation run.

In attempting to overcome this problem, whenever p+q(F-M) is less than a+b(F-M), the allocation formula uses the constants a,b. This is because it is known that the variance of the required distribution will always exceed a+b(F-M).

APPENDIX THREE

DETAILS OF THE LINEAR REGRESSION IN APPENDIX TWO

For convenience we put $R = (F-M)$. The best straight line through the graph of points of Var vs. R in order to secure the best available approximation to the constants p,q (of the Var∼R relation of Appendix Two) to date, is obtained by regression analysis. The procedure is as follows:

All the different values of R for which an estimate of variance is available are summed, to yield ΣR. The number of different values of R is denoted by n. The sum of R^2 is called ΣR^2. The product of R and Var for each (Var,R) couple is computed and summed over all values of R to yield ΣProd. The sum of the squares of variance values is called ΣVarsqd.

If the best straight line through the points on the Var∼R graph of points is called p+qR, then p,q are given by:

$$p = (\Sigma R^2 \Sigma \text{Varsqd} - \Sigma R \Sigma \text{Prod})/E$$

$$q = (n\Sigma \text{Prod} - \Sigma \text{Varsqd} \Sigma R)/E$$

where $E = n\Sigma R^2 - (\Sigma R)^2$

APPENDIX FOUR

REFERENCES

1. F.Hanssmann, "Optimal Inventory Location in Control and Production Networks", Operations Research, Vol.7, No.4 (1959).

2. F. Hanssman, "Operations Research in Production and Inventory Control", Wiley, New York, 1962.

3. W. Lampkin, "Some Aspects of Stock Control in a Storage System with a Central Store and Several Sub-stores", Master of Science University Thesis, London, October 1963.

4. W. Lampkin, "A Second Look at Hanssmann's Inventory Control Model with Special Reference to the Central Store/Sub-Store Problem", Operations Research, Vol.14, No.1, Jan-Feb, 1966.

5. J.R. Lawrence, G.G. Stephenson, W. Lampkin, "A Stock Control Policy for Important Spares in a Two Level Stores System", Operational Research Quarterly, Vol.12, No. 4, 1961.

6. W. Lampkin and A.D.J. Flowerdew, "Computation of Optimum Reorder Levels and Quantities for a Reorder Level Stock Control System", Operational Research Quarterly, Vol.14, No.3, 1963.

7. P.R. Winters, "Inventory Control in a Multiple Warehouse System", Unpublished Thesis, Carnegiė Institute of Technology, 1960.

8. A.J. Clark, "A Dyanmic, Single Item, Multi-Echelon, Inventory Model", Rand Corporation Report No. RM-2297, Dec. 1958, Santa Monica, Calif.

9. A.J. Clark and H. Scarf, "Optimal Policies for a Multi-Echelon Inventory Problem", Planning Research Corporation Report No.P.R.C.R.-113, Los Angeles, Calif., 1959.
And in Management Science, Vol.6, No.4, July 1960, pp.475-490.

10. A.J. Clark and H. Scarf, "Approximate solutions to a Simple Multi-Echelon Inventory Problem", from "Studies in Applied Probability and Management Science", edited by K.J. Arrow, S. Karlin, and H.Scarf. Stanford University Press. And in Planning Research Corporation Report PRCR 209, May 1961.

11. K.J. Arrow, S. Karlin and H. Scarf, "Studies in the Mathematical Theory of Inventory and Production", Stanford University Press, 287pp., 1962.

12. G. Hadley and T.M. Whitin, "A Model for Procurement, Allocation, and Redistribution for Low Demand Items", Naval Research Logistics Quarterly, Vol.8, No.4, Dec. 1961.

13. M.F. Shakun, "An Application of Operations Research to Multiwarehouse Inventory Control", The Journal of Industrial Engineering, Vol.13, No.6, Nov. 1962.

14. J.A. Cran, "A Two-Level Inventory Control Rule in a Distribution System", ~~———~~ paper to the I.F.O.R.S. Conference, 1966 (Boston).

15. Kenneth F. Simpson, Jr., "A Theory of Allocation of Stocks to Warehouses", Operations Research, 7, No.6, Nov-Dec. 1959.

16. A.J. Gradwohl, "Case Studies in the Multi-Echelon Inventory Problem", P.R.C.R. -133, Planning Research Corp. Report, Los Angeles, Calif. Dec. 1959.

17. S.G. Allen, "Redistribution of Stock over Several User Locations", Naval Research Logistics Quarterly, Vol.5, No.4, Dec. 1958.

18. J.F. Magee, "Guides to Inventory Policy: II", Harvard Business Review, 34 No.2, March-Apl. 1956 (pp. 103-116).

19. S.A. Bessler, "A multi-Echelon Provisioning and Resupply Decision Procedure", U.S.G.R. & D.R. Order AD 611.175, Dec. 1964.

20. D. Hoekstra, "Multi-Level Inventory Systems: The Case of the Spare Aircraft Engines", U.S.G.R. & D.R. Order AD-615.892 (Frankford Arsenal, Pa., 1964).

21. P. Winters, "Multiple Triggers and Lot Sizes", Operations Research 9, No.5, Sept-Oct. 1961, pp. 621-634.

22. E. Berman, "The Monte Carlo Dtermination of Stock Redistribution", Operations Research 10, No.4, July-Aug. 1962, pp.500-506.

23. K. Borch, "Models for Hierarchial Decisions Involving Inventories", Academy of Management Journal (U.S.) 8 (1965), 3 (Sept.) pp. 179-189.

24. A.J. Gradwohl, "Inventory Policies for a Multi-Echelon Supply System with Fixed Costs at Lower Echelons", U.S.G.R. & D.R. Order AD-408.521. P.R.C. Report PRC-R-202, April, 1961.

APPENDIX FIVE

THE NATURE OF $\psi(\overline{S},\overline{L},\overline{\lambda})$ AND $\psi(s,\ell)$

5.1 Introduction

This appendix gives the method of computation of $\psi(\bar{S},\bar{L},\bar{\lambda})$, the number of item-days of shortage resulting from replenishing a store with stock level $\bar{S}$, when its rate of demand is $\bar{\lambda}$ and when stock will arrive at it in time $\bar{L}$.

For the special case of $\bar{\lambda} = \lambda_i$, $\bar{L} = \ell$, $\bar{S} = s$, this function is simply referred to as $\psi(s,\ell)$.

Obtaining the expression for $(\bar{S},\bar{L},\bar{\lambda})$

$$\psi(\bar{S},\bar{L},\bar{\lambda}) = \int_o^{\bar{L}} \sum_{i=\bar{S}+1}^{\infty} (i-\bar{S}) \frac{(\bar{\lambda}t)^i}{i!} e^{-\bar{\lambda}t} dt$$

$$= \sum_{i=\bar{S}+1}^{\infty} (i-\bar{S}) \int_o^{\bar{L}} \frac{(\bar{\lambda}t)^i}{i!} e^{-\bar{\lambda}t} dt \tag{1}$$

Now $$\int_o^{\bar{L}} \frac{(\bar{\lambda}t)^i}{i!} e^{-\bar{\lambda}t} dt = \frac{1}{\bar{\lambda}} \int_o^{\bar{L}} \frac{(\bar{\lambda}t)^i}{(i)!} e^{-\bar{\lambda}t} \bar{\lambda} dt$$

$$= \frac{1}{\bar{\lambda}} \{\text{Probability that } (i+1)^{th} \text{ demand occurs between 0 and } \bar{L}\}$$

$$= \frac{1}{\bar{\lambda}} \{\text{Probability that } (i+1) \text{ or more demands occur by time } \bar{L}\}$$

$$= \frac{1}{\bar{\lambda}} \sum_{j=i+1}^{\infty} \frac{(\bar{\lambda}\,\bar{L})^j}{j!} e^{-\bar{\lambda}\,\bar{L}} \tag{2}$$

Using equation (2), (1) becomes:-

$$\psi(\bar{S},\bar{L},\bar{\lambda}) = \frac{1}{\bar{\lambda}} \sum_{i=\bar{S}+1}^{\infty} (i-\bar{S}) \sum_{j=i+1}^{\infty} \frac{(\bar{\lambda}\,\bar{L})^j}{j!} e^{-\bar{\lambda}\,\bar{L}} \tag{3}$$

If we introduce a function $\rho(x)$ defined as follows:-

$$\frac{1}{\frac{1}{\lambda}} = \frac{(\bar{\lambda}\,\bar{L})^x}{x!} e^{-\bar{\lambda}\,\bar{L}} \equiv \rho(x),$$

then (3) becomes:-

$$\psi(\bar{S},\bar{L},\bar{\lambda}) = \sum_{i=\bar{S}+1}^{\infty} (i-\bar{S}) \sum_{j=i+1}^{\infty} \rho(j)$$

This summation can be seen to lead to:-

$$\psi(\bar{S},\bar{L},\bar{\lambda}) = \rho(\bar{S}+2) + 3\rho(\bar{S}+3) + 6\rho(\bar{S}+4) + 10\rho(\bar{S}+5) + \ldots$$

i.e.
$$\psi(\bar{S},\bar{L},\bar{\lambda}) = \frac{1}{\bar{\lambda}} \sum_{m=\bar{S}+2}^{\infty} \frac{(m-\bar{S}-1)(m-\bar{S})}{2} \frac{(\bar{\lambda}\bar{L})^m}{m!} e^{-\bar{\lambda}\bar{L}} \qquad (4)$$

Now $(m-\bar{S}-1)(m-\bar{S}) = m(m-1) - 2m\bar{S} + \bar{S}^2 + \bar{S}$ and (4) resolves into:-

$$\psi(\bar{S},\bar{L},\bar{\lambda}) = \frac{1}{\bar{\lambda}} \sum_{m=\bar{S}+2}^{\infty} \frac{m(m-1)}{2} \frac{(\bar{\lambda}\bar{L})^m}{m!} e^{-\bar{\lambda}\bar{L}}$$

$$- \frac{1}{\bar{\lambda}} \sum_{m=\bar{S}+2}^{\infty} \frac{2m\bar{S}}{2} \frac{(\bar{\lambda}\bar{L})^m}{m!} e^{-\bar{\lambda}\bar{L}}$$

$$+ \frac{1}{\bar{\lambda}} \sum_{m=\bar{S}+2}^{\infty} \frac{\bar{S}^2+\bar{S}}{2} \frac{(\bar{\lambda}\bar{L})^m}{m!} e^{-\bar{\lambda}\bar{L}}$$

which in turn resolves into:-

$$\psi(\bar{S},\bar{L},\bar{\lambda}) = \frac{\bar{\lambda}\bar{L}^2}{2} \left\{1 - \sum_{j=0}^{\bar{S}-1} \frac{(\bar{\lambda}\bar{L})^j}{j!} e^{-\bar{\lambda}\bar{L}}\right\}$$

$$- \bar{S}\bar{L} \left\{1 - \sum_{j=0}^{\bar{S}} \frac{(\bar{\lambda}\bar{L})^j}{j!} e^{-\bar{\lambda}\bar{L}}\right\}$$

$$+ \frac{\bar{S}^2+\bar{S}}{2\bar{\lambda}} \left\{1 - \sum_{j=0}^{\bar{S}+1} \frac{(\bar{\lambda}\bar{L})^j}{j!} e^{-\bar{\lambda}\bar{L}}\right\} \qquad (5)$$

The form of equation (5) is the form best suited to computational methods.

APBENDIX SIX

DETAILS OF THE FIRST DYNAMIC PROGRAMMING MODEL

6.1 The $\psi(s,\ell)$ Calculation for a Few s-Values

For the particular case of Model III we have:

$\lambda_i = .04, \quad \ell = 25.$

Then

$$\psi(s,\ell) = 12.5\,\{1-\lambda_i \sum_{j=0}^{s-1} \rho(j)\}$$

$$-25s\,\{1-\lambda_i \sum_{j=0}^{s} \rho(j)\}$$

$$+12.5(s^2+s)\,\{1-\lambda_i \sum_{j=0}^{s+1} \rho(j)\} \qquad (1)$$

from equation (5) of Appendix 5.

Let (1) be called:-

$\psi(s,\ell) = 12.5A' - 25sB' + C'D'$

where $C' = 12.5(s^2+s)$.

Then the component parts of the function are tabulated below for a few s-values.

s	A'	B'	C'	D'	$\psi(s,\ell)$
1	.63212	.26424	25	.08030	3.3026
2	.26424	.08030	75	.01899	0.7123
3	.08030	.01899	150	.00366	0.0189
4	.01899	.00366	250	.00059	
5	.00366	.00059	375		
6	.00059				

6.2 Illustration of Computation of a Few Cost Functions (refer to 9.2.3)

An interesting example to consider is C(1,1).

In the case of Model III, where the shortage cost, c_S, equals .4 we have:-

$C_A(1,1) = c_S\,\psi(1,\ell) + c_R = 0.4 \times 3.3026 + 0.3 = 1.6210$

$C_B(1,1) = (1-\bar{\lambda}_i)C(1,0) + \bar{\lambda}_i C(0,0)$

$= (1-\bar{\lambda}_i)c_S\psi(1,\ell) + \bar{\lambda}_i c_S\,\psi(0,\ell)$

Equation (1) yields $\psi(0,\ell) = \lambda_1 \ell^2/2 = 12.5$ for Model III.

Then $C_B(1,1) = .96 \times .4 \times 3.3026 + .04 \times .4 \times 12.5$

$= 1.4682.$

Since $C_B(1,1) < C_A(1,1)$,

we have $C(1,1) = C_B(1,1) = 1.4682$, and the decision function associated with the state (s,T) of (1,1), viz. D(1,1) is 0 (i.e. wait until the next time instant before further review).

Consider now the (s,T) state of (1,2)

$$C_A(1,2) = c_S\psi(1,\ell) + c_R = 1.6210$$

$$C_B(1,2) = .96\ C(1,1) + .04\ C(0,1)$$

$$= .96 \times 1.4682 + .04\ C(0,1)$$

We require the value of C(0,1).

Consider $C_A(0,1) = c_S\psi(0,\ell) + c_R = .4 \times 12.5 + .3 = 5.3$

$$C_B(0,1) = .96\ C(0,0) + .04\ C(-1,0)$$

$$= .96 \times 5.0 + .04 \times c_S\big(\psi(0,\ell) + \ell\big)$$

$$= 4.8 + .04 \times .4 \times 37.5$$

$$= 5.4$$

Thence $C(0,1) = C_A(0,1) = 5.3$

and $C_B(1,2) = .96 \times 1.4682 + .04 \times 5.3$

$$= 1.6215$$

Thence $C(1,2) = \text{Min}\ \{C_A(1,2);\ C_B(1,2)\}$

$$= 1.6210$$

and the Decision Function D(1,2) is (marginally) 1.

6.3 To Show that, if D(s,T) = 1 for T = 1, then it is also 1 for all $L \geq T \geq 1$

Consider the state (s,T) for $T \geq 2$.

$$C_A(s,T) = c_S\psi(s,\ell) + c_R \qquad (a)$$

$$C_B(s,T) = (1-\bar{\lambda}_1)\ C(s,T-1) + \bar{\lambda}_1 C(s-1,T-1) \qquad (b)$$

Equation (b) shows $C_B(s,T) > C(s,T-1)$ (c)

since it is clear that $C(s-1,T) > C(s,T)$ for all s,T.

(a) implies $C_A(s,T) = C_A(s,T-1)$ (d)

From (c), (d), it follows that if $C(s,T-1) = C_A(s,T-1)$ (which implies $D(s,T-1) = 1$), then

$$C_B(s,T) > C_A(s,T),$$

i.e. $$D(s,T) = 1.$$

Now the original supposition gives us $D(s,1) = 1$, and so $D(s,T) = 1$ for all $L \geq T \geq 1$.

6.4 Method of Computation of the D(s,T) Values

(i) Calculation of C(1,1):

$$C_A(1,1) = c_R + c_S\psi(1,\ell)$$

$$C_B(1,1) = (1-\bar{\lambda}_i)C(1,0) + \bar{\lambda}_i C(0,0)$$

$$= (1-\bar{\lambda}_i)c_S\psi(1,\ell) + \bar{\lambda}_i c_S\psi(0,\ell)$$

hence C(1,1) and D(1,1).

(ii) Calculation of C(1,2):

$$C_A(1,2) = c_R + c_S\psi(1,\ell)$$

$$C_B(1,2) = (1-\bar{\lambda}_i)C(1,1) + \bar{\lambda}_i C(0,1)$$

(ii.a) Sub-calculation for C(0,1):

as in 6.2, hence C(1,2) and D(1,2).

(iii) Calculation of C(1,T) for all defined T > 2:

$$C(1,T) = C_A(1,T) = c_S\psi(1,\ell) + c_R, \text{ independent of } T$$

(iv) Calculation of C(s,1) for all s > 2 (in order; s = 2,3...):

$$C_A(s,1) = c_R + c_S\psi(s,\ell)$$

$$C_B(s,1) = (1-\bar{\lambda}_i)C(s,0) + \bar{\lambda}_i C(s-1,0)$$

Hence $$C(s,1) = \text{Min}\left(C_A(s,1);\ C_B(s,1)\right)$$

and D(s,1)

(v) Calculation of C(s,T) for all remaining s,T is obtainable from results included in (i) to (iv).

6.5 Consideration of Decisions for Negative Stock Levels

It is not true, as might have been considered obvious, that for all cases where a sub-store has a negative stock level, it should necessarily be replenished forthwith.

6.5.1 The (s,T) state (-1,1)

$$C_A(-1,1) = c_R + c_S\{\tfrac{1}{2}\lambda_i \ell^2 + \ell\}$$

$$\begin{aligned} C_B(-1,1) &= (1-\bar{\lambda}_i)C(-1,0) + \bar{\lambda}_i C(-2,0) + c_S \\ &= (1-\bar{\lambda}_i)\{c_S(\tfrac{1}{2}\lambda_i\ell^2+\ell)\} + \bar{\lambda}_i\{c_S(\tfrac{1}{2}\lambda_i\ell^2 + 2\ell)\} + c_S \\ &= c_S(\tfrac{1}{2}\lambda_i\ell^2 + \ell) + c_S(1+\lambda_i\ell) \end{aligned}$$

$$\left.\begin{aligned} C(-1,1) &= C_A(-1,1) \\ D(-1,1) &= 1 \end{aligned}\right\} \text{ ONLY IF } c_R/c_S < (1+\lambda_i\ell)$$

6.5.2 The (s,T) state (-2,1)

$$C_A(-2,1) = c_R + c_S\{\tfrac{1}{2}\lambda_i\ell^2 + 2\ell\}$$

$$\begin{aligned} C_B(-2,1) &= (1-\bar{\lambda}_i)C(-2,0) + \bar{\lambda}_i C(-3,0) + 2c_S \\ &= (1-\bar{\lambda}_i)\{c_S(\tfrac{1}{2}\lambda_i\ell^2 + 2\ell)\} + \bar{\lambda}_i\{c_S(\tfrac{1}{2}\lambda_i\ell^2 + 3\ell)\} + 2c_S \\ &= c_S(\tfrac{1}{2}\lambda_i\ell^2+2\ell) + c_S(2+\lambda_i\ell) \end{aligned}$$

$$\left.\begin{aligned} C(-2,1) &= C_A(-2,1) \\ D(-2,1) &= 1 \end{aligned}\right\} \text{ ONLY IF } c_R/c_S < (2+\lambda_i\ell)$$

In general $D(-n,1) = 1$ for negative n only if $c_R/c_S < (n+\lambda_i\ell)$.

Similarly $D(-1,2) = 1$ and $D(s,T) = 1$ for all $L \geq T \geq 1$, and $s \leq 0$, in the case $c_R/c_S < (1+\lambda_i\ell)$.

6.6 Model III Specialities

The fact that $D(0,1) = 1$ (since $C_A(0,1) < C_B(0,1)$ in 6.2) and the theorem of 6.3 together give $D(0,T) = 1$ for all $1 \leq T \leq L$.

Since for Model III, $c_R < c_S(1+\lambda_i\ell)$, we have, for all negative s, and all $1 \leq T \leq L$, $D(s,T) = 1$.

APPENDIX SEVEN

THE PROBABILITY FUNCTIONS $P(s,\ell)$; $P'(s,\ell)$

(A): THE PROBABILITY OF A SUB-STORE WITH A GIVEN NOTIONAL STOCK LEVEL 's' AT TIME OF PROCUREMENT ORDER RUNNING DOWN ITS STOCK TO REORDER LEVEL IN CENTRAL STORE LEAD TIME - THE PROBABILITY $P(s,\ell)$

(B): THE PROBABILITY OF A SUB-STORE WITH A GIVEN NOTIONAL STOCK LEVEL 's' AT TIME $T(T < L)$ BEFORE PROCUREMENT ARRIVAL RUNNING DOWN ITS STOCK TO REORDER LEVEL WITHIN TIME T HENCE:- THE PROBABILITY $P'(s,\ell)$

7A The Probability $P(s,\ell)$

Consider a sub-store with a notional stock level 's' at time of the reorder level of the complex. The probability* of it reaching a reorder point again before the ordered procurement arrives** is equal to the probability that the sub-store experiences demand such that the reorder level step function line is crossed before procurement arrival. This function is illustrated for Model III by Fig. 39.

Let the step lengths be of time values $t_1, t_2 \ldots t_5$ etc., as in Fig. 39. The demand in time intervals t_1, t_2, etc. will be r_1, r_2 etc.

The probability of the notional stock level of the sub-store meeting or going below the step-line will equal:-

(Probability of incurring such a demand that the stock level drops below the step-line in the interval t_1) +

(Probability of a demand in t_2 such that the stock level crosses the step-line in time interval t_2, for all possible values of $s-r_1 > 4$, i.e. of $r_1 \leq s-5$) +

Probability of crossing step-line in intervals t_3, t_4, t_5

$$\text{i.e.}\quad P(s,\ell) = \sum_{r_1=s-4}^{\infty} p_1(r_1)$$

$$+ \sum_{r_1=0}^{s-5} p_1(r_1) \sum_{r_2=s-r_1-3}^{\infty} p_2(r_2)$$

$$+ \sum_{r_1=0}^{s-5} p_1(r_1) \sum_{r_2=0}^{s-r_1-4} p_2(r_2) \sum_{r_3=s-r_1-r_2-2}^{\infty} p_3(r_3)$$

$$+ \sum_{r_1=0}^{s-5} p_1(r_1) \sum_{r_2=0}^{s-r_1-4} p_2(r_2) \sum_{r_3=0}^{s-r_1-r_2-r_3} p_3(r_3)$$

$$+ \sum_{r_4=s-r_1-r_2-r_3-3}^{\infty} p_4(r_4)$$

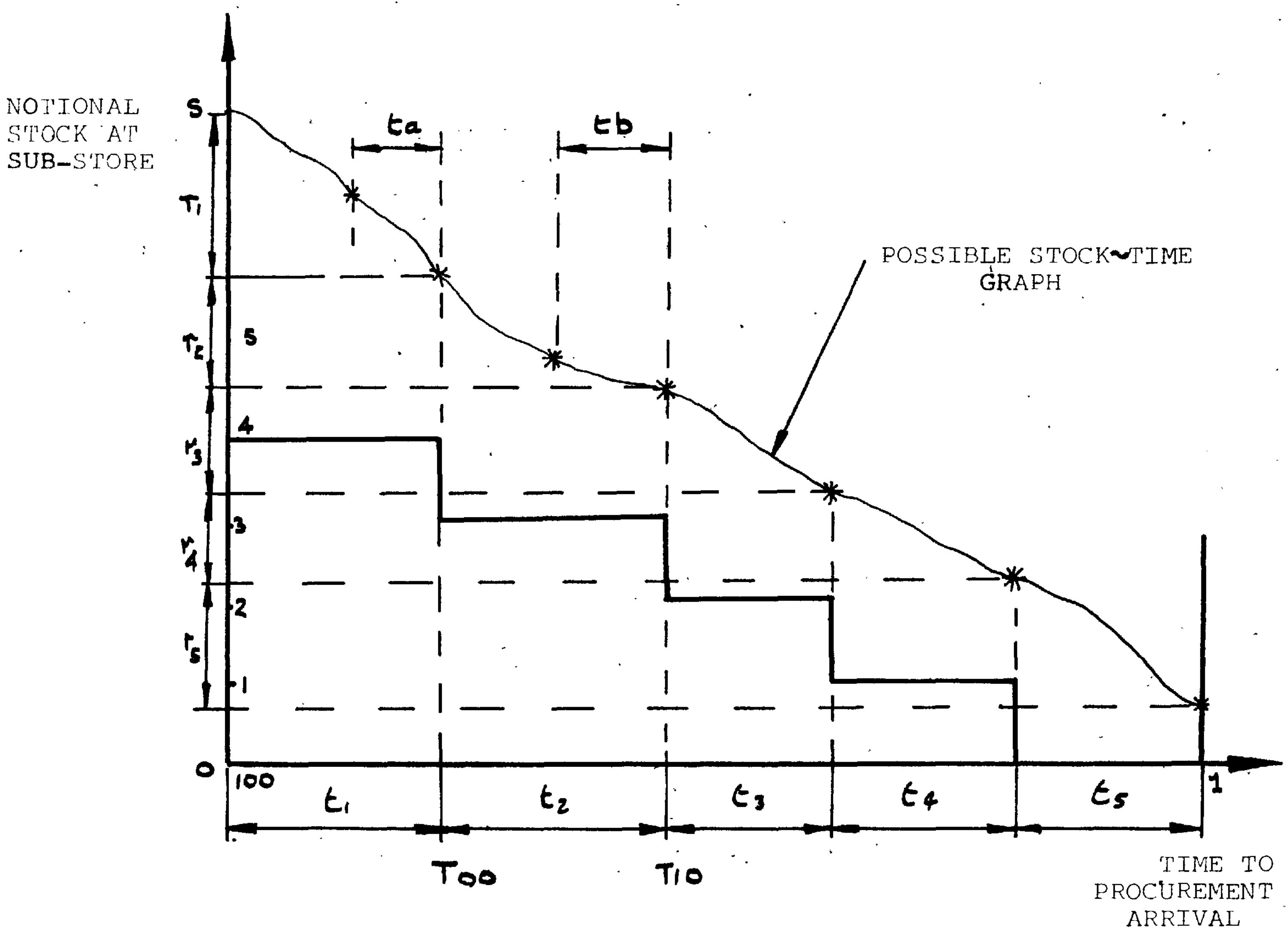
---

* This probability, $P(s,\ell)$ is required for use in the 2nd Dynamic Programming Model.

** If the reorder point is in fact met, the assumption is made that the Ration Rule will decide upon a RationQuantity which is greater than the present notional stock, i.e. a shipment will always occur when the reorder level is met.

FIGURE 39

THE REORDER LEVEL VS. T STEP FUNCTION FOR MODEL III
(NOT TO SCALE)



$$+ \sum_{r_1=0}^{s-5} p_1(r_1) \sum_{r_2=0}^{s-r_1-4} p_2(r_2) \sum_{r_3=0}^{s-r_1-r_2-3} p_3(r_3) \sum_{r_4=0}^{s-r_1-r_2-r_3-4} p_4(r_4)$$

$$\sum_{r_5=s-r_1-r_2-r_3-r_4-4}^{\infty} p_5(r_5)$$

where $p_k(r_k)$ indicates the probability of a demand of r_k units in the time interval t_k. Thus, for Model III we have:-

$$p_1(r_1) = e^{-\mu_1} \frac{\mu_1^{r_1}}{r_1!} \; ; \; \mu_1 = \lambda_i t_1 = .04 \times (100-62) = 1.52$$

$$p_2(r_2) = e^{-\mu_2} \mu_2^{r_2}/r_2! \; ; \; \mu_2 = \lambda_i t_2 = .04 \times (62-25) = 1.56$$

$$p_3(r_3) = e^{-\mu_3} \mu_3^{r_3}/r_3! \; ; \; \mu_3 = \lambda_i t_3 = (23-7) \times .04 = 0.64$$

$$p_4(r_4) = e^{-\mu_4} \mu_4^{r_4}/r_4! \; ; \; \mu_4 = \lambda_i t_4 = (7-2) \times .04 = 0.20$$

$$p_5(r_5) = e^{-\mu_5} \mu_5^{r_5}/r_5! \; ; \; \mu_5 = \lambda_i t_5 = (2-1) \times .04 = 0.04.$$

The probability of crossing the step-line at all, then, $P(s,\ell)$ is given by:

$$P(s,\ell)$$

$$= 1 - \sum_{r_1=0}^{s-5} p_1(r_1)$$

$$+ \sum_{r_1=0}^{s-5} p_1(r_1) \{1 - \sum_{r_2+o}^{s-r_1-4} p_2(r_2)\}$$

$$+ \underbrace{\sum_{r_1=0}^{s-5} p_1(r_1)}_{A} \underbrace{\sum_{r_2=0}^{s-r_1-4} p_2(r_2)}_{B} \{1 - \sum_{r_3=0}^{s-r_1-r_2-r_3} p_3(r_3)\}$$

$$+ A \times B \underbrace{\sum_{r_3=0}^{s-r_1-r_2-r_3} p_3(r_3)}_{C} \{1 - \sum_{r_4=0}^{s-r_1-r_2-r_3-r} p_4(r_4)\}$$

$$+ \quad A \times B \times C \sum_{r_4 = 0}^{s-r_1-r_2-r_3-2} p_4(r_4) \{1 - \sum_{r_5 = 0}^{s-r_1-r_2-r_3-r_4-1} p_5(r_5)\}$$

This is not too difficult a function for electronic computation.

7B <u>The Probability P'(s,ℓ)</u>

The probability of a sub-store with a given notional stock level and at a given time T_1 before procurement arrival running down it stock to re-order point before the procurement arrives is a more generalized case of the probability considered in 7A.

If we require P'(s,ℓ) for a sub-store at time $T=T_o$, then we require the probability of crossing the step-line in intervals t_a, t_2, t_3, t_4, t_5 (see Fig. 39). This probability is as given in 7A where $t_a = t_1$, and thus $\mu_1 = \lambda_i t_a$. If P'(s,ℓ) is required for $T_{oo} > T > T_{10}$ then we require the probability of crossing the step-line in intervals t_b, t_3, t_4, t_5. The latter probability is equal to:-

(Probability of the notional stock level meeting or going below the step-line in interval t_b) +

(Probability of a demand in t_2 such that the stock level crosses the step-line in time interval t_3, for all possible values of $s-r_b > 3$, i.e. of $r_b \leq s-4$)* +

(Probability of crossing step-line in intervals t_4, t_5)

$$= \sum_{r_b=s-3}^{\infty} p_b(r_b)$$

$$+ \sum_{r_b=0}^{s-4} p_b(r_b) \sum_{r_3=s-r_b-2}^{\infty} p_3(r_3)$$

$$+ \sum_{r_b=0}^{s-4} p_b(r_b) \sum_{r_3=0}^{s-r_b-3} p_3(r_3) \sum_{r_4=s-r_b-r_3-3}^{\infty} p_4(r_4)$$

$$+ \sum_{r_b=0}^{s-4} p_b(r_b) \sum_{r_3=0}^{s-r_b-3} p_3(r_3) \sum_{r_4=0}^{s-r_b-r_3-4} p_4(r_4) \sum_{r_5=s-r_b-r_3-r_4-r_5}^{\infty} p_5(r_5)$$

---

* r_b represents the demand in time interval t_b.

As in 7A $p_k(r_k)$ represent the probability of r_k demands in time period t_k.

Thus $$p_b(r_b) = e^{-\mu_b} \frac{\mu_b^{r_b}}{r_b!} \; ; \; \mu_b = \lambda_i t_b$$

$$p_1(r_1) = e^{-\mu_1} \frac{\mu_1^{r_1}}{r_1!} \; ; \; \mu_1 = \lambda_i t_1, \text{ etc.}$$

Similar summations for $P'(s,\ell)$ when $T_{10} > T_{00} > T_{20}$ are obtainable in the same way.

APPENDIX EIGHT

A NOTE ON THE METHOD OF COMPUTATION OF REORDER LEVEL OF SUB-STORES IN THE CASE WHEN A PROCUREMENT ORDER IS NOT OUTSTANDING (THE SECOND DYNAMIC PROGRAMMING MODEL OF CHAPTER ELEVEN)

For any (s,F) couple,

$C_A(s,F) = c_R$ ~~$\times P(s,\ell)$~~ $+ c_S\psi(s,\ell) + C_{M_1}(F)$ is easily computed.

We have, also,

$$C_B(s,F) = \bar{\lambda}_i C(s-1,F-1) + 4\bar{\lambda}_i C(s,F-1) + (1-5\bar{\lambda}_i)C(s,F) \qquad \text{(A)}$$

Thus, <u>if $C(s,F) = C_A(s,F)$</u>

this implies $C_A(s,F) < C_B(s,F)$

i.e. that $C(s,F) < \bar{\lambda}_i C(s-1,F-1) + 4\bar{\lambda}_i C(s,F-1) + (1-5\bar{\lambda}_i)C(s,F)$

and the implication is then that

$$5\bar{\lambda}_i C(s,F) < \bar{\lambda}_i C(s-1,F-1) + 4\bar{\lambda}_i C(s,F-1)$$

$$\equiv \quad C_A(s,F) < .2C(s-1,F-1) + .8C(s,F-1)$$

The test then is:-

<u>is</u> $\quad C_A(s,F) < .2C(s-1, F-1) + .8C(s,F-1)$?

If true, then $C(s,F) = C_A(s,F)$

whence the decision $D(s,F) = 1$ (replenish now)

If false, then $C(s,F) = C_B(s,F)$ (B)

Using (A), (B) is thus transformed ...

$$C(s,F) = \bar{\lambda}_i C(s-1,F-1) + 4\bar{\lambda}_i C(s,F-1) + (1-5\bar{\lambda}_i)C(s,F)$$

i.e. $C(s,F) = .2C(s-1,F-1) + .8C(s,F-1)$

and $\quad D(s,F) = 0$ (wait)

The first requirement for solution of the dynamic programming model is the nature of $D(0,F)$. For most cost and lead time data combinations (certainly for that of Model III) it is clear that $D(0,F) = 1$ (replenish now).

APPENDIX NINE

FURTHER NOTES ON THE 2ND DYNAMIC PROGRAMMING MODEL

(SOME TABULATED VALUES FOR THE SOLUTION OF THE DYNAMIC PROGRAMMING MODEL FOR THE REORDER LEVEL OF THE COMPLEX WHILST A PROCUREMENT IS NOT ON ORDER (PLUS AN INDICATION OF THE TESTING PROCEDURE FOR OBTAINING COSTS OF BEST DECISIONS IN THE MODEL))

9.1 Tabulated Data for Solution of the 2nd d.p. Model for Model III for the Case (M = 50, z = 0.8)

s	1	2	3	4	5	6	7	8	9	10
$P(s,\ell)^{+}$	1.0	1.0	1.0	1.0	.894	.675	.433	.239	.116	.050
$c_S\psi(s,\ell)$	1.321	0.285	0.051	0.008	0.0001					

F	A(F)	$C_{M_1}(F)$	ROL(F)	F	A(F)	$C_{M_1}(F)$	ROL*(F)
51-57	Various	0	4	76	13	.0199	4
58-60	10	0	5	77	13	.0171	5
61	10	.0034	5	78	13	.0147	5
62	10	.0027	5	79	14	.0341	3
63	10	.0022	5	80	14	.0261	4
64	10	.0017	5	81	14	.0117	6
65	11	.0100	4	82	14	.0034	8
66	11	.0083	4	83	15	.0040	5
67	11	.0069	5	84	15	.0006	8
68	11	.0057	5	85	15	.0003	8
69	11	.0047	5	86	15	.0002	8
70	12	.0165	3	87	16	.0003	6
71	12	.0139	4	88	16	.0002	9
72	12	.0118	4	89	16	.0002	9
73	12	.0100	5	90	16	.0001	9
74	12	.0268	3	91	16	.0001	9
75	12	.0231	4	92	17	.0001	7
				93	17	.0001	9

It will be seen that the ROL~F function is not monotonic. As with the $C_M(F)$ function, this is a result of the step nature of the A(F) function. What occurs is that when the allocation quantity steps up one as F is increased, the associated cost of Maldistribution Type 1 increases, and the dynamic programming model reflects this in lowering the reorder level of sub-stores (hence tending to prevent a replenishment).

+ See Appendix Seven.

* Reorder Level at Sub-stores as a function of Free Stock.

9.2 An Example of the Testing Procedure for Obtaining Costs of Best Decisions at a Few s-values for an F-value of 46

Test: Is $C_A(s,F) \overset{?}{<} .2C(s-1,F-1) + .8C(s,F-1)$

For s=1, $C_A(1,46) = c_S\psi(s,\ell) + c_R$

$= 1.321 + .3 = 1.621$

The test, then, is:-

Is $1.621 \overset{?}{<} .2($~~5~~ 5.3$) + .8(1.621)$

Answer: Yes

Hence $C(1,46) = C_A(1,46) = 1.621$

$D(1,46) = 1.$

For s=2, $C_A(2,46) = c_S\psi(2,\ell) + c_R = .585$

$C(1,45) = c_S\psi(1,\ell) + c_R = 1.621$

The test is:-

Is $0.585 \overset{?}{<} .2(1.621) + .8($~~1.621~~ 0.585$)$

Answer: Yes

Hence $C(2,46) = .585$

$D(2,46) = 1.$

The result is similar for s=3, 4.

But for s=5;

$C_A(5,46) = c_S\psi(5,\ell) + c_R$

$= .001 + .3 = .301$

$C(4,45) = c_S\psi(4,\ell) + c_R$

$= .308$

The test is:-

Is $.301 \overset{?}{<} .2(.308) + .8C(5,45)$

i.e. Is $.301 \overset{?}{<} .2(.308) + .8\{\psi(5,\ell)c_S + c_R \times P(s,\ell)\}$

i.e. Is $.301 \overset{?}{<} .2(.308) + .8($~~.001~~ .0001 $+ .3 \times .8939)$

$\equiv$ Is $.301 \overset{?}{<}$ ~~.2692~~ .2762

Answer: No.

Then $C(5,46) =$ ~~.2692~~ .2762; $D(5,46) = 0$ (Wait)

The reorder level for an F-value of 46 is thus 4.

APPENDIX TEN

CALCULATION OF REORDER QUANTITY OF THE COMPLEX FOR THE VARIOUS MODELS

10.1 Summary of Reorder Quantities for the Complex (Procurement Order Quantities) for the Various Models

These incorporate the computation:

$$Q = \sqrt{\left(2\lambda_T/hP\ \{c_P + c_R \times \varepsilon(N_R) + \varepsilon(C_{RO})\}\right)}$$

Table A

						Estimates		
Model No.	λ_T units/yr	c_P	c_R	hP	ℓ days	$\varepsilon(C_{RO})$	$\varepsilon(N_R)$	Q
I	50	0.5	0.3	0.1	0	0	10	60
II	125	0.5	0.03	0.1	0	2	10	85
III	50	0.5	0.3	0.1	25	0	10	60
IV	50	0.5	0.03	0.1	25	1	20	45
V	50	0.5	0.3	0.1	50	1	15	75
VI	37.5	0.5	0.3	0.1	25	1	10	60
VII	37.5	0.5	0.3	0.1	25	0.8	5.5	47

APPENDIX ELEVEN

LIST OF SYMBOLS

11.1 Introduction to the List of Symbols

In the text of the thesis, many symbols are defined wherever they occur. However, a large number have universal application. It is these that the author has particularly in mind in compling this list. This list, does, however, include symbols often defined where they occur in the work.

11.2 List of Symbols

A, $A(F,M,z)$, $A(z)$, $A(F)_{M,z}$	Allocation Quantity
$\overline{A}$	Cran Hold Back Factor (related to HBF in Appendix One)
b_i, $i = 1,2,3 \ldots$	Values of Extra Delivery Quantities
$\hat{B}$	Optimal Value of B
B	Considered Ration Quantity
$\overline{B}$	Average Buffer Stock Maintained in Complex
$C(s,T)$, $C_A(s,T)$, $C_B(s,T)$	Costs involved in First Dynamic Programming Model
$C(s,F)$, $C_A(s,F)$, $C_B(s,F)$	Costs involved in Second Dynamic Programming Model
$C(B)$	Costs associated with a Ration Quantity B
C_{M_1}	Cost of Maldistribution Type One
c_P	Cost of a unit procurement
c_R	Cost of a unit replenishment
C_{RO}	Cost of Run-out per procurement
$\overline{C}_{RO}$	Cost of Run-out per Simulation
c_S	Cost of shortage per item-day
c_S'	Cost of unit shortage (Experiment 12)
$\overline{C}_S(d,B)$	Total cost of shortages a s function of d,B
d	Random variable of Demand in time (T_o-1)
$D(s,T)$	Decision function in 1st d.p. Method
dof	Degrees of Freedom for "t" significance test

$\overline{d}$	Average difference in "t" significance test
D(s,F)	Decision function in 2nd d.p. method.
$\mathcal{E}(x)$	Expected value of x
F	Free Stock
G	The Physical Stock in System at any time
h	Rate per annum cost of capital
HBF	Cran Hold Back Factor
H.C.	Holding Costs for the length of a simulation run
I(x)	Integer value of x
k(d,B)	Number of extra deliveries if Ration Quentity is B
ℓ	Lead time to Sub-stores
$\overline{L}$	Argument in $(\overline{S},\overline{L},\overline{\lambda})$ computation
L	Procurement Lead Time
L_c	Combined Lead Time (= L+ℓ)
M	Reorder Level for Complex
N	Number of Sub-stores
$\overline{N}$	Number of observstions for the "t" significance test
N_R	Number of Replenishments per procurement
$\overline{N}_P$	Number of Procurements per simulation
$\overline{N}_R$	Number of Replenishments per simulation
P	Value per unit inventory
P(s,ℓ)	Probability used in the 2nd d.p. Model
P'(s,ℓ)	Probability used in the Ration Rule "Share Mk III"
Q	Procurement Quantity

$Q_{OPTIMAL}$	Optimal Procurement Quantity
ROL, ROL_{SS}	Sub-store Reorder Level
s	Sub-store Notional Stock Level
s_o	Sub-store Reorder Level at T=L
S	System Stock
$\overline{S}$	Argument for stock level in $\psi(\overline{S},\overline{L},\overline{\lambda})$ computation
T, T_o	Time before Procurement Arrival
T_b	Decision Boundary T in Experiment 12
T.C.	Total Cost of Control in a Simulation
y	Control Parameter for Reorder Level in Model VI
z	Control Parameter in Allocation Rule
λ_i	Mean Sub-store Demand Rate per unit time
$\overline{\lambda}_i$	Mean Sub-store Demand Rate per day
$\overline{\lambda}$	Argument for Demand Rate used in $\psi(\overline{S},\overline{L},\overline{\lambda})$ computation
λ'	Defined as $(N-1)\lambda_i$
λ_T	Mean Total Demand Rate in complex per unit time
$\overline{\lambda}_T$	Mean Total Demand Rate in complex per day
Δ	Difference between Free Stock and System Stock at time of reorder for the complex
$\psi(\overline{S},\overline{L},\overline{\lambda})$	Expected cost of shortage for a store with stock $\overline{S}$ with replenishment due time $\overline{L}$ hence, with a demand rate $\overline{\lambda}$
$\psi(s,\ell)$	As $\psi(\overline{S},\overline{L},\overline{\lambda})$ where ($\overline{S} = s$ ($\overline{L} = \ell$ ($\overline{\lambda} = \lambda_i$
α	Experimental Smmothing Factor as used in Experiment 12

ε	Value of "Step-End" (see 16.3.3)
η	Defined as $\lambda_T L_c$ in Experiment 12
$\phi(\overline{S},\overline{L},\overline{\lambda})$	As ψ function except shortage cost is not time-dependent.

APPENDIX TWELVE

GLOSSARY

Allocation

When a sub-store reaches its reorder point whilst no procurement is on order, it is "allocated" a certain quantity of stock. This quantity is called the "Allocation Quantity" and the actual quantity to be shipped will equal the Allocation Quantity less the Notional Stock of the Sub-store. The Allocation Quantity is denoted by A, where $A = A(F,M,z)$

Complex

This is the name given to the whole system of stores.

Configuration

This is the complete description of the stock levels for all stores in the complex.

Coverage Time

This equals the expected time until a sub-store can next be replenished from the next procurement. Thus when a procurement is expected in time T, the coverage time is $(T+\ell)$.

Demand

Demands are in unit quantities and occur on sub-stores only.

Decision Function

In the First Dynamic Programming Model, Decision Functions are functions of notional sub-store stock 's' and time before procurement arrives in the complex, 'T'. For all the three Dynamic Programming Models, Decisions Functions may be either 0 or 1, zero corresponding to "do not replenish" and unity corresponding to "replenish now".

Free Stock

This is a stock level of the complex especially conceived to represent a useful stock level. It is denoted by F, and is sometimes termed the F-value.

Fair Share

This is equal to the value of F divided by the number of sub-stores.

Maldistribution

Whenever the stock is distributed over the sub-stores in a non-optimal fashion and either shortages result or extra inventory holding occurs (as a consequence of ordering a procurement when the System Stock exceeds M) then maldistribution is said to have occurred.

"Necessarily Replenish" and "Not Necessarily Replenish" Policies

The above policies are the alternative procedures adopted for sub-stores when the ordered procurement arrives in the complex at the central store.

Notional Stock

The sum of a store's physical stock and stock on order less any back-orders.

Order and Reorder

The above synonyms refer to the decision to consider shipment from the next higher level to the store under consideration. If a sub-store is at a "reorder point", then an order from the central store is considered for it.

Procurement

The name given to the order for the complex. It is received at the central store.

Ration Rule

The name given to the formula or procedure for establishing the amount to be "rationed" to a sub-store when the complex has a procurement on order. The quantity shipped equals the Ration Quantity less the notional stock of the sub-store.

Replenishment

The name given to a sub-store delivery.

System Stock

The sum of the central store stock and sub-store notional stocks.

Trigger

The mechanism deciding when to order a procurement.

Virtual Stock

The stock present at a store less any backorders.

APPENDIX THIRTEEN

EXHIBIT DETAILS

EXHIBIT B: Extract from Simulation of Experiment 10
Control 9 applied to Model IV
Shows unsuitability of "Share Mk V" (see 15.3)

EXHIBIT C: Extract from Simulation of Experiment 10 with
Control 9 (using "Share Mk VA") - see 15.4 to 15.6

EXHIBIT D: Simulation Details for Control 11 applied to
Model VI for a close-to-optimal parameter combination (Experiment 12, see 16.4)

Exhibit B

EXHIBIT 'B'

```
50 -0.301729
0
0
0
0
NIM= 108
NIM= 106
NIM= 104
```

TIME	STOCK(1)	TRANSIT(1)	STOCK(2)	TRANSIT(2)	STOCK(3)	TRANSIT(3)	STOCK(4)	TRANSIT(4)	STOCK(5)	TRANSIT(5)	CENTRAL STORE STOCK
0	5	6	5	0	5	0	5	0	5	0	54
0	5	6	5	6	5	0	5	0	5	0	48
0	5	6	5	6	5	6	5	0	5	0	42
0	5	6	5	6	5	6	5	6	5	0	36
0	5	6	5	6	5	6	5	6	5	6	30
3	5	6	5	6	5	6	5	6	4	6	30
5	5	6	5	6	5	6	5	6	3	6	30
6	5	6	5	6	5	6	4	6	3	6	30
11	5	6	4	6	5	6	4	6	3	6	30
15	5	6	4	6	4	6	4	6	3	6	30
17	5	6	4	6	4	6	3	6	3	6	30
19	5	6	4	6	4	6	3	6	2	6	30
20	4	6	4	6	4	6	3	6	2	6	30
25	10	0	4	6	4	6	3	6	2	6	30
25	10	0	10	0	4	6	3	6	2	6	30
25	10	0	10	0	10	0	3	6	2	6	30
25	10	0	10	0	10	0	9	0	2	6	30
25	10	0	10	0	10	0	9	0	8	0	30
26	10	0	10	0	9	0	9	0	8	0	30
26	10	0	10	0	9	0	9	0	7	0	30
26	10	0	10	0	9	0	9	0	7	2	28
28	10	0	10	0	8	0	9	0	7	2	28
39	9	0	10	0	8	0	9	0	7	2	28
45	9	0	10	0	8	0	8	0	7	2	28
47	9	0	10	0	8	0	7	0	7	2	28
47	9	0	10	0	8	0	7	1	7	2	27
51	9	0	10	0	8	0	7	1	9	0	27
61	9	0	9	0	8	0	7	1	9	0	27
63	9	0	9	0	7	0	7	1	9	0	27
[illegible]	8	0	9	0	7	0	7	1	9	0	27
67	8	0	9	0	7	0	6	1	9	0	27
70	8	0	9	0	7	0	6	1	8	0	27

72	8	0	9	0	7	0	7	0	8	0	27	
73	8	0	9	0	7	0	7	0	7	0	27	
73	7	0	9	0	7	0	7	0	7	0	27	
81	7	0	9	0	7	0	6	0	7	0	27	
81	7	0	9	0	7	0	6	1	7	0	26	
83	7	0	9	0	7	0	5	1	7	0	26	
83	7	0	9	0	7	0	5	2	7	0	25	
92	6	0	9	0	7	0	5	2	7	0	25	
94	5	0	9	0	7	0	5	2	7	0	25	
94	5	1	9	0	7	0	5	2	7	0	24	
95	5	1	8	0	7	0	5	2	7	0	24	
100	5	1	7	0	7	0	5	2	7	0	24	
106	5	1	7	0	7	0	6	1	7	0	24	
108	5	1	7	0	6	0	6	1	7	0	24	
108	5	1	7	0	6	0	7	0	7	0	24	
111	5	1	7	0	6	0	7	0	6	0	24	
115	5	1	7	0	6	0	6	0	6	0	24	
115	5	1	6	0	6	0	6	0	6	0	24	
119	5	1	5	0	6	0	6	0	6	0	24	
119	6	0	5	0	6	0	6	0	6	0	24	
119	6	0	5	0	5	0	6	0	6	0	24	
123	6	0	5	0	5	0	5	0	6	0	24	
125	6	0	5	0	5	0	4	0	6	0	24	*
125	6	0	5	7	5	0	4	0	6	0	17	DELIVERY
125	6	0	5	7	5	3	4	0	6	0	14	QTY TOO
125	6	0	5	7	5	3	4	3	6	0	11	HIGH
125	5	0	5	7	5	3	4	3	6	0	11	
125	5	3	5	7	5	3	4	3	6	0	8	
125	5	3	4	7	5	3	4	3	6	0	8	
130	5	3	4	7	4	3	4	3	6	0	8	
134	5	3	4	7	3	3	4	3	6	0	8	
135	4	3	4	7	3	3	4	3	6	0	8	
136	4	3	4	7	2	3	4	3	6	0	8	
136	4	3	4	7	2	4	4	3	6	0	7	
141	4	3	4	7	2	4	4	3	5	0	7	
141	4	3	4	7	2	4	4	3	5	1	6	
143	3	3	4	7	2	4	4	3	5	1	6	
143	2	3	4	7	2	4	4	3	5	1	6	
143	2	5	4	7	2	4	4	3	5	1	4	
143	2	5	4	7	1	4	4	3	5	1	4	
143	2	5	4	7	1	5	4	3	5	1	3	
144	2	5	4	7	1	5	4	3	4	1	3	
144	2	5	4	7	1	5	4	3	4	2	2	
147	2	5	3	7	1	5	4	3	4	2	2	
148	2	5	3	7	0	5	4	3	4	2	2	
148	2	5	3	7	0	6	4	3	4	2	1	
150	2	5	10	0	0	6	4	3	4	2	1	
150	2	5	10	0	3	3	4	3	4	2	1	
150	2	5	10	0	3	3	7	0	4	2	1	
150	5	2	10	0	3	3	7	0	4	2	1	
154	5	2	10	0	2	3	7	0	4	2	0	
154	5	2	10	0	2	4	7	0	4	2	0	
158	5	2	10	0	2	4	6	0	4	2	0	
161	5	2	10	0	3	3	6	0	4	2	0	
161	5	2	9	0	3	3	6	0	4	2	0	
165	5	2	9	0	2	3	6	0	4	2	0	
166	5	2	9	0	2	3	6	0	5	1	0	
168	7	0	9	0	2	3	6	0	5	1	0	
168	7	0	9	0	3	2	6	0	5	1	0	
168	7	0	9	0	3	2	5	0	5	1	0	
168	7	0	8	0	3	2	5	0	5	1	0	
169	7	0	8	0	3	2	5	0	4	1	0	
169	7	0	8	0	3	2	5	0	5	0	0	
173	6	0	8	0	3	2	5	0	5	0	0	
173	6	0	8	0	4	1	5	0	5	0	0	
175	5	0	8	0	4	1	5	0	5	0	0	
177	5	0	8	0	4	1	5	0	4	0	0	
179	5	0	8	0	5	0	5	0	4	0	0	
179	5	0	8	0	4	0	5	0	4	0	0	
185	5	0	8	0	4	0	4	0	4	0	0	
193	5	0	8	0	3	0	4	0	4	0	0	
199	4	0	8	0	3	0	4	0	4	0	0	
206	4	0	8	0	3	0	4	0	3	0	0	
207	4	0	7	0	3	0	4	0	3	0	0	
212	4	0	7	0	3	0	3	0	3	0	0	
212	4	0	7	0	2	0	3	0	3	0	0	
216	4	0	7	0	2	0	2	0	3	0	0	
218	4	0	7	0	1	0	2	0	3	0	0	
223	3	0	7	0	1	0	2	0	3	0	0	
225	3	3	7	0	1	0	2	0	3	0	42	* PROCURE-
225	3	3	7	0	1	5	2	0	3	0	37	-MENT
225	3	3	7	0	1	5	2	4	3	0	33	ARRIVAL
225	3	3	7	0	1	5	2	4	3	3	30	
225	3	3	7	0	0	5	2	4	3	3	30	
225	3	3	7	0	0	6	2	4	3	3	29	
227	3	3	7	0	-1	6	2	4	3	3	29	
227	3	3	7	0	-1	7	2	4	3	3	28	
228	3	3	7	0	-1	7	2	4	3	3	28	

Exhibit C

EXHIBIT 'C'

```
50 -0.301729
    0
    0
    0
    0
NIM= 108
NIM= 106
NIM= 104
```

0	5	6	5	0	5	0	5	0	5	0	54
0	5	6	5	6	5	0	5	0	5	0	48
0	5	6	5	6	5	6	5	0	5	0	42
0	5	6	5	6	5	6	5	6	5	0	36
0	5	6	5	6	5	6	5	6	5	6	30
3	5	6	5	6	5	6	5	6	4	6	30
5	5	6	5	6	5	6	5	6	3	6	30
6	5	6	5	6	5	6	4	6	3	6	30
11	5	6	4	6	5	6	4	6	3	6	30
15	5	6	4	6	4	6	4	6	3	6	30
17	5	6	4	6	4	6	3	6	3	6	30
19	5	6	4	6	4	6	3	6	2	6	30
20	4	6	4	6	4	6	3	6	2	6	30
25	10	0	4	6	4	6	3	6	2	6	30
25	10	0	10	0	4	6	3	6	2	6	30
25	10	0	10	0	10	0	3	6	2	6	30
25	10	0	10	0	10	0	9	0	2	6	30
25	10	0	10	0	10	0	9	0	8	0	30
26	10	0	10	0	9	0	9	0	8	0	30
26	10	0	10	0	9	0	9	0	7	0	30
26	10	0	10	0	9	0	9	0	7	2	28
28	10	0	10	0	8	0	9	0	7	2	28

39	9	0	10	0	8	0	9	0	7	2	28
45	9	0	10	0	8	0	8	0	7	2	28
47	9	0	10	0	8	0	7	0	7	2	28
47	9	0	10	0	8	0	7	1	7	2	27
51	9	0	10	0	8	0	7	1	9	0	27
61	9	0	9	0	8	0	7	1	9	0	27
63	9	0	9	0	7	0	7	1	9	0	27
65	8	0	9	0	7	0	7	1	9	0	27
67	8	0	9	0	7	0	6	1	9	0	27
70	8	0	9	0	7	0	6	1	8	0	27
72	8	0	9	0	7	0	7	0	8	0	27
73	8	0	9	0	7	0	7	0	7	0	27
73	7	0	9	0	7	0	7	0	7	0	27
81	7	0	9	0	7	0	6	0	7	0	27
81	7	0	9	0	7	0	6	1	7	0	26
83	7	0	9	0	7	0	5	1	7	0	26
83	7	0	9	0	7	0	5	2	7	0	25
92	6	0	9	0	7	0	5	2	7	0	25
94	5	0	9	0	7	0	5	2	7	0	25
94	5	1	9	0	7	0	5	2	7	0	24
95	5	1	8	0	7	0	5	2	7	0	24
100	5	1	7	0	7	0	5	2	7	0	24
106	5	1	7	0	7	0	6	1	7	0	24
108	5	1	7	0	6	0	6	1	7	0	24
108	5	1	7	0	6	0	7	0	7	0	24
111	5	1	7	0	6	0	7	0	6	0	24
115	5	1	7	0	6	0	6	0	6	0	24
115	5	1	6	0	6	0	6	0	6	0	24
119	5	1	5	0	6	0	6	0	6	0	24
119	6	0	5	0	6	0	6	0	6	0	24
119	6	0	5	0	5	0	6	0	6	0	24
123	6	0	5	0	5	0	5	0	6	0	24
125	6	0	5	0	5	0	4	0	6	0	21
125	6	0	5	3	5	0	4	0	6	0	19
125	6	0	5	3	5	2	4	0	6	0	16
125	6	0	5	3	5	2	4	3	6	0	16
125	5	0	5	3	5	2	4	3	6	0	14
125	5	2	5	3	5	2	4	3	6	0	14
125	5	2	4	3	5	2	4	3	6	0	14
130	5	2	4	3	4	2	4	3	6	0	14
134	5	2	4	3	3	2	4	3	6	0	13
134	5	2	4	3	3	3	4	3	6	0	13
135	4	2	4	3	3	3	4	3	6	0	13
136	4	2	4	3	2	3	4	3	6	0	11
136	4	2	4	3	2	5	4	3	5	0	11
141	4	2	4	3	2	5	4	3	5	1	10
141	4	2	4	3	2	5	4	3	5	1	10
143	3	2	4	3	2	5	4	3	5	1	8
143	3	4	4	3	2	5	4	3	5	1	8
143	3	4	4	3	1	5	4	3	5	1	8
143	2	4	4	3	1	5	4	3	4	1	8
144	2	4	4	3	1	5	4	3	4	2	7
144	2	4	4	3	1	5	4	3	4	2	7
147	2	4	3	3	1	5	4	3	4	2	7
148	1	4	3	3	1	5	4	3	4	2	6
148	1	5	3	3	1	5	4	3	4	2	6
150	1	5	6	0	1	5	4	3	4	2	6
150	1	5	6	0	3	3	4	3	4	2	6
150	1	5	6	0	3	3	7	0	4	2	6
150	3	3	6	0	3	3	7	0	4	2	6
154	2	3	6	0	3	3	7	0	4	2	5
154	2	4	6	0	3	3	7	0	4	2	5
158	2	4	6	0	3	3	6	0	4	2	5
159	2	4	6	0	4	2	6	0	4	2	5
161	2	4	6	0	6	0	6	0	4	2	5
161	2	4	5	0	6	0	6	0	4	2	4
161	2	4	5	1	6	0	6	0	4	2	4
163	1	4	5	1	6	0	6	0	4	2	3
163	1	5	5	1	6	0	6	0	5	1	3
166	1	5	5	1	6	0	6	0	5	1	3
168	3	3	5	1	6	0	6	0	5	1	3
168	3	3	5	1	6	0	5	0	5	1	2
168	3	3	5	1	6	0	5	1	5	1	2
168	3	3	4	1	6	0	5	1	5	1	1
168	3	3	4	2	6	0	5	1	5	1	1
169	3	3	4	2	6	0	5	1	4	2	0 } lines 'a'
169	3	3	4	2	6	0	5	1	5	1	0
169	3	3	4	2	5	0	5	1	5	1	0
173	3	2	4	2	5	0	5	1	5	1	0
173	4	2	4	2	5	0	5	1	5	1	0
175	4	2	4	2	4	0	5	1	4	1	0
177	4	2	4	2	4	0	5	1	4	1	0
179	5	1	4	2	4	0	5	1	4	1	0
179	4	1	4	2	4	0	5	1	4	1	0
185	4	1	4	2	4	0	4	1	4	1	0
[illegible]	4	1	5	1	4	0	4	1	4	1	0
[illegible]	4	0	5	1	4	0	4	0	4	1	0
[illegible]	4	0	5	1	4	0	5	0	4	1	0

193	5	0	6	0	4	0	5	0	4	1	0
193	4	0	6	0	4	0	5	0	4	1	0
194	4	0	6	0	4	0	5	0	5	0	0
199	4	0	6	0	3	0	5	0	5	0	0
206	4	0	6	0	3	0	5	0	4	0	0
207	4	0	5	0	3	0	5	0	4	0	0
212	4	0	5	0	3	0	4	0	4	0	0
212	3	0	5	0	3	0	4	0	4	0	0
216	3	0	5	0	3	0	3	0	4	0	0
218	2	0	5	0	3	0	3	0	4	0	0
223	2	0	5	0	2	0	3	0	4	0	41
225	2	4	5	0	2	0	3	0	4	0	40
225	2	4	5	1	2	0	3	0	4	0	36
225	2	4	5	1	2	4	3	3	4	0	33
225	2	4	5	1	2	4	3	3	4	2	31
225	2	4	5	1	2	4	3	3	4	2	31
225	1	4	5	1	2	4	3	3	4	2	30
225	1	5	5	1	2	4	3	3	4	2	30
227	0	5	5	1	2	4	3	3	4	2	29
227	0	6	5	1	2	4	3	3	3	2	29
228	0	6	5	1	2	4	3	3	3	3	28
228	0	6	5	1	2	4	3	3	3	3	28
232	0	6	5	1	1	4	3	3	3	3	27
232	0	6	5	1	1	5	3	3	3	3	27
239	-1	6	5	1	1	5	3	3	3	3	27
240	-1	6	5	1	1	5	2	3	3	3	27
246	-1	6	5	1	1	5	2	3	2	3	27
250	-1	6	4	1	1	5	2	3	2	3	27
250	3	2	4	1	1	5	2	3	2	3	27
250	3	2	5	0	1	5	2	3	2	3	27
250	3	2	5	0	5	1	2	3	2	3	27
250	3	2	5	0	5	1	5	0	2	3	27

2

PROCURE-MENT ARRIVAL

Exhibit D

EXHIBIT 'D'

Initial Forecasts From 4-Year Smoothing

0

0

0

0 — λ_1

0.011 — λ_2

0.016 — λ_3

0.025 — λ_4

0.041 — λ_5

0.062 — y

0.158 — Z

α — Stream No.

Latest Forecast of λ_i

Old Forecast of λ_i

0.0-0.30.9001729

Clock Time	Stock(1)	Transit(1)	Stock(2)	Transit(2)	Stock(3)	Transit(3)	Stock(4)	Transit(4)	Stock(5)	Transit(5)	Central Store Stock
0	4	0	6	0	9	0	16	0	24	0	5

New Forecast λ_7 → 0.164 0.068 0.062 $i = 5$

```
  2     4    0    6    0    9    0   16    0   23    0    5
0.171 0.075 0.068    5
  4     4    0    6    0    9    0   16    0   22    0    5
0.174 0.045 0.041    4
  6     4    0    6    0    9    0   15    0   22    0    5
0.175 0.076 0.075    5
 15     4    0    6    0    9    0   15    0   21    0    5
0.178 0.047 0.045    4
 17     4    0    6    0    9    0   14    0   21    0    5
0.179 0.026 0.025    3
 20     4    0    6    0    8    0   14    0   21    0    5
0.180 0.018 0.016    2
 22     4    0    5    0    8    0   14    0   21    0    5
0.183 0.050 0.047    4
 28     4    0    5    0    8    0   13    0   21    0    5
0.184 0.019 0.018    2
 37     4    0    4    0    8    0   13    0   21    0    5
0.186 0.021 0.019    2
 41     4    0    3    0    8    0   13    0   21    0    5
0.186 0.050 0.050    4
 47     4    0    3    0    8    0   12    0   21    0    5
0.173 0.063 0.076    5
 55     4    0    3    0    8    0   12    0   20    0    5
0.173 0.026 0.026    3
 58     4    0    3    0    7    0   12    0   20    0    5
0.176 0.029 0.026    3
 61     4    0    3    0    6    0   12    0   20    0    5
0.174 0.061 0.063    5
 76     4    0    3    0    6    0   12    0   19    0    5
0.174 0.011 0.011    1
 81     3    0    3    0    6    0   12    0   19    0    5
0.170 0.046 0.050    4
 82     3    0    3    0    6    0   11    0   19    0    5
0.171 0.029 0.029    3
 88     3    0    3    0    5    0   11    0   19    0    5
0.174 0.049 0.046    4
 90     3    0    3    0    5    0   10    0   19    0    5
0.179 0.054 0.049    4
 93     3    0    3    0    5    0    9    0   19    0    5
0.175 0.057 0.061    5
103     3    0    3    0    5    0    9    0   18    0    5
0.176 0.031 0.029    3
107     3    0    3    0    4    0    9    0   18    0    5
0.179 0.034 0.031    3
```

```
110    3   0   3   0   3   0   9   0   18   0   5
0.180  0.058  0.057   5
118    3   0   3   0   3   0   9   0   17   0   5
0.186  0.064  0.058   5
120    3   0   3   0   3   0   9   0   16   0   5
0.184  0.019  0.021   2
129    3   0   2   0   3   0   9   0   16   0   5
0.179  0.049  0.054   4
131    3   0   2   0   3   0   8   0   16   0   5
0.180  0.021  0.019   2
138    3   0   1   0   3   0   8   0   16   0   5
138    3   0   1   2   3   0   8   0   16   0   3
0.177  0.060  0.064   5
145    3   0   1   2   3   0   8   0   15   0   3
0.178  0.050  0.049   4
146    3   0   1   2   3   0   7   0   15   0   3
0.177  0.032  0.034   3
152    3   0   1   2   2   0   7   0   15   0   3
152    3   0   1   2   2   2   7   0   15   0   1
0.180  0.053  0.050   4
154    3   0   1   2   2   2   6   0   15   0   1
```

Clock= 154   1 ← Np

```
0.183  0.035  0.032   3
158    3   0   1   2   1   2   6   0   15   0   1
0.184  0.022  0.021   2
160    3   0   0   2   1   2   6   0   15   0   1
160    3   0   0   3   1   2   6   0   15   0   0
0.188  0.057  0.053   4
160    3   0   0   3   1   2   5   0   15   0   0
163    3   0   2   1   1   2   5   0   15   0   0
0.190  0.024  0,022   2
163    3   0   1   1   1   2   5   0   15   0   0
0.188  0.057  0.060   5
169    3   0   1   1   1   2   5   0   14   0   0
0.189  0.037  0.035   3
173    3   0   1   1   0   2   5   0   14   0   0
177    3   0   1   1   2   0   5   0   14   0   0
0.193  0.040  0.037   3
178    3   0   1   1   1   0   5   0   14   0   0
0.197  0.044  0.040   3
180    3   0   1   1   0   0   5   0   14   0   0
0.198  0.026  0.024   2
182    3   0   0   1   0   0   5   0   14   0   0
185    3   0   1   0   0   0   5   0   14   0   0
```

0.198	0.057	0.057	5								
186	3	0	1	0	0	0	5	0	13	0	0
0.204	0.063	0.057	5								
188	3	0	1	0	0	0	5	0	12	0	0
0.211	0.070	0.063	5								
188	3	0	1	0	0	0	5	0	11	0	0
0.213	0.047	0.044	3								
190	3	0	1	0	-1	0	5	0	11	0	0
0.209	0.052	0.057	4								
193	3	0	1	0	-1	0	4	0	11	0	0
0.211	0.028	0.026	2								
195	3	0	0	0	-1	0	4	0	11	0	0
0.214	0.051	0.047	3								
197	3	0	0	0	-2	0	4	0	11	0	0
0.217	0.054	0.051	3								
205	3	0	0	0	-3	0	4	0	11	0	0
0.213	0.066	0.070	5								
212	3	0	0	0	-3	0	4	0	10	0	0
0.211	0.051	0.052	4								
218	3	0	0	0	-3	0	3	0	10	0	0
0.213	0.055	0.054	3								
218	3	0	0	0	-4	0	3	0	10	0	0
0.216	0.069	0.066	5								
220	3	0	0	0	-4	0	3	0	9	0	0
0.216	0.028	0.028	2								
224	3	0	-1	0	-4	0	3	0	9	0	0
0.220	0.054	0.051	4								
225	3	0	-1	0	-4	0	2	0	9	0	0
0.223	0.057	0.054	4								
233	3	0	-1	0	-4	0	1	0	9	0	0
0.223	0.069	0.069	5								
234	3	0	-1	0	-4	0	1	0	8	0	0
0.229	0.063	0.057	4								
235	3	0	-1	0	-4	0	0	0	8	0	0
0.228	0.054	0.055	3								
239	3	0	-1	0	-5	0	0	0	8	0	0
0.226	0.067	0.069	5								
253	3	0	-1	0	-5	0	0	0	7	0	0
254	3	0	-1	8	-5	0	0	0	7	0	52
254	3	0	-1	8	-5	19	0	0	7	0	33
254	3	0	-1	8	-5	19	0	16	7	0	17
0.226	0.054	0.054	3								
258	3	0	-1	8	-6	19	0	16	7	0	17
0.224	0.010	0.011	1								

261	2	0	-1	8	-6	19	0	16	7	0	17
0.219	0.058	0.063		4							
264	2	0	-1	8	-6	19	-1	16	7	0	17
0,217	0.064	0.067		5							
274	2	0	-1	8	-6	19	-1	16	6	0	17
279	2	0	7	0	-6	19	-1	16	6	0	17
279	2	0	7	0	13	0	-1	16	6	0	17
279	2	0	7	0	13	0	15	0	6	0	17
0.215	0.052	0.054		3							
284	2	0	7	0	12	0	15	0	6	0	17
0.213	0.057	0.058		4							
286	2	0	7	0	12	0	14	0	6	0	17
0.217	0.056	0.052		3							
289	2	0	7	0	11	0	14	0	6	0	17
0.221	0.061	0.057		4							
292	2	0	7	0	11	0	13	0	6	0	17
0.225	0.064	0.061		4							
299	2	0	7	0	11	0	12	0	6	0	17
0.222	0.025	0.028		2							
302	2	0	6	0	11	0	12	0	6	0	17
0.224	0.027	0.025		2							
307	2	0	5	0	11	0	12	0	6	0	17
0.217	0.057	0.064		5							
308	2	0	5	0	11	0	12	0	5	0	17
0.220	0.067	0.064		4							
308	2	0	5	0	11	0	11	0	5	0	17
0.216	0.052	0.056		3							
320	2	0	5	0	10	0	11	0	5	0	17
0.217	0.058	0.057		5							
322	2	0	5	0	10	0	11	0	4	0	17
322	2	0	5	0	10	0	11	0	4	8	9
0.218	0.029	0.027		2							
330	2	0	4	0	10	0	11	0	4	8	9
0.220	0.030	0.029		2							
344	2	0	3	0	10	0	11	0	4	8	9
347	2	0	3	0	10	0	11	0	12	0	9
0.217	0.055	0.058		5							
349	2	0	3	0	10	0	11	0	11	0	9
0.221	0.059	0.055		5							
354	2	0	3	0	10	0	11	0	10	0	9
0.209	0.055	0.067		4							
356	2	0	3	0	10	0	10	0	10	0	9
0.209	0.010	0.010		1							
357	1	0	3	0	10	0	10	0	10	0	9

0.214	0.060	0.055	4								
358	1	0	3	0	10	0	9	0	10	0	9
0.213	0.059	0.059	5								
373	1	0	3	0	10	0	9	0	9	0	9
0.203	0.042	0.052	3								
385	1	0	3	0	9	0	9	0	9	0	9
0.198	0.055	0.060	4								
389	1	0	3	0	9	0	8	0	9	0	9
0.192	0.053	0.059	5								
409	1	0	3	0	9	0	8	0	8	0	9
0.198	0.059	0.053	5								
409	1	0	3	0	9	0	8	0	7	0	9
0.195	0.027	0.030	2								
416	1	0	2	0	9	0	8	0	7	0	9
416	1	0	2	2	9	0	8	0	7	0	7
0.198	0.062	0.059	5								
417	1	0	2	2	9	0	8	0	6	0	7
0.205	0.069	0.062	5								
417	1	0	2	2	9	0	8	0	5	0	7
417	1	0	2	2	9	0	8	0	5	4	3
0.203	0.040	0.042	3								
420	1	0	2	2	8	0	8	0	5	4	3
441	1	0	4	0	8	0	8	0	5	4	3
0.199	0.064	0.069	5								
441	1	0	4	0	8	0	8	0	4	4	3
442	1	0	4	0	8	0	8	0	8	0	3
0.188	0.044	0.055	4								
450	1	0	4	0	8	0	7	0	8	0	3
0.189	0.066	0.064	5								
453	1	0	4	0	8	0	7	0	7	0	3
0.187	0.038	0.040	3								
457	1	0	4	0	7	0	7	0	7	0	3
0.187	0.027	0.027	2								
458	1	0	3	0	7	0	7	0	7	0	3
458	2										
458	1	1	3	0	7	0	7	0	7	0	2
0.188	0.067	0.066	5								
465	1	1	3	0	7	0	7	0	6	0	2
0.191	0.040	0.038	3								
469	1	1	3	0	6	0	7	0	6	0	2
0.192	0.069	0.067	5								
476	1	1	3	0	6	0	7	0	5	0	2
483	2	0	3	0	6	0	7	0	5	0	2
0.188	0.040	0.044	4								

498	2	0	3	0	6	0	6	0	5	0	2
0.187	0.039	0.040	3								
499	2	0	3	0	5	0	6	0	5	0	2
0.189	0.042	0.039	3								
511	2	0	3	0	4	0	6	0	5	0	2
0.191	0.044	0.042	3								
524	2	0	3	0	3	0	6	0	5	0	2
0.190	0.039	0.040	4								
526	2	0	3	0	3	0	5	0	5	0	2
0.176	0.055	0.069	5								
527	2	0	3	0	3	0	5	0	4	0	2
0.181	0.048	0.044	3								
527	2	0	3	0	2	0	5	0	4	0	2
0.186	0.061	0.055	5								
528	2	0	3	0	2	0	5	0	3	0	2
528	2	0	3	0	2	0	5	0	3	2	0
0.185	0.009	0.010	1								
530	1	0	3	0	2	0	5	0	3	2	0
0.190	0.052	0.048	3								
530	1	0	3	0	1	0	5	0	3	2	0
0.193	0.042	0.039	4								
533	1	0	3	0	1	0	4	0	3	2	0
0.196	0.045	0.042	4								
542	1	0	3	0	1	0	3	0	3	2	0
0.199	0.048	0.045	4								
551	1	0	3	0	1	0	2	0	3	2	0
553	1	0	3	0	1	0	2	0	5	0	0
558	1	0	3	0	1	16	2	0	5	0	44
558	1	0	3	0	1	16	2	14	5	0	30
558	1	0	3	0	1	16	2	14	5	15	15
0.196	0.049	0.052	3								
560	1	0	3	0	0	16	2	14	5	15	15
0.197	0.010	0.009	1								
565	0	0	3	0	0	16	2	14	5	15	15
565	0	3	3	0	0	16	2	14	5	15	12
0.198	0.049	0.048	4								
566	0	3	3	0	0	16	1	14	5	15	12
0.193	0.022	0.027	2								
575	0	3	2	0	0	16	1	14	5	15	12
0.182	0.049	0.061	5								
581	0	3	2	0	0	16	1	14	4	15	12
583	0	3	2	0	16	0	1	14	4	15	12
583	0	3	2	0	16	0	15	0	4	15	12
583	0	3	2	0	16	0	15	0	19	0	12

590	3	0	2	0	16	0	15	0	19	0	12
0.179	0.047	0.049	3								
591	3	0	2	0	15	0	15	0	19	0	12
0.181	0.023	0.022	2								
595	3	0	1	0	15	0	15	0	19	0	12
595	3	0	1	6	15	0	15	0	19	0	6
0.181	0.047	0.047	3								
612	3	0	1	6	14	0	15	0	19	0	6
620	3	0	7	0	14	0	15	0	19	0	6
0.183	0,049	0.047	3								
625	3	0	7	0	13	0	15	0	19	0	6
0.177	0.044	0.049	5								
625	3	0	7	0	13	0	15	0	18	0	6
0.168	0.041	0.049	4								
629	3	0	7	0	13	0	14	0	18	0	6
0.172	0.048	0,044	5								
630	3	0	7	0	13	0	14	0	17	0	6
0.170	0,046	0.048	5								
658	3	0	7	0	13	0	14	0	16	0	6
0.170	0.010	0.010	1								
660	2	0	7	0	13	0	14	0	16	0	6
0.167	0,045	0.049	3								
661	2	0	7	0	12	0	14	0	16	0	6
0.166	0.045	0.046	5								
683	2	0	7	0	12	0	14	0	15	0	6
0.162	0.036	0.041	4								
684	2	0	7	0	12	0	13	0	15	0	6
0.165	0.039	0.036	4								
689	2	0	7	0	12	0	12	0	15	0	6
0.162	0.020	0.023	2								
695	2	0	6	0	12	0	12	0	15	0	6
0.163	0.046	0.045	5								
700	2	0	6	0	12	0	12	0	14	0	6
0.160	0.042	0.045	3								
701	2	0	6	0	11	0	12	0	14	0	6
0.163	0.045	0.042	3								
705	2	0	6	0	10	0	12	0	14	0	6
0.166	0.048	0.045	3								
717	2	0	6	0	9	0	12	0	14	0	6
0.166	0.047	0.046	5								
718	2	0	6	0	9	0	12	0	13	0	6
0.169	0.050	0.048	3								
728	2	0	6	0	8	0	12	0	13	0	6
0.171	0.049	0.047	5								

731	2	0	6	0	8	0	12	0	12	0	6
0.171	0.050	0.050	3								
747	2	0	6	0	7	0	12	0	12	0	6
0.165	0.034	0.039	4								
756	2	0	6	0	7	0	11	0	12	0	6
0.165	0.050	0.050	3								
769	2	0	6	0	6	0	11	0	12	0	6
0.159	0.044	0.049	5								
775	2	0	6	0	6	0	11	0	11	0	6
0.162	0.052	0.050	3								
779	2	0	6	0	5	0	11	0	11	0	6
0.160	0.019	0.020	2								
784	2	0	5	0	5	0	11	0	11	0	6
0.163	0.055	0.052	3								
790	2	0	5	0	4	0	11	0	11	0	6
790	2	0	5	0	4	5	11	0	11	0	1
0.166	0.058	0.055	3								
798	2	0	5	0	3	5	11	0	11	0	1
0.172	0.064	0.058	3								
→ 798	2	0	5	0	2	5	11	0	11	0	1
798	3										
0.171	0.032	0.034	4								
800	2	0	5	0	2	5	10	0	11	0	1
0.172	0.020	0.019	2								
803	2	0	4	0	2	5	10	0	11	0	1
0.171	0.009	0.010	1								
808	1	0	4	0	2	5	10	0	11	0	1
808	1	1	4	0	2	5	10	0	11	0	0
0.172	0.065	0.064	3								
813	1	1	4	0	1	5	10	0	11	0	0
0.169	0.041	0.044	5								
814	1	1	4	0	1	5	10	0	10	0	0
0.173	0.045	0.041	5								
815	1	1	4	0	1	5	10	0	9	0	0
815	1	1	4	0	6	0	10	0	9	0	0
0.175	0.033	0.032	4								
817	1	1	4	0	6	0	9	0	9	0	0
0.178	0.048	0.045	5								
823	1	1	4	0	6	0	9	0	8	0	0
0.180	0.036	0.033	4								
824	1	1	4	0	6	0	8	0	8	0	0
0.181	0.021	0.020	2								
828	1	1	3	0	6	0	8	0	8	0	0
0.184	0.023	0.021	2								

APPENDIX FOURTEEN

PROGRAM DETAILS

Program 1: The C.S.L. Simulation of Model 1 with Control 1A (see 6.4.2, 6.8)

Program 2: The "Simon" Simulation of Model V with Control 10 (see 15.18, 15.20)

Program 3: The "Simon" Simulation of Model VI with Control 11 (see 16.4, 16.5)

Program 4: The "Simon" Simulation of Model VII with Cran's and author's best control policies (see Chapter 17)

Program 1

```
$*          MOUNT SCRATCH TAPE ON A5
$EXECUTE         CSL
$CSL        12
$ILCSL INVA        2
      CONTROL
      FUNCTION REAL SQRT
      CLASS TIME DELSS.5 DEM.5
      INTEGER STREAMA
      INTEGER V
      REAL A
      REAL TF R B E SPROD SVAR SK SKS              SIGMA(200) VAR(200)
      INTEGER J S(5) FRSTK
      INTEGER RMAX N(5)      TOTSO(5)
      REAL MEAN(5,200) DEVN D(200) NU DD
      INTEGER RVAL(5,200) U(5,200) W(5,200) USGE(5,200)
      INTEGER        TOTBO BORDS(5) VSS X T    BOR
      REAL DESS AVSTK SD F SUMP C Z  PP(5) BESTVAR
      REAL AREA
      INTEGER ATOTBO
      INTEGER STOCK(5) M SYSTK Q ROLSS CWSTK L LSS         TOT AMT(5)
      REAL NOPRC REPTO TCOST CBO
      INTEGER REPS(5) SHARE(5) TRSTK        G
      TIME FIN SYPROC SYPRCA SYPRCB
      TIME BO(5,30) EVENT
      INITL
      ACTIVITIES
      CWDD DSS SYRO SSRO SSDD RESU
      END
```

```
$ILCSL SRGB       2
      SECTOR INITL
      Q=60
      T.FIN=2500
      C=.4
      L=100
      DESS=.04
      READ(5,7)M,Z,STREAMA
    7 FORMAT(I3,F4.2,I5)
      WRITE(6,STD)M,Z,STREAMA
      TF=T.FIN
      SYSTK=35+M
      VSS=35+M
      ROLSS=-1
      FOR J=1,5
        FOR K=1,200
          USGE(J,K)=0
          RVAL(J,K)=0
          W(J,K)=0
          U(J,K)=0
          MEAN(J,K)=0
      FOR K=1,5
        N(K)=0
        TOTSO(K)=0
        STOCK(K)=5
        T.DEM.K=NEGEXP(25,STREAMA)
      CWSTK=M+10
      LSS=0
      T.SYPROC =-1
      T.EVENT=0
      DD=0
      AVSTK=0
      NOPRC=0
      G=0
      AREA= 0
      ATOTBO=0
      CBO=0
      FOR J=1,5
        REPS(J)=0
        AMT(J)=0
        BORDS(J) GT 0,73
        BORDS(J)=0
        FOR K=1,100
          T.BO(J,K) =0
   73   DUMMY
      TOTBO=0
      TCOST=0
      REPTO=0
      TRSTK=0
      END
```

```
$ILCSL CWDD         2
       SECTOR CWDD
         EQ T.SYPROC
     5 NOPRC+1
       CALL CALCG
       AREA-T.EVENT*G
       T.EVENT=0
       SYSTK+Q
       CWSTK+Q
       END
```

```
$ILCSL DSS        2
     SECTOR DSS
     FOR J=1,5
  9    O GE T.DEM.J'23
       CALL CALCG
       AREA-T.EVENT*G
       T.EVENT=0
       STOCK (J)-1
       VSS-1
       SYSTK-1
       O GT STOCK (J)'24
       BORDS(J)+1
       TOTBO+1
       BOR=BORDS(J)
       T.BO(J,BOR)=0
 24    T.DEM.J=NEGEXP(25,STREAMA)
       GO TO 9
 23    DUMMY
   1 LINGEN
    1DEMAND AT SS
     END
```

```
$ILCSL SYRO        2
      SECTOR SYRO
      CALL SFRST
      M GE FRSTK
   99 O GE T.SYPROC
      T.SYPROC=L
    9 VSS+Q
   10 END
```

```
$ILCSL SSRO      2
     SECTOR SSRO
     CWSTK GT 0
     ATOTBO=TOTBO
     FOR J=1,5
       CWSTK GT 0
       ROLSS GE STOCK (J) '6
       X=CWSTK-ATOTBO
       0 GE X'2
       BORDS(J) GE CWSTK '91
       AMT(J)=CWSTK
       GO TO 5
  91   BORDS(J) GT 0'6
   9   AMT(J)=BORDS(J)
       GO TO 5
   2   T.SYPROC GT 0'51
  61   CALL SHARE
       AMT(J)=BORDS(J)+SHARE(J)
       GO TO 5
  51   CALL CALCV
       X GE V'8
       AMT(J)=BORDS(J)+V
       GO TO 5
   8   AMT(J)=BORDS(J)+X
   5   AMT(J) GT 0'6
       CWSTK-AMT(J)
       TRSTK+AMT(J)
       AMT(J) GE BORDS(J)'55
       ATOTBO-BORDS(J)
       GO TO 11
  55   ATOTBO-AMT(J)
  11   T.DELSS.J=LSS
   6   DUMMY
   1   LINGEN
    1 ROL AT A SS
     END
```

```
$ILCSL SSDD        2
      SECTOR SSDD
      FOR J=1,5
        AMT(J) GT 0'20
        O EQ T.DELSS.J'20
        AC=3
        REPS(J)+1
        REPTO+1
        BORDS(J) GT 0'6
        CALL CALCG
        AREA-T.EVENT*G
        T.EVENT=0
        BORDS(J) GT AMT (J)'5
        BORDS(J)-AMT(J)
        TOTBO-AMT(J)
        FOR K=1,AMT(J)
          CBO-C*T.BO(J,K)
          T.BO(J,K)=0
        FOR I=1,BORDS(J)
          Y=I+AMT(J)
          T.BO(J,I)=T.BO(J,Y)
          T.BO(J,Y)=0
        GO TO 6
    5   FOR K=1,BORDS(J)
          CBO-C*T.BO(J,K)
          T.BO(J,K)=0
        TOTBO-BORDS(J)
        BORDS(J)=0
    6   TOTSO(J)+AMT(J)
        STOCK(J)+AMT(J)
        TRSTK-AMT(J)
        AMT(J)=0
   20   DUMMY
      AC EQ 3'46
      AC=0
      GO TO SYRO
   46 END
```

```
$ILCSL RESU        2
      SECTOR RESU
        GE T.FIN
      CALL CALCG
      AREA-T.EVENT*G
      FOR J=1,5
        BORDS(J) GT 0,7
        FOR K=1,BORDS(J)
          CBO-C*T.BO(J,K)
   7    DUMMY
      AVSTK=AREA/TF
      TCOST=.5*NOPRC+.3*REPTO+CBO+AVSTK
      WRITE(6,STD)NOPRC,REPTO,CBO,CLOCK
      WRITE(6,STD)AVSTK,TCOST,Z,M
      GO TO INITL
      END
```

```
$ILCSL CALCG        2
      SUBROUTINE CALCG
      SUMS=0
      FOR K=1,5
        STOCK(K) GT 0,37
        SUMS+STOCK(K)
   37   DUMMY
      G=SUMS+CWSTK+TRSTK
      RETURN
      END
```

```
$ILCSL CALCV       2
      SUBROUTINE CALCV
      CALL SFRST
      R=FRSTK-M
      N(J)+1
      FF=N(J)
      SD=SQRT(4+.24*R)
      MEAN(J,FF)=.2*R+4
    2 W(J,FF)=TOTSO(J)
      U(J,FF)=STOCK(J)
      RVAL(J,FF)=R
      V=MEAN(J,FF)+Z*SD
        GT V,5
      V=0
    5 RETURN
      END
```

```
$ILCSL SHARE      2
      SUBROUTINE SHARE
      CALL SFRST
      SHARE(J)=FRSTK/5
      SHARE(J) EQ 0,4
      X GT 0,4
      SHARE(J)=1
    4 0 GT SHARE(J),5
      SHARE(J)=0
    5 RETURN
      END
```

```
$ILCSL SFRSTK      2
      SUBROUTINE SFRST
      FOR K=1,5
        S(K)=STOCK(K)
      H=1
      CB=0
    4 S(H) GT S(H+1)'1
      B=S(H+1)
      S(H+1)=S(H)
      S(H)=B
      CB=1
    1 H GE 4'2
      CB EQ 1'3
      H=1
      CB=0
      GO TO 4
    2 H+1
      GO TO 4
    3 RR=CWSTK+TRSTK
      S(1) EQ S(2)'5
      S(1) EQ S(3)'6
      S(1) EQ S(4)'7
      S(1) EQ S(5)'8
      H=5
      GO TO 9
    5 H=1
      GO TO 9
    6 H=2
      GO TO 9
    7 H=3
      GO TO 9
    8 H=4
    9 RR GT 0'20
      S(H)+1
      RR-1
      RR GT 0'20
      H GT 1'10
      H-1
      GO TO 9
   10 S(1) EQ S(2)'11
      S(2) EQ S(3)'6
      S(3) EQ S(4)'7
      S(4) EQ S(5)'8
      H=5
   11 GO TO 9
   20 Y=S(1)+1
      S(2) GT Y'12
      S(2)=Y
   12 S(3) GT Y'13
      S(3)=Y
   13 S(4) GT Y'14
      S(4)=Y
   14 S(5) GT Y'15
      S(5)=Y
   15 FRSTK=S(1)+S(2)+S(3)+S(4)+S(5)
      RETURN
      END
```

```
$EOF
$IEDIT          SYSCK2.SRCH
$IBMAP INVA     NOLIST,MFTC,NOREF
$IBMAP SRGB     NOLIST,MFTC,NOREF
$IBMAP CWDD     NOLIST,MFTC,NOREF
$IBMAP DSS      NOLIST,MFTC,NOREF
$IBMAP SYRO     NOLIST,MFTC,NOREF
$IBMAP SSRO     NOLIST,MFTC,NOREF
$IBMAP SSDD     NOLIST,MFTC,NOREF
$IBMAP RESU     NOLIST,MFTC,NOREF
$IBMAP CALCG    NOLIST,MFTC,NOREF
$IBMAP CALCV    NOLIST,MFTC,NOREF
$IBMAP SHARE    NOLIST,MFTC,NOREF
```

Progam 2

```
// FOR
*IOCS(CARD,1132 PRINTER,DISK)
      REAL L,MU,LAM
      INTEGER S,T
      COMMON SIG(10), C(10,100),D(10,100),CO(10),
     1 S,MU,L,BETA,LAM,CR,T
      LAM=.04
      L=50
      MU=LAM*L
      BETA=EXP(-MU)
      CR=.3
      DEN=1.
      A=1.
      DO 2 S=1,10
      FS=FLOAT(S)
      DEN=DEN*FS
      A=A+MU**FS/DEN
      SIG(S)=A
    2 CONTINUE
      FS=1.
      CO(1)=.5*MU*L*(1.-BETA)-  L*(1.-BETA*SIG(1))+.5*FS*(FS+1.)*(1.-BET
     1A*SIG(2))/LAM
      CO(1)=.4*CO(1)
      HA      =CO(1)+CR
      HB      =(1.-LAM)*CO(1)+.5*MU*L*LAM*.4
      IF (HB-HA) 77,77,52
   77 D(1,1)=0.
      C(1,1)=HB
      GO TO 42
   52 D(1,1)=1.
      C(1,1)=HA
   42 DO 44 T=2,100
      S=1
      HA      =CO(S)+CR
      HB      =(1.-LAM)*C(S,T-1)+LAM*(CR+.2*LAM*L*L)
      IF (HB-HA) 78,78,56
   78 D(S,T)=0
      C(S,T)=HB
      GO TO 47
   56 D(S,T)=1.
      C(S,T)=HA
   47 WRITE(3,1)S,T,D(S,T)
   44 CONTINUE
      CALL SUBX
    1 FORMAT(7(I3,I4,F2.0,4X))
      CALL EXIT
      END
```

```
// DUP
*STORE      WS  UA  NNNN
// FOR
      SUBROUTINE SUBX
      REAL L,MU,LAM
      INTEGER S,T
      COMMON SIG(10), C(10,100),D(10,100),CO(10),
     1 S,MU,L,BETA,LAM,CR,T
      DO 3 S=2,10
      FS=FLOAT(S)
      F=FLOAT(S)
      CO(S) =.5*MU*L*(1.-BETA*SIG(S-1))-F*L*(1.-BETA*SIG(S))+.5*FS*(FS+1.)+
     1.)*(1.-BETA*SIG(S+1))/LAM
      CO(S)=.4*CO(S)
      HA     =CO(S)+CR
      HB     =(1.-LAM)*CO(S)+LAM*CO(S-1)
      IF (HB-HA)  53,53,51
   53 D(S,1)=0.
      C(S,1)=HB
      GO TO 41
   51 D(S,1)=1.
      C(S,1)=HA
   41 DO 4 T=2,100
      HA     =CO(S)+CR
      HB     =(1.-LAM)*C(S,T-1)+LAM*C(S-1,T-1)
      IF (HB-HA) 8,8,5
    8 D(S,T)=0.
      C(S,T)=HB
      GO TO 4
    5 D(S,T)=1.
      C(S,T)=HA
    4 CONTINUE
      WRITE(3,1)(S,T,D(S,T),T=1,100)
    3 CONTINUE
    1 FORMAT(7(I3,I4,F2.0,4X))
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SUBX

// FOR
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER,DISK)
*EXTENDED PRECISION
      DEFINE FILE 1(1,120,U,IUP)
      REAL LAM,MU
      INTEGER DA(120)
      DIMENSION SIG(11)
      COMMON C(10,120),M,CMAL(120),ALPHA,CR,CO(10),DA,Z
      READ(2,88)M,Z,ISTRE
   88 FORMAT(I2,F3.1,I4)
      LAM=.04
      CR=.3
      LSS=50
      U=LSS
      MU=LAM*LSS
      DEN=1.
      A=1.
      BETA=EXP(-MU)
      BE=BETA
      DO 2 K=1,11
      DEN=DEN*K
      REAG=MU**K/DEN
  231 FORMAT(F12.6)
      A=A+REAG
      SIG(K)=A
    1 FORMAT(F12.6)
    2 CONTINUE
      CO(1)=.4*(.5*MU*U*(1.-BETA)-U*(1.-BETA*SIG(1))+(1.-BETA*SIG(2))/
     1LAM)
      ALPHA=CR+.5*MU*.4*U
      DO 93 K=2,10
      CO(K)=.5*MU*U*(1.-BE*SIG(K-1) )- K*U*(1.-BETA*SIG(K))+.5* K*( K+1.
     1)*(1.-BETA*SIG(K+1))/LAM
      CO(K)=.4*CO(K)
   93 CONTINUE
      DO 98 K=1,6
      C(K,M)=CO(K)+CR
   98 CONTINUE
      CALL DSJG
      CALL SUBC
      CALL SUBA
      WRITE(1'1) DA
      READ(1'1) DA
      WRITE(3,2132) DA
 2132 FORMAT(40I3/)
      CALL EXIT
      END
```

```
// DUP
*STORE      WS  UA  DSJF
// FOR
*EXTENDED PRECISION
* ONE WORD INTEGERS
      SUBROUTINE DSJG
      INTEGER W,DA(120)
      COMMON C(10,120),M,CMAL(120),ALPHA,CR,CO(10),DA,Z
      DO 95 K=6,10
      SGA=0
      BOT=1.
      PTOT=0
      JA=K-6
      DO 76 IA=1,JA
      I=IA-1
      IF (I) 69,69,7
    7 BOT=BOT*I
      TGA=SGA
      SGA=SGA+(1.36**I)/BOT
      GO TO 70
   69 SGA=1.
      TGA=0
   70 VA=EXP(-1.36)*(SGA-TGA)
      SGB=0
      COT=1.
      JB=K-I-5
      DO 75 KW=1,JB
      W=KW-1
      IF (W) 80,80,8
    8 COT=COT*W
      TGB=SGB
      SGB=SGB+(1.28**W)/COT
      GO TO 90
   80 SGB=1.
      TGB=0
   90 VB=EXP(-1.28)*(SGB-TGB)
      SGC=0
      DOT=1
      JC=K-I-W-4
      DO 84 NA=1,JC
      N=NA-1
      IF (N) 81,81,9
    9 DOT=DOT*N
      TGC=SGC
      SGC=SGC+(0.60**N)/DOT
      GO TO 99
   81 SGC=1.
      TGC=0
   99 VC=EXP(-.6)*(SGC-TGC)
      SGD=0.
      EOT=1
      JD=K-I-W-N-3
      DO 73 KZ=1,JD
      KA=KZ-1
      IF (KA) 82,82,10
   10 EOT=EOT*KA
      TGD=SGD
      SGD=SGD+(0.20**KA)/EOT
      GO TO 97
```

```
   82 SGD=1.
      TGD=0
   97 VD= EXP(-0.20)*(SGD-TGD)
      SGE=0
      FOT=1.
      KE=K-I-W-N-KA-2
      DO 72 KX=1,KE
      KB=KX-1
      IF (KB) 74,74,11
   11 FOT=FOT*KB
      TGE=SGE
      SGE=SGE+(+0.08**KB)/FOT
      GO TO 100
   74 SGE=1
      TGE=0
  100 VE=EXP(-.08)*(SGE-TGE)
      SGF=0
      GOT=1
      KF=K-I-W-N-KA-KB-1
      DO 71 KY=1,KF
      KC=KY-1
      IF (KC) 65,65,12
   12 GOT=GOT*KC
      TGF=SGF
      SGF=SGF+(.04**KC)/GOT
      GO TO 101
   65 SGF=1
      TGF=0
  101 VF=EXP(-.04)*(SGF-TGF)
      PTOT=PTOT+VA*VB*VC*VD*VE*(1-SGF)
      SGF=EXP(-.04)*SGF
   71 CONTINUE
   72 CONTINUE
      SGE=EXP(-0.08)*SGE
      PTOT=PTOT+VA*VB*VC*VD*(1.-SGE)
   73 CONTINUE
      SGD=EXP(-0.20)*SGD
      PTOT=PTOT+VA*VB*VC*(1.-SGD)
   84 CONTINUE
      SGC=EXP(-0.60)*SGC
      PTOT=PTOT+VA*VB*(1.-SGC)
   75 CONTINUE
      SGB=EXP(-1.28)*SGB
      PTOT=PTOT+VA*(1.-SGB)
   76 CONTINUE
      SGA=EXP(-1.36)*SGA
      PTOT=PTOT+1.-SGA
      C(K,M)=CO(K)+CR*PTOT
      WRITE(3,300) K,CO(K),PTOT
  300 FORMAT(I3,F12.6,F12.6)
   95 CONTINUE
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  DSJG
// FOR
*LIST ALL
*ONE WORD INTEGERS
*EXTENDED PRECISION
      SUBROUTINE SUBA
      INTEGER DA(120)
      COMMON C(10,120),M,CMAL(120),ALPHA,CR,CO(10),DA,Z
      CMAL(M)=0.
      MB=M+1
      MA=120
   38 DO 74 N=MB,MA
      KOU=0
   16 IF (5.*(C(1,M)+CMAL(N))-ALPHA-CMAL(N-1)-4.*C(1,N-1)) 19,19,20
   19 C(1,N)=C(1,M)+CMAL(N)
      GO TO 25
   20 C(1,N)=.2*CMAL(N-1)+.2*ALPHA+.8*C(1,N-1)
   25 DO 21 K=2,10
      IF (CO(K)+CR+CMAL(N)-.2*C(K-1,N-1)-.8*C(K,N-1)) 22,22,23
   22 C(K,N)=CO(K)+CR+CMAL(N)
      GO TO 21
   23 C(K,N)=.2*C(K-1,N-1)+.8*C(K,N-1)
      IF (KOU) 444,444,21
  444 DA(N)=K-1
      KOU=2000
   21 CONTINUE
      IF (KOU) 7,7,74
    7 DA(N)=10
   74 CONTINUE
      WRITE(3,300) (DA(N),N=MB,MA)
  300 FORMAT(30I4)
  972 CONTINUE
      RETURN
      END
```

```
// DUP
*STORE       WS  UA  SUBA
// FOR
* ONE WORD INTEGERS
*EXTENDED PRECISION
      SUBROUTINE SUBC
      INTEGER DA(120)
      INTEGER A,H
      COMMON C(10,120),M,CMAL(120),ALPHA,CR,CO(10),DA,Z
      MB=120
      MA=M+1
      DO 16 H=MA,MB
      A=.2*(H-M)+6.5+Z*SQRT(.24*(H-M)+6)
    4 IF (A-.2*M) 1,1,3
    1 CMAL(H)=0
      GO TO 16
    3 K=H-A-.8*M+1
      IF (K) 8,8,9
    8 CMAL(H)=.15*(H-M)
      GO TO 16
    9 U=A-.2*M
      R=.8
      Q=.2
      IF(U-1) 10,70,11
   70 CMAL(H)=.15*R**K
      GO TO 16
   10 CMAL(H)=0
      GO TO 16
   11 STG=0.
      SHRK=1.
      JF=K-1
      DO 13 N=1,JF
      SHRK=SHRK*FLOAT(N)
   13 CONTINUE
      JG=U-1
      DO 12 I=1,JG
      SHR=1.
      JH=K+I-1
      DO 14 N=1,JH
      SHR=SHR*FLOAT(N)
   14 CONTINUE
      SH=1.
      DO 15 N=1,I
      SH=SH*FLOAT(N)
   15 CONTINUE
      STG=STG+(U-I)*SHR*(R**K)*(Q**I)/(SHRK*SH)
   12 CONTINUE
      CMAL(H)=.15*(U*(R**K)+STG)
   16 CONTINUE
      WRITE(3,300) (CMAL(H),H=MA,MB)
  300 FORMAT(10F12.6)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SUBC

// FOR
*LIST ALL
*ONE WORD INTEGERS
*IOCS(TYPEWRITER)
*IOCS(CARD,1132 PRINTER,DISK)
      DEFINE FILE 1(1,120,U,IUP)
      REAL MU,LAM,LAMTO,AREA
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120),DEL(25,2),RR,TTG,DOT,W
      INTEGER R
      INTEGER     BACKO(14,2),FIN(2),      COUNT,EVENT(2),      REPTO,Q,
     1TFIN
      INTEGER REFN,TIMV,MEMN
      DIMENSION NUM(5)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK    ,TRS    ,L,ONORD,M,SYPRO    ,CWSTK,ISTRE,DEM
     1  ,G,S    ,LAM,B,MU,CR,KOUNT,CO(10),BORDS    ,ND(5),TR        ,NOTON,
     2          T,LSS,ALPHA,DA            ,NOPRC     ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
  658 FORMAT(I6,F12.4,2I8)
  848 READ(2,88) M,  Z,ISTRE
   88 FORMAT(I2,F6.2,I4)
      WRITE(1,88) M,Z,ISTRE
      READ (1'1) DA
 2132 FORMAT(40I3/)
      WRITE(3,2132) DA
      CALL SIMO
 2145 FORMAT(' M=',I4,'    Z=',F12.6,' ISTRE=',I6)
      CALL GROU(BACKO,14,6)
      CALL GROU (DEM,5,1)
      CALL GROU(DEL,25,4)
      AREA=0
      CBO=0
      Q=75
      KKK=0
      LSS=50
      L=100
      LAM=.04
      CR=.3
      G=0
      NOPRC=0
      REPTO=0
      NOTON=0
      NE=0
      ONORD=1
      VC=12
      NDEL=0
      R=9999
      CALL ENTI(SYPRO,2)
      CALL ENTI(EVENT,7)
      CALL ENTI(FIN,3)
      DO 78 K=1,14
      BACKO(K,STATE)=1000
   78 CONTINUE
      TFIN=1000
```

```
      CLOCK=0
      SYPRO(STATE)=NOTON
      CALL SETT(FIN(NAME),TFIN)
      CALL ADDL  (FIN(NAME),TIMES)
      CALL SETT (EVENT(NAME),0)
      DO 79 J=1,5
      CALL GEN(KKK,KJH,INBO)
      STOCK(J)=5
      ND(J)=0
      TRS(J)=0
      NUM(J)=0
      DO 79 K=1,10
   79 TR(J,K)=0
      CWSTK=10+M
      RR=CWSTK
      DO 564 K=1,5
  564 S(K)=STOCK(K)+TRS(K)
      CALL SFRS
      GO TO 304
11111 CALL SCA (TIMES,MEMBE ,CLOCK)
      K=REFN  (MEMBE )
      CALL DELE  (MEMBE ,TIMES)
      CALL CALG
  500 AREA=AREA+G*(CLOCK-TIMV  (EVENT(NAME)))
  800 CALL SETT  (EVENT(NAME),CLOCK)
  116 GO TO (1,2,3,4),K
C                              A.PHASE
C                 DEMAND ON SUB-STORE
    1 J=MEMN  (MEMBE )
   69 STOCK(J)=STOCK(J)-1
      WRITE(3,175) CLOCK,STOCK(1),TRS(1),STOCK(2),TRS(2),STOCK(3),TRS(3)
     1,STOCK(4),TRS(4),STOCK(5),TRS(5),CWSTK
  910 IF (STOCK(J)) 10,1000,1000
C     BACKORDERS EXIST
   10 NB=NB+1
      IF (NB-15) 60,61,61
   61 NB=1
   60 CALL SETT(BACKO(NB,NAME),CLOCK)
      BACKO(NB,STATE)=J
 1000 CALL GEN(KKK,KJH,INBO)
      GO TO 4000
C                 DELIVERY AT CENTRAL STORE
    2 CWSTK=CWSTK+Q
      SYPRO(STATE)=NOTON
      VC=12
      KAB=917
      GO TO 4001
C                FINISH SIMULATION
    3 DO 720 K=1,5
      DO 720 N=1,14
      IF (BACKO(N,STATE)-K) 720,729,720
  729 CBO=CBO+.4*(CLOCK-TIMV  (BACKO  (N,NAME)))
  474 FORMAT(' CBO=',F12.6)
  720 CONTINUE
      CALL CALG
      AREA=AREA+G*(CLOCK-TIMV  (EVENT(NAME)))
      TCOST=.5*NOPRC+ CR*REPTO+CBO+.0004*AREA
      WRITE(1,262)NOPRC,REPTO,CBO,AREA,TCOST
  262 FORMAT(' NOPRC=',I4,' REPTO=',I4,' CBO=',F8.4,' AREA=',F14.0,' TCO
```

```
     1ST=',F12.4)
      GO TO 848
C        DELIVERY AT SUBSTORE
    4 NN=MEMN(MEMBE)
      J=DEL(NN,STATE)
      NUM(J)=NUM(J)+1
      K=NUM(J)
      IF (K-10) 91,104,91
  104 NUM(J)=0
   91 IF(STOCK(J)) 112,2000,2000
C     BACKORDERS EXIST
  112 COUNT   =0
      IF (TR(J,K      )+STOCK(J)) 26, 27,27
C     ALL BORDS AT THIS STORE CAN BE ELIMINATED
   27 LIM=-STOCK(J)
      GO TO 119
   26 LIM=TR(J,K)
  119 DO 120 N=1,14
      IF (BACKO   (N,STATE)-J) 120,29,120
   29 CBO=CBO+.4*(CLOCK-TIMV   (BACKO   (N,NAME)))
      BACKO(N,STATE)=1000
      COUNT=COUNT+1
      IF (COUNT   -LIM) 120,2000,2000
  120 CONTINUE
 2000 STOCK(J)=STOCK(J)+TR(J,K)
      TRS(J)=TRS(J)-TR(J,K)
      WRITE(3,175) CLOCK,STOCK(1),TRS(1),STOCK(2),TRS(2),STOCK(3),TRS(3)
     1,STOCK(4),TRS(4),STOCK(5),TRS(5),CWSTK
C                                C-PHASE
C     TEST FOR SUBSTORE ORDER OR SYSTEM ORDER
 4000 IF (SYPRO(STATE)-NOTON) 304,4001,304
 4001 CALL SYRO
  304 DO 777 J=1,5
  915 CALL SSRO
  497 IF (KAUNT-94)    777,338,339
  659 FORMAT(10I6)
  338 CALL PNOO
      RR=B
      IF (VC-12) 470,48,470
   48 IF (RR-F/5) 470,470,49
   49 RR=F/5
  470 GO TO 38
  339 CALL P001
      RR=B-STOCK(J)-TRS(J)
   38 IF (RR) 777,777,18
   18 IF (RR-CWSTK)       118,118,117
  117 RR=CWSTK
  118 ND(J)=ND(J)+1
  122 FORMAT(I6)
      KT=ND(J)
      IF (KT-10)  222,223,222
  223 ND(J)=0
  222 TR(J,KT)=RR
      TRS(J)=TRS(J)+RR
      REPTO=REPTO+1
 2148 FORMAT(10I6)
      NDEL=NDEL+1
      CWSTK=CWSTK-RR
      IF (NDEL-26) 55,56,55
```

```
   56 NDEL=1
   55 CALL SETT   (DEL(NDEL,NAME),(CLOCK+LSS))
      CALL ADDL   (DEL(NDEL,NAME),TIMES)
      DEL(NDEL,STATE)=J
      WRITE(3,175) CLOCK,STOCK(1),TRS(1),STOCK(2),TRS(2),STOCK(3),TRS(3)
     1,STOCK(4),TRS(4),STOCK(5),TRS(5),CWSTK
  175 FORMAT (12I5)
  777 CONTINUE
      VC=0
      GO TO 11111
      END
```

```
// DUP
*STORE      WS  UA  MMMM
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE CALCS(ISDA,ILDA,RLAMD)
      REAL MU,LAM,LAMTO
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120)          ,RR,TTG,DOT,W
      DIMENSION TIGM(60)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK    ,TRS    ,L,ONORD,M,SYPRO    ,CWSTK,ISTRE,DEM
     1  ,G,S    ,LAM,B,MU,CR,KOUNT,CO(10),BORDS    ,ND(5),TR      ,NOTON,
     2          T,LSS,ALPHA,DA              ,NOPRC     ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
  368 FORMAT(2F12.6)
      RMUDA=RLAMD*ILDA
      FT=EXP(-RMUDA)
      TM=FT
      SIGM=0
      ISV=ISDA+1
      ISW=ISDA-1
      DO 2 I=1,ISV
      TM=TM*RMUDA/FLOAT(I)
      SIGM=SIGM+TM
      TIGM(I)=1-(SIGM+FT)
    2 CONTINUE
      A1=RMUDA*.5*ILDA
      A2=-ISDA*ILDA
      A3=ISDA*(ISDA+1)*.5/RLAMD
      IF (ISDA-1) 49,42,40
   40 T1=TIGM(ISW)*A1
      GO TO 43
   42 T1=A1*(1-FT)
   43 T2=A2*TIGM(ISDA)
      T3=A3*TIGM(ISV)
      VA=(T1+T2+T3)*.4
      RETURN
   49 VA=.4*(-ISDA+.5*RMUDA)*ILDA
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  CALCS
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE CALG
      REAL MU,LAM,LAMTO
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120)            ,RR,TTG,DOT,W
      INTEGER SUMS
      COMMON NIM,TAILM,MAST1,MAST2         ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK    ,TRS    ,L,ONORD,M,SYPRO    ,CWSTK,ISTRE,DEM
     1  ,G,S    ,LAM,B,MU,CR,KOUNT,CO(10),BORDS    ,ND(5),TR        ,NOTON,
     2          T,LSS,ALPHA,DA          ,NOPRC     ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
      SUMS=0.
      DO 2 K=1,5
      IF (STOCK(K)) 2,2,8
    8 SUMS=SUMS+STOCK(K)
    2 CONTINUE
      G=SUMS+TRS(1)+TRS(2)+TRS(3)+TRS(4)+TRS(5) +CWSTK
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  CALG
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE CALKB(D)
      REAL MU,LAM,LAMTO
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120)           ,RR,TTG,DOT,W
      DIMENSION IEND(8)
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK    ,TRS    ,L,ONORD,M,SYPRO    ,CWSTK,ISTRE,DEM
     1   ,G,S    ,LAM,B,MU,CR,KOUNT,CO(10),BORDS    ,ND(5),TR        ,NOTON,
     2          T,LSS,ALPHA,DA              ,NOPRC     ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
      VB=0
      JC=0
      IBL=B-S(J)
      IW=S(J)
      IT=T
      IEND(6)=56
      IEND(5)=24
      IEND(4)=9
      IEND(3)=4
      IEND(2)=2
      IEND(1)=1
      IFL=1
      V=D/FLOAT(T-1)
    3 IF (V*(IT-IEND(IW))-IBL) 5,4,4
    5 IF (IW-IFL) 6,1,6
    6 IBL=IBL+1
      IW=IW-1
      GO TO 3
    4 IT=IT-FLOAT(IBL)/V+.5
      JC=JC+1
      IBL=(IFL+1)+(IT-1)*(FLOAT(B-IFL-1))/FLOAT(T-1)+.5-IW
      IF (IBL) 27,27,28
   27 IBL=1
   28 VB=VB+IBL
C         VB IS SUM OF LITTLE B
      IF (IW-IFL) 3,1,3
    1 RETURN
      END
```

```
// DUP
*STORE       WS  UA  CALKB
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE GEN(KKK,KJH,INBO)
      REAL MU,LAM,LAMTO
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120)          ,RR,TTG,DOT,W
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK    ,TRS    ,L,ONORD,M,SYPRO    ,CWSTK,ISTRE,DEM
     1   ,G,S    ,LAM,B,MU,CR,KOUNT,CO(10),BORDS    ,ND(5),TR        ,NOTON,
     2           T,LSS,ALPHA,DA          ,NOPRC     ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
  325 INT=ALOG(RANDY(ISTRE  ))*(-25.)+.5
 6666 INTA=CLOCK+INT
      CALL SETT   (DEM(J,NAME),INTA)
      CALL ADDL   (DEM(J,NAME),TIMES)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  GEN
// FOR
* ONE WORD INTEGERS
      SUBROUTINE SFRS
      REAL MU,LAM,LAMTO
      INTEGER H
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120)          ,RR,TTG,DOT,W
      COMMON NIM,TAILM,MAST1,MAST2          ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK     ,TRS     ,L,ONORD,M,SYPRO     ,CWSTK,ISTRE,DEM
     1    ,G,S     ,LAM,B,MU,CR,KOUNT,CO(10),BORDS     ,ND(5),TR         ,NOTON,
     2           T,LSS,ALPHA,DA               ,NOPRC      ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
      H=1
      CB=0
  914 IF (S(H)-S(H+1)) 4,4,67
   67 B=S(H+1)
      S(H+1)=S(H)
      S(H)=B
      CB=1.
    4 IF (H-4) 99,98,98
   98 IF (CB-1.) 97,96,97
   96 H=1
      CB=0
      GO TO 914
   99 H=H+1
      GO TO 914
   97 IF (S(1)-S(2)) 80,81,80
   81 IF (S(1)-S(3)) 82,83,82
   83 IF (S(1)-S(4)) 84,85,84
   85 IF (S(1)-S(5)) 86,87,86
   87 H=5
      GO TO 9
   80 H=1
      GO TO 9
   82 H=2
      GO TO 9
   84 H=3
      GO TO 9
   86 H=4
    9 IF (RR) 70,70,71
   71 S(H)=S(H)+1.
   70 RR=RR-1.
      IF (RR) 60,60,61
   61 IF (H-1) 50,50,51
   51 H=H-1
      GO TO 9
   50 IF (S(1)-S(2)) 40,41,40
   41 IF (S(2)-S(3)) 82,46,82
   46 IF (S(3)-S(4)) 84,78,84
   78 IF (S(4)-S(5)) 86,63,86
   63 H=5
   40 GO TO 9
   60 Y=S(1)+1.
      IF (S(2)-Y) 215,215,212
  212 S(2)=Y
  215 IF (S(3)-Y) 214,214,217
```

```
217 S(3)=Y
214 IF (S(4)-Y) 218,218,90
 90 S(4)=Y
218 IF (S(5)-Y) 55,55,53
 53 S(5)=Y
 55 F=S(1)+S(2)+S(3)+S(4)+S(5)
    RETURN
    END
```

```
// DUP
*STORE      WS  UA  SFRS
// FOR
*ONE WORD INTEGERS
      SUBROUTINE PNOO
      REAL MU,LAM,LAMTO
      INTEGER PSOQ
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120)          ,RR,TTG,DOT,W
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK    ,TRS    ,L,ONORD,M,SYPRO    ,CWSTK,ISTRE,DEM
     1   ,G,S    ,LAM,B,MU,CR,KOUNT,CO(10),BORDS    ,ND(5),TR        ,NOTON,
     2           T,LSS,ALPHA,DA              ,NOPRC    ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
      PSOQ=.2*(F-M)+6+Z*SQRT(.24*(F-M)+6)+.5-STOCK(J)-TRS(J)
   68 IF (PSOQ-CWSTK) 9,9,8
    8 PSOQ=CWSTK
    9 B=PSOQ
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  PNOO
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE P001
      U=LAM*(T-1)
      B=STOCK(J)+TRS(J)+1
   76 D=0
      PTM=EXP(-U)
    1 JC=0
      VB=0
   77 IST=0
      DO 33 K=1,5
      S(K)=STOCK(K)+TRS(K)
   33 IST=IST+S(K)
      IST=IST+CWSTK
      IF (D) 31,31,773
  773 CALL CALKB(D)
   31 RR=CWSTK-B+S(J)-VB
      DO 23 K=1,5
      IF (K-J) 23,23,28
   28 IK=K-1
      S(IK)=S(K)
   23 CONTINUE
      H=1
      CB=0
  914 IF (S(H)-S(H+1)) 4,4,67
   67 IB=S(H+1)
      S(H+1)=S(H)
      S(H)=IB
      CB=1.
    4 IF (H-3) 99,98,98
   98 IF (CB-1.) 97,96,97
   96 H=1
      CB=0
      GO TO 914
   99 H=H+1
      GO TO 914
    9 IF (RR) 70,70,71
  181 IF (S(1)-S(3)) 82,83,82
   83 IF (S(1)-S(4)) 84,85,84
   85 H=4
      GO TO 9
  180 H=1
      GO TO 9
   82 H=2
      GO TO 9
   84 H=3
   97 IF (S(1)-S(2)) 180,181,180
   71 S(H)=S(H)+1
   70 RR=RR-1.
      IF (RR) 91,91,61
   61 IF (H-1) 50,50,51
   51 H=H-1
      GO TO 9
   50 IF (S(1)-S(2)) 40,41,40
   41 IF (S(2)-S(3)) 82,46,82
   46 IF (S(3)-S(4)) 84,178,84
```

```
178 H=4
 40 GO TO 9
 91 VT=0
    DO 20 K=1,4
  7 ILDA=T+LSS
    ISDA=S(K)
    RLAMD=LAM
    CALL CALCS(ISDA,ILDA,RLAMD)
273 FORMAT(F12.6,2I8)
    VT=VT+VA
 20 CONTINUE
    VA=VT
183 TC=VA+JC*CR
    IF (D) 18,18,19
 18 SUM=TC*PTM
    SIGP=PTM
    GO TO 2
 19 PTM=U*PTM/D
    SIGP=SIGP+PTM
    SUM=SUM+TC*PTM
    IF (SIGP-.95) 2,3,3
  2 D=D+1
    GO TO 77
  3 IF (B-STOCK(J)-TRS(J)-1)       37,22,37
 22 SUMOL=SUM
 80 IF (B-STOCK(J)-TRS(J)-CWSTK) 75,75,81
 75 B=B+1
    GO TO 76
 37 IF (SUM-SUMOL) 22,81,81
 81 B=B-1
    IS=CWSTK
    DO 658 K=1,5
    S(K)=STOCK(K)+TRS(K)
658 IS=IS+S(K)
    IF (CWSTK-10) 400,400,700
400 FEXB=IS-T*LAM*5
    IEXB=FEXB/5+1
    IF (IEXB-B) 277,700,700
277 B=IEXB
700 RETURN
    END
```

```
// DUP
*STORE      WS  UA  P001
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE SSRO
      REAL MU,LAM,LAMTO
      INTEGER TIMV
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120)          ,RR,TTG,DOT,W
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK    ,TRS    ,L,ONORD,M,SYPRO     ,CWSTK,ISTRE,DEM
     1  ,G,S    ,LAM,B,MU,CR,KOUNT,CO(10),BORDS     ,ND(5),TR         ,NOTON,
     2          T,LSS,ALPHA,DA               ,NOPRC      ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
      IF (CWSTK) 66,66,72
   72 SJ=STOCK(J)+TRS(J)
      IF (SYPRO (STATE)-ONORD  ) 42,52,52
   42 IF (F-120) 44,44,43
   44 IF (SJ-DA(F)) 94,94,66
   94 KAUNT=94
      RETURN
   43 IF (SJ-10) 94,94,66
   52 T=TIMV(SYPRO(NAME))-CLOCK
      IF (SJ) 95,95,797
  797 IF (T-56) 1,2,2
    2 IF (SJ-6) 95,95,66
    1 IF (T-24) 4,3,3
    3 IF (SJ-5) 95,95,66
    4 IF (T-9) 7,5,5
    5 IF (SJ-4) 95,95,66
    7 IF (T-4) 9,8,8
    8 IF (SJ-3) 95,95,66
    9 IF (T-2) 11,10,10
   10 IF (SJ-2) 95,95,66
   11 IF (T-1) 6,12,12
   12 IF (SJ-1) 95,95,66
   95 KAUNT=95
 2148 FORMAT(10I3)
    6 RETURN
   66 KAUNT=0
      RETURN
      END
```

```
// DUP
// FOR
*STORE      WS  UA  SSRO
* ONE WORD INTEGERS
      REAL MU,LAM,LAMTO
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      INTEGER STOCK(5),F,TRS(5),ONORD,SYPRO(2),CWSTK,DEM( 5,2),G,S(5),B,
     1BORDS(5),TR(5,10),T,DA(120)          ,RR,TTG,DOT,W
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON J,F,Z,STOCK    ,TRS    ,L,ONORD,M,SYPRO    ,CWSTK,ISTRE,DEM
     1   ,G,S    ,LAM,B,MU,CR,KOUNT,CO(10),BORDS    ,ND(5),TR         ,NOTON,
     2           T,LSS,ALPHA,DA              ,NOPRC     ,RR,LAMTO,TTG
      COMMON JC,DOT,SGC,GI,GG,I,W,N,VA,VB,VC,VD,VE,KAUNT,PTOT,SGB,SGA
      SUBROUTINE SYRO
      DO 10 K=1,5
   10 S(K)=STOCK(K)+TRS(K)
      RR=CWSTK
  195 CALL SFRS
      IF (F-M) 80,80,111
   80 CALL SETT  (SYPRO (NAME),(L+CLOCK))
      CALL ADDL  (SYPRO (NAME),TIMES)
      NOPRC=NOPRC+1
      SYPRO (STATE)=ONORD
   88 FORMAT(I2,F5.1,I4)
  111 RETURN
      END
```

```
// DUP
*STORE      WS  UA  SYRO

// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE SIMO
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      I=0
    1 FORMAT(I4)
      WRITE(1,1) I
      CALL PLAN
      WRITE(1,1) I
      STATE=2
      NAME=1
      WRITE(1,1) I
      CALL QUEUE(TIMES)
      WRITE(1,1) I
      RETURN
      END
```

```
// DUP
*STORE        WS  UA  SIMO
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE DELE(M,IS)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
   40 J=MAST1(IS)
      DO 1080 I=1,J
   42 K=MAST2(IS)
   44 K=MAST2(K)
   46 IF(MAST1(K)-M)1080,1081,1080
 1080 MAST2(IS)=K
  100 FORMAT(29H MEMBER NOT PRESENT IN DELETE)
      WRITE(3,100)
      CALL EXIT
 1081 K=MAST2(IS)
      KK=MAST2(K)
      MAST2(K)=MAST2(KK)
      MAST2(KK)=TAILM
      TAILM=KK
      NIM=NIM+1
      MAST1(IS)=J-1
      IF(I-1)1084,1082,1084
 1084 I=J-I
      IF(I)1085,1082,1085
 1085 DO 1083 K=1,I
      KK=MAST2(IS)
 1083 MAST2(IS)=MAST2(KK)
 1082 RETURN
      END
```

```
// DUP
*STORE      WS  UA  DELE
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE ADDL(M,L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(NIM)1070,1070,1071
  100 FORMAT(33H MASTER LIST EXHAUSTED IN ADDLAST)
 1070 WRITE(3,100)
      CALL EXIT
 1071 J=TAILM
      TAILM=MAST2(J)
      NIM=NIM-1
      MAST1(J)=M
      IF(MAST1(L))1072,1073,1074
  101 FORMAT(18H ADDLAST TO ENTITY)
 1072 WRITE(3,101)
      CALL EXIT
 1073 MAST2(J)=J
      MAST2(L)=J
      MAST1(L)=1
      GO TO 1075
 1074 MAST1(L)=MAST1(L)+1
      K=MAST2(L)
      MAST2(J)=MAST2(K)
      MAST2(K)=J
      MAST2(L)=J
 1075 RETURN
      END
```

```
// DUP
*STORE      WS  UA  ADDL
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE BEHE(L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(MAST1(L))1030,1030,1031
  100 FORMAT(18H BEHEAD EMPTY LIST)
 1030 WRITE(3,100)
      CALL EXIT
 1031 K=MAST2(L)
      J=MAST2(K)
      MAST2(K)=MAST2(J)
      MAST2(J)=TAILM
      TAILM=J
      NIM=NIM+1
      MAST1(L)=MAST1(L)-1
      RETURN
      END
```

```
// DUP
*STORE       WS  UA  BEHE
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE GROU(IE,IN,IL)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      DIMENSION IE(1,1)
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(NIM-2*IN)1150,1151,1151
 1150 WRITE(3,100)
  100 FORMAT(37H MASTER LIST EXHAUSTED IN GROUPENTITY)
      CALL EXIT
 1151 DO 1152 J=1,IN
      IE(J,1)=TAILM
      TAILM=MAST2(TAILM)
      K=MAST2(TAILM)
      MAST1(TAILM)=IL
      MAST2(TAILM)=J
      TAILM=K
      K=IE(J,1)
 1152 MAST1(K)=-1
      NIM=NIM-2*IN
      RETURN
      END
```

```
// DUP
*STORE       WS  UA  GROU
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE SETT(N,M)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      MAST1(N)=-M-1
  213 FORMAT(3I10)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SETT
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE SCA(IS,IM,IL)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
  456 NIS=MAST1(IS)
      IF(NIS)1090,1090,1091
  100 FORMAT(15H SCAN EMPTY SET)
 1090 WRITE(1,100)
      CALL EXIT
 1091 IP=MAST2(IS)
      IL=-29999
      DO 1092 J=1,NIS
      M=MAST1(IP)
      L=MAST1(M)
      IF(L-IL)1092,1092,1093
 1093 IL=L
      IM=M
 1092 IP=MAST2(IP)
      IL=-1-IL
      RETURN
      END
```

```
// DUP
*STORE       WS  UA  SCA
// FOR
*LIST ALL
*ONE WORD INTEGERS
      INTEGER FUNCTION TIMV(N)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      TIMV=-MAST1(N)-1
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  T+MV
// FOR
*LIST ALL
*ONE WORD INTEGERS
      INTEGER FUNCTION REFN(IE)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      K=MAST2(IE)
      REFN=MAST1(K)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  REFN
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE ENTI(N,M)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      DIMENSION N(1)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(NIM-2)1051,1051,1050
  100 FORMAT(32H MASTER LIST EXHAUSTED IN ENTITY)
 1051 WRITE(3,100)
      CALL EXIT
 1050 N(1)=TAILM
      TAILM=MAST2(TAILM)
      J=MAST2(TAILM)
      NIM=NIM-2
      MAST1(TAILM)=M
      MAST2(TAILM)=TAILM
      TAILM=J
      M=N(1)
      MAST1(M)=-1
      WRITE(1,453)NIM
  453 FORMAT(' NIM=',I4)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  ENTI
// FOR
*LIST ALL
*ONE WORD INTEGERS
      INTEGER FUNCTION SIZE(L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      SIZE=MAST1(L)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SIZE
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE QUEUE(L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2     ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF (NIM)1010,1010,1011
  100 FORMAT(29H MASTER LIST EXHAUSTED IN SET)
 1010 WRITE(3,100)
      CALL EXIT
 1011 L=TAILM
      J=L
      TAILM=MAST2(J)
      NIM=NIM-1
      MAST1(J)=0
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  QUEUE
// FOR
*LIST ALL
*ONE WORD INTEGERS
      INTEGER FUNCTION HEAD (L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(MAST1(L))3,1,2
    1 WRITE(3,100)
  100 FORMAT(18H HEADOF EMPTY LIST)
      CALL EXIT
    3 WRITE(3,200)
  200 FORMAT(14H HEADOF ENTITY)
      CALL EXIT
    2 K=MAST2(L)
      K=MAST2(K)
      HEAD=MAST1(K)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  HEAD
// FOR
*LIST ALL
*ONE WORD INTEGERS
      INTEGER FUNCTION MEMN(IE)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      K=MAST2(IE)
      MEMN=MAST2(K)
      RETURN
      END
```

```
// DUP
*STORE        WS  UA  MEMN

// JOB
// XEQ DSJF       1
*LOCALDSJF,DSJG,SUBA,SUBC
65+091115
// JOB
// XEQ MMMM       1
*LOCALMMMM,CALG,GEN,SYRO,SSRO,P001,      BEHE,DELE,ENTI,GROU,HEAD,MEMN,
*LOCALREFN,SCA,SIMO,SIZE,PNOO
65 +0.901729
```

Program 3

```
// JOB
// DUP
*DELETE                 NNNY
// FOR
*IOCS(CARD,1132 PRINTER,DISK)
*ONE WORD INTEGERS
      REAL L,MU,LAM
      INTEGER S,T
      COMMON SIG(7),C(7,140),CSCAL,CNEG(140),D(7,140),CO(7),S,MU,L,CR,T
      COMMON LAM
      L=25
      DO 5 I=1,12
      LAM=.01*I
      CR=.3
      CS=.4
      ITBL=CR/(CS*LAM)
      ISDA=0
      CALL CAL(ISDA)
      CSO=CSCAL
      ISDA=1
      CALL CAL(ISDA)
      CS1=CSCAL
      DO 3 K=1,ITBL
    3 CNEG(K)=K*LAM*CS+CSO
      CO(1)=CS1
      IX=ITBL+1
      DO 4 K=IX,140
    4 CNEG(K)=CR+CSO
      HA=CO(1)+CR
      HB=(1-LAM)*CO(1)+LAM*CSO
      IF (HB-HA) 77,77,52
   77 D(1,1)=0
      C(1,1)=HB
      GO TO 42
   52 D(1,1)=1
      C(1,1)=HA
   42 DO 44 T=2,140
      S=1
      HA=CO(1)+CR
      HB=(1-LAM)*C(S,T-1)+LAM*CNEG(T-1)
      IF (HB-HA) 78,78,56
   78 D(S,T)=0
      C(S,T)=HB
      GO TO 44
   56 D(S,T)=1
      C(S,T)=HA
   44 CONTINUE
   47 WRITE(3,1) (S,T,D(S,T),T=1,140)
      CALL SUBY
    1 FORMAT(7(I3,I4,F2.0,4X))
    5 CONTINUE
      CALL EXIT
      END
```

```
// DUP
*STORE       WS  UA  NNNY
// JOB
// XEQ NNNY

// JOB
// DUP
*DELETE             SUBY
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE SUBY
      REAL L,MU,LAM
      INTEGER S,T
      COMMON SIG(7),C(7,140),CSCAL,CNEG(140),D(7,140),CO(7),S,MU,L,CR,T
      COMMON LAM
      DO 3 S=2,7
      ISDA=S
      CALL CAL(ISDA)
      CO(S)=CSCAL
      HA=CO(S)+CR
      HB=(1-LAM)*CO(S)+LAM*CO(S-1)
      IF (HB-HA) 53,53,51
   53 D(S,1)=0
      C(S,1)=HB
      GO TO 41
   51 D(S,1)=1
      C(S,1)=HA
   41 DO 4 T=2,140
      HA=CO(S)+CR
      HB=(1-LAM)*C(S,T-1)+             LAM*C(S-1,T-1)
      IF (HB-HA) 8,8,5
    8 D(S,T)=0
      C(S,T)=HB
      GO TO 4
    5 D(S,T)=1
      C(S,T)=HA
    4 CONTINUE
      WRITE(3,1)(S,T,D(S,T),T=1,140)
    3 CONTINUE
    1 FORMAT(7(I3,I4,F2.0,4X))
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SUBY
// FOR
*ONE WORD INTEGERS
      SUBROUTINE CAL(ISDA)
      REAL L,MU,LAM
      INTEGER S,T
      COMMON SIG(7),C(7,140),CSCAL,CNEG(140),D(7,140),CO(7),S,MU,L,CR,T
      COMMON LAM
      A=0
      CD=0
      RM=LAM*L
      TM=EXP(-RM)
    3 A=A+1
      TM=TM*RM/A
      IF (ISDA) 2,5,5
    2 ISDA=0
    5 IF (A-ISDA) 3,3,4
    4 CI=TM*(A-ISDA)*.4
      CD=CD+CI
      IF (CI-.0001) 7,7,3
    7 CSCAL=CD
      RETURN
      END
```

```
// DUP
*STORE       WS  UA  CAL
// FOR
*ONE WORD INTEGERS
*IOCS(TYPEWRITER)
*IOCS(CARD,1132 PRINTER,DISK)
      REAL ITFN(5)
      REAL LAM(5),LAMT,RA(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      INTEGER STOCK(5)
      INTEGER                 REPTO,TR(5,10)   ,DEL(25,2),FIN(2),EVENT(2)
     1,TFIN,TIMV,REFN,MEMN,Q,JUMP(2)
      DIMENSION          NUM(5),KCHEC(5),ND(5)
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
  171 READ (2,88) Y,Z,ALPHA,ISTRE,INDY
      CALL SIMO
   88 FORMAT(2F4.1,F5.3,2I4)
      CALL GROU (DEM,5,1)
      CALL GROU(DEL,25,4)
      AREA=0
      CBO=0
      KOUN=0
      BET=1-ALPHA
   39 DO 46 K=1,5
   46 RA(K)=-1/(.01*K)
      DO 74 K=1,5
   19 IT=ALOG(RANDY(INDY))*RA(K)+.5
      ITFN(K)=IT
      ITTE=IT
  190 IT=ALOG(RANDY(INDY))*RA(K)+.5
      ITTE=ITTE+IT
      IF (ITTE-1000) 139,139,74
  139 ITFN(K)=ALPHA*ITFN(K)+BET*IT
      GO TO 190
   74 CONTINUE
  138 LAMT=0
      DO 47 K=1,5
      LAM(K)=1/ITFN(K)
      WRITE(1,40) LAM(K)
   40 FORMAT(F5.3)
   47 LAMT=LAM(K)+LAMT
      WRITE(1,40) LAMT
      Q=60
      LC=125
      LSS=25
      L=100
      CR=.3
      G=0
      NOPRC=0
      KA=-50
      REPTO=0
      NOTON=0
      ONORD=1
      VC=12
      NDEL=0
      CALL ENTI(JUMP,5)
```

```
      CALL ENTI(SYPRO,2)
      CALL ENTI(EVENT,7)
      CALL ENTI(FIN,3)
      TFIN=1000
      IJUMP=400
      CLOCK=0
      SYPRO(STATE)=NOTON
      CALL SETT(FIN(NAME),TFIN)
      CALL SETT(JUMP(NAME),IJUMP)
      CALL ADDL(JUMP(NAME),TIMES)
      WRITE(1,88) Y,Z,ALPHA,ISTRE
      CALL ADDL  (FIN(NAME),TIMES)
      CALL SETT (EVENT(NAME),0)
      FMEA=LAMT*LC
      M=FMEA+Y*SQRT(FMEA)+.5
      F=M-LAMT*L+Q+.5
      CWSTK=F
      DO 217 J=1,5
      CALL PNOX
      IF (B-F*LAM(J)/LAMT) 200,200,201
  201 B=F*LAM(J)/LAMT
  200 CWSTK=CWSTK-B
      IF (CWSTK) 15,15,14
   14 STOCK(J)=B
      GO TO 217
   15 STOCK(J)=CWSTK
      CWSTK=0
      GO TO 888
  217 CONTINUE
  888 DO 79 J=1,5
      CALL GENX
      ND(J)=0
      TRS(J)=0
      NOSTK(J)=TRS(J)+STOCK(J)
      KCHEC(J)=0
      NUM(J)=0
      DO 79 K=1,10
   79 TR(J,K)=0
      WRITE(1,175) CLOCK,STOCK(1),TRS(1),STOCK(2),TRS(2),STOCK(3),TRS(3)
     1,STOCK(4),TRS(4),STOCK(5),TRS(5),CWSTK
      GO TO 4001
11111 CALL SCA (TIMES,MEMBE ,CLOCK)
      K=REFN  (MEMBE )
      CALL DELE  (MEMBE ,TIMES)
      CALL CALGX
  500 AREA=AREA+G*(CLOCK-TIMV  (EVENT(NAME)))
  800 CALL SETT  (EVENT(NAME),CLOCK)
  116 GO TO (1,2,3,4,5),K
C                         A.PHASE
C               DEMAND ON SUB-STORE
    1 J=MEMN  (MEMBE )
   69 STOCK(J)=STOCK(J)-1
      NOSTK(J)=NOSTK(J)-1
      ITFN(J)=ALPHA*ITFN(J)+BET*(CLOCK-KCHEC(J))
      OLJ=LAM(J)
      LAM(J)=1/ITFN(J)
      KCHEC(J)=CLOCK
      LAMT=LAMT+LAM(J)-OLJ
      WRITE(1,90) LAMT,LAM(J),OLJ,J
```

```
   90 FORMAT(3F7.3,I6)
  910 IF (STOCK(J)) 10,1000,1000
   10 CBO=CBO+.4
 1000 CALL GENX
      WRITE(1,175) CLOCK,STOCK(1),TRS(1),STOCK(2),TRS(2),STOCK(3),TRS(3)
     1,STOCK(4),TRS(4),STOCK(5),TRS(5),CWSTK
      GO TO 4000
C             DELIVERY AT CENTRAL STORE
    2 CWSTK=CWSTK+Q
      SYPRO(STATE)=NOTON
      VC=12
      GO TO 4001
C            FINISH SIMULATION
    3 CALL CALGX
      AREA=AREA+G*(CLOCK-TIMV  (EVENT(NAME)))
      HC=.0004*AREA
      TCOST=.5*NOPRC+ CR*REPTO+CBO+HC
      WRITE(1,262)NOPRC,REPTO,CBO,HC,  TCOST
  262 FORMAT(' NOPRC=',I4,' REPTO=',I4,' CBO=',F8.4,' AREA=',F10.2,' TCO
     1ST=',F12.4)
      GO TO 171
C        DELIVERY AT SUBSTORE
    4 NN=MEMN(MEMBE)
      J=DEL(NN,STATE)
      NUM(J)=NUM(J)+1
      K=NUM(J)
      IF (K-10) 91,104,91
  104 NUM(J)=0
   91 STOCK(J)=STOCK(J)+TR(J,K)
      TRS(J)=TRS(J)-TR(J,K)
      WRITE(1,175) CLOCK,STOCK(1),TRS(1),STOCK(2),TRS(2),STOCK(3),TRS(3)
     1,STOCK(4),TRS(4),STOCK(5),TRS(5),CWSTK
      GO TO 4000
C        JUMP IS APPLIED
    5 RA(3)=-1/.06
      GO TO 11111
C                              C-PHASE
C     TEST FOR SUBSTORE ORDER OR SYSTEM ORDER
 4000 IF (SYPRO(STATE)-NOTON) 304,4001,304
 4001 CALL SYRX
  304 IF (CWSTK) 11111,11111,305
  305 DO 777 J=1,5
      IF (CWSTK) 11111,11111,306
  306 IF (NOSTK(J)-11) 111,111,777
  111 CALL SSRX
  497 IF (KAUNT-94)   777,338,339
  338 CALL PNOX
      IF (VC-12) 38,48,38
   48 IF (B-F*LAM(J)/LAMT) 38,38,49
   49 B=F*LAM(J)/LAMT
      GO TO 38
  339 CALL POOX
   38 B=B-NOSTK(J)
      IF (B-CWSTK) 8,8,7
    7 B=CWSTK
      GO TO 118
    8 IF (B) 777,777,118
  118 ND(J)=ND(J)+1
      KT=ND(J)
```

```
      IF (KT-10)  222,223,222
  223 ND(J)=0
  222 TR(J,KT)=B
      TRS(J)=TRS(J)+B
      NOSTK(J)=NOSTK(J)+B
      REPTO=REPTO+1
      NDEL=NDEL+1
      CWSTK=CWSTK-B
      IF (NDEL-26) 55,56,55
   56 NDEL=1
   55 CALL SETT  (DEL(NDEL,NAME),(CLOCK+LSS))
      CALL ADDL  (DEL(NDEL,NAME),TIMES)
      DEL(NDEL,STATE)=J
      WRITE(1,175) CLOCK,STOCK(1),TRS(1),STOCK(2),TRS(2),STOCK(3),TRS(3)
     1,STOCK(4),TRS(4),STOCK(5),TRS(5),CWSTK
  175 FORMAT (12I5)
  777 CONTINUE
      VC=0
      GO TO 11111
      END
```

```
// DUP
*STORE      WS  UA  MMMX
// JOB
// DUP
*DELETE             CAIEX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE CAIEX
      REAL RA(5),LAM(5),LAMT
      INTEGER IE(12,11)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      IF (KA)  30,4444,4444
   30 DO 75  K=1,12
      DO 75 I=1,11
   75 IE(K,I)=11111
      IE(1,1)=160
      IE(2,1)=68
      IE(2,2)=145
      IE(3,1)=38
      IE(3,2)=74
      IE(3,3)=138
      IE(4,1)=26
      IE(4,2)=45
      IE(4,3)=82
      IE(4,4)=137
      IE(5,1)=19
      IE(5,2)=31
      IE(5,3)=53
      IE(5,4)=89
      IE(5,5)=138
      IE(6,1)=15
      IE(6,2)=22
      IE(6,3)=37
      IE(6,4)=62
      IE(6,5)=97
      IE(6,6)=140
      IE(7,1)=13
      IE(7,2)=17
      IE(7,3)=27
      IE(7,4)=45
      IE(7,5)=70
      IE(7,6)=103
      IE(8,1)=11
      IE(8,2)=14
      IE(8,3)=21
      IE(8,4)=34
      IE(8,5)=53
      IE(8,6)=78
      IE(8,7)=109
      IE(9,1)=10
      IE(9,2)=12
      IE(9,3)=17
      IE(9,4)=26
      IE(9,5)=40
```

```
      IE(9,6)=60
      IE(9,7)=85
      IE(9,8)=141
      IE(10,1)=9
      IE(10,2)=11
      IE(10,3)=14
      IE(10,4)=20
      IE(10,5)=31
      IE(10,6)=46
      IE(10,7)=67
      IE(10,8)=92
      KA=100
 4444 FLAM=100*LAM(J)
      ILAM=FLAM
      ILAM1=ILAM+1
      A=FLAM-ILAM
      IEX=A*IE(ILAM,N)+(1-A)*IE(ILAM1,N)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  CAIEX
// JOB
// DUP
*DELETE             CALCX
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE CALCX(ISDA,ILDA,RLAMD)
      REAL LAM(5),LAMT,RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      A=0
      CO=0
      RMUDA=RLAMD*ILDA
      TM=EXP(-RMUDA)
    3 A=A+1
      TM=TM*RMUDA/A
      IF (ISDA) 2,5,5
    2 ISDA=0
    5 IF (A-ISDA) 3,3,4
    4 CI=TM*(A-ISDA)*.4
      CO=CO+CI
      IF (CI-.01) 7,7,3
    7 VA=CO
      RETURN
      END
```

```
// DUP
*STORE        WS  UA  CALCX
// JOB
// DUP
*DELETE             CALGX
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE CALGX
      REAL RA(5),LAM(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      INTEGER SUMS
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      SUMS=0.
      DO 2 K=1,5
      KS=NOSTK(K)-TRS(K)
      IF (KS) 2,2,8
    8 SUMS=SUMS+KS
    2 CONTINUE
      G=SUMS+TRS(1)+TRS(2)+TRS(3)+TRS(4)+TRS(5) +CWSTK
      RETURN
      END
```

```
// DUP
*STORE       WS  UA  CALGX
// JOB
// DUP
*DELETE              CALBX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE CALBX(D)
      REAL RA(5),LAM(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      DIMENSION IEN(12)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      VB=0
      JC=0
      IBL=B-NOSTK(J)
      IW=NOSTK(J)
      IT=T
      DO 33 N=1,12
      CALL CAIEX
      IEN(N)=IEX
      IF (IEX-100) 33,33,34
   34 K=N-1
      GO TO 35
   33 CONTINUE
   35 IFL=0
   30 V=D/FLOAT(T-1)
    3 IF (IW) 130,130,13
  130 IF (V*(IT-.75/LAM(J))-IBL) 5,4,4
   13 IF (V*(IT-IEN(IW)) -IBL) 5,4,4
    5 IF (IW-IFL) 6,1,6
    6 IBL=IBL+1
      IW=IW-1
      GO TO 3
    4 IT=IT-FLOAT(IBL)/V+.5
      JC=JC+1
      IBL=(IFL+1)+(IT-1)*(FLOAT(B-IFL-1))/FLOAT(T-1)+.5-IW
      IF (IBL) 27,27,28
   27 IBL=1
   28 VB=VB+IBL
C         VB IS SUM OF LITTLE B
      IF (IW-IFL) 3,1,3
    1 RETURN
      END
```

```
// DUP
*STORE       WS  UA  CALBX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE GENX
      REAL LAM(5),RA(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
   21 FORMAT(F12.6,I4)
      CAM=ALOG(RANDY(ISTRE))
      INT=CAM*RA(J)+.5
 6666 INTA=CLOCK+INT
      CALL SETT  (DEM(J,NAME),INTA)
      CALL ADDL  (DEM(J,NAME),TIMES)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  GENX
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE PNOX
      REAL RA(5)
      REAL LAM(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      RL=LAM(J)/LAMT
      SRB=F-M
      HI=SRB*RL+LAM(J)*LC
      B=HI+Z*SQRT(SRB*RL*RL+HI)+.5
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  PNOX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE POOX
      REAL LAM(5),LAMT,RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      DIMENSION S(5)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      U=LAM(J)*(T-1)
      B=NOSTK(J)+1
   76 D=0
      PTM=EXP(-U)
    1 JC=0
      VB=0
      IF (D) 31,31,773
  773 CALL CALBX(D)
   31 RR=CWSTK-(B+VB-NOSTK(J))
      DO 23 K=1,5
   23 S(K)=NOSTK(J)/LAM(K)
      S(J)=9999
  100 IF (RR) 91,91,21
   21 I=1
      K=2
   11 IF (S(I)-S(K)) 32,32,8
   32 K=K+1
      IF (K-5) 11,17,17
    8 I=K
      GO TO 3
   17 RR=RR-1
      S(I)=S(I)+1/LAM(J)
      GO TO 100
   91 VT=0
      DO 20 K=1,5
      IF (K-J) 7,20,7
    7 ILDA=T+LSS
      ISDA=S(K)*LAM(K)
      RLAMD=LAM(K)
      CALL CALCX(ISDA,ILDA,RLAMD)
      VT=VT+VA
   20 CONTINUE
      VA=VT
  183 TC=VA+JC*CR
      IF (D) 18,18,19
   18 SUM=TC*PTM
      SIGP=PTM
      GO TO 2
   19 PTM=U*PTM/D
      SIGP=SIGP+PTM
      SUM=SUM+TC*PTM
      IF (SIGP-.95) 2,3,3
    2 D=D+1
      GO TO 773
    3 IF (B-NOSTK(J)-1)  37,22,37
   22 SUMOL=SUM
```

```
 80 IF (B-NOSTK(J)-CWSTK)  75,75,81
 75 B=B+1
    GO TO 76
 37 IF (SUM-SUMOL) 22,81,81
 81 B=B-1
700 RETURN
    END
```

```
// DUP
*STORE        WS  UA  POOX

// JOB
// DUP
*DELETE             SFRX
// FOR
* ONE WORD INTEGERS
      SUBROUTINE SFRX
      REAL LAM(5),LAMT
      REAL RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      INTEGER RR
      DIMENSION S(5)
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      DO 1 K=1,5
    1 S(K)=NOSTK(K)/LAM(K)
      RR=CWSTK
    2 I=1
      K=2
   11 IF (S(I)-S(K)) 3,3,8
    3 K=K+1
      IF (K-6) 11,7,7
    8 I=K
      GO TO 3
    7 IF (RR) 34,34,21
   21 S(I)=S(I)+1/LAM(I)
      RR=RR-1
      GO TO 2
   34 F=S(I)*LAMT
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SFRX
// JOB
// DUP
*DELETE             SSRX
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE SSRX
      REAL RA(5),LAM(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      INTEGER TIMV
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      C=LAM(J)
      IF (SYPRO(STATE)-ONORD) 76,41,41
   76 IF (C-.095) 7,6,6
    6 IR=8
      GO TO 21
    7 IF (C-.083) 5,4,4
    4 IR=7
      GO TO 21
    5 IF (C-.071) 2,1,1
    1 IR=6
      GO TO 21
    2 IF (C-.059)  11,12,12
   12 IR=5
      GO TO 21
   11 IF (C-.046) 13,14,14
   14 IR=4
      GO TO 21
   13 IF (C-.035) 15,16,16
   16 IR=3
      GO TO 21
   15 IF (C-.025) 17,18,18
   18 IR=2
      GO TO 21
   17 IF (LAM(J)-.015) 19,20,20
   20 IR=1
      GO TO 21
   19 IR=0
   21 IF (NOSTK(J)-IR) 26,26,27
   27 KAUNT=0
      RETURN
   26 KAUNT=94
      RETURN
   41 N=NOSTK(J)
   52 T=TIMV(SYPRO(NAME))-CLOCK
      IF (N) 42,42,43
   42 IF (T-.75/C) 26,95,95
   43 CALL CATEX
      IF (T-IEX) 27,95,95
   95 KAUNT=95
   34 FORMAT(2I2)
      RETURN
      END
```

```
// DUP
*STORE        WS   UA   SSRX
// FOR
*LIST ALL
* ONE WORD INTEGERS
      SUBROUTINE SYRX
      REAL LAM(5),LAMT
      REAL RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      CALL SFRX
      FMEA=LAMT*LC
      M=FMEA+Y*SQRT(FMEA)+.5
      IF (F-M) 80,80,111
   80 CALL SETT  (SYPRO (NAME),(LC-LSS+CLOCK))
      CALL ADDL  (SYPRO (NAME),TIMES)
      NOPRC=NOPRC+1
      SYPRO (STATE)=ONORD
      WRITE(1,1) CLOCK,NOPRC
    1 FORMAT(2I6)
  111 RETURN
      END
```

Program 4

```
// JOB
// FOR
*ONE WORD INTEGERS
*IOCS(TYPEWRITER)
*IOCS(CARD,1132 PRINTER,DISK)
      REAL LAM(5),LAMT,RA(5)
      INTEGER IROL(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      INTEGER STOCK(5)
      INTEGER                 REPTO,TR(5,10)    ,DEL(25,2),FIN(2),EVENT(2)
     1,TFIN,TIMV,REFN,MEMN,Q
      DIMENSION          NUM(5),KCHEC(5),ND(5)
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
  171 READ(2,88) M,ISTRE,Q,TFIN,PR,HBC,CP
   88 FORMAT(4I4,3F4.2)
      CALL SIMO
      CALL GROU (DEM,5,1)
      CALL GROU(DEL,25,4)
      AREA=0
      CBO=0
      KOUN=0
   39 DO 46 K=1,5
   46 RA(K)=-1/(.01*K)
      LC=125
      LSS=25
      L=100
      CR=.3
      G=0
      NOPRC=0
      KA=-50
      REPTO=0
      IROL(1)=3
      IROL(2)=4
      IROL(3)=4
      IROL(4)=5
      IROL(5)=5
      NOTON=0
      ONORD=1
      VC=12
      NDEL=0
      CALL ENTI(SYPRO,2)
      CALL ENTI(EVENT,7)
      CALL ENTI(FIN,3)
      CLOCK=0
      SYPRO(STATE)=NOTON
      CALL SETT(FIN(NAME),TFIN)
      CALL ADDL   (FIN(NAME),TIMES)
      CALL SETT (EVENT(NAME),0)
      DO 3523 K=1,5
 3523 LAM(K)=.01*K
      LAMT=.15
      F=M-LAMT*L+Q+.5
      CWSTK=F
      DO 217 J=1,5
      KCHEC(J)=0
      CRA=LAM(J)/LAMT
```

```
      B=F*CRA-HBC*SQRT(F*(CRA+CRA*CRA))+.5
   14 STOCK(J)=B
      IF (B-CWSTK)      465,465,466
  466 STOCK(J)=CWSTK
      CWSTK=0
      GO TO 888
  465 CWSTK=CWSTK-B
  217 CONTINUE
  888 DO 79 J=1,5
      CALL GENX
      ND(J)=0
      TRS(J)=0
      NOSTK(J)=TRS(J)+STOCK(J)
      NUM(J)=0
      DO 79 K=1,10
   79 TR(J,K)=0
      GO TO 4001
11111 CALL SCA (TIMES,MEMBE ,CLOCK)
      K=REFN  (MEMBE )
      CALL DELE  (MEMBE ,TIMES)
      CALL CALGX
  500 AREA=AREA+G*(CLOCK-TIMV  (EVENT(NAME)))
  800 CALL SETT  (EVENT(NAME),CLOCK)
  116 GO TO (1,2,3,4),K
C                            A.PHASE
C                DEMAND ON SUB-STORE
    1 J=MEMN  (MEMBE )
   69 STOCK(J)=STOCK(J)-1
      NOSTK(J)=NOSTK(J)-1
   90 FORMAT(3F7.3,I6)
      F=NOSTK(1)+NOSTK(2)+NOSTK(3)+NOSTK(4)+NOSTK(5)+CWSTK
  910 IF (STOCK(J)) 10,1000,1000
   10 CBO=CBO+.4
  374 FORMAT(2F10.6)
 1000 CALL GENX
      GO TO 4000
C                DELIVERY AT CENTRAL STORE
    2 CWSTK=CWSTK+Q
      SYPRO(STATE)=NOTON
      VC=12
      GO TO 4001
C               FINISH SIMULATION
    3 CALL CALGX
      AREA=AREA+G*(CLOCK-TIMV  (EVENT(NAME)))
      HC=.0004*PR*AREA
      TCOST=CP*NOPRC+ CR*REPTO+CBO+HC
      WRITE(1,262)NOPRC,REPTO,CBO,HC,  TCOST
  262 FORMAT(' NOPRC=',I4,' REPTO=',I4,' CBO=',F8.4,' HC  =',F10.2,' TCO
     1ST=',F12.4)
      GO TO 171
C          DELIVERY AT SUBSTORE
    4 NN=MEMN(MEMBE)
      J=DEL(NN,STATE)
      NUM(J)=NUM(J)+1
      K=NUM(J)
      IF (K-10) 91,104,91
  104 NUM(J)=0
   91 STOCK(J)=STOCK(J)+TR(J,K)
      TRS(J)=TRS(J)-TR(J,K)
```

```
      GO TO 4000
C                                 C-PHASE
C     TEST FOR SUBSTORE ORDER OR SYSTEM ORDER
 4000 IF (SYPRO(STATE)-NOTON) 304,4001,304
 4001 CALL SYRY
  304 IF (CWSTK) 11111,11111,305
  305 DO 777 J=1,5
      IF (CWSTK) 11111,11111,306
  306 IF (VC-12) 307,40,307
  307 IF (NOSTK(J)-IROL(J)) 40,40,777
   40 F=NOSTK(1)+NOSTK(2)+NOSTK(3)+NOSTK(4)+NOSTK(5)+CWSTK
      CRA=LAM(J)/LAMT
      B=F*CRA-HBC*SQRT(F*(CRA+CRA*CRA))+.5
      IF (B-LAM(J)*LSS) 50,60,60
   50 B=LAM(J)*LSS+.5
   60 B=B-NOSTK(J)
      IF (B-CWSTK) 8,8,7
    7 B=CWSTK
      GO TO 118
    8 IF (B) 777,777,118
  118 ND(J)=ND(J)+1
      IF (J-5) 713,210,713
  210 IF (HBC) 713,613,713
  613 IF (VC-12) 713,513,713
  513 B=CWSTK
  713 KT=ND(J)
      IF (KT-10)  222,223,222
  223 ND(J)=0
  222 TR(J,KT)=B
      TRS(J)=TRS(J)+B
      NOSTK(J)=NOSTK(J)+B
      REPTO=REPTO+1
  653 FORMAT(I6,F6.0)
      NDEL=NDEL+1
      CWSTK=CWSTK-B
      IF (NDEL-26) 55,56,55
   56 NDEL=1
   55 CALL SETT  (DEL(NDEL,NAME),(CLOCK+LSS))
      CALL ADDL  (DEL(NDEL,NAME),TIMES)
      DEL(NDEL,STATE)=J
  175 FORMAT (12I5)
  777 CONTINUE
      VC=0
      GO TO 11111
      END
```

```
// DUP
*STORE      WS  UA  MMMC
// FOR
*IOCS(CARD,1132 PRINTER,DISK,TYPEWRITER)
*ONE WORD INTEGERS
      REAL LAM(5),LAMT,RA(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      INTEGER STOCK(5)
      INTEGER                    REPTO,TR(5,10)   ,DEL(25,2),FIN(2),EVENT(2)
     1,TFIN,TIMV,REFN,MEMN,Q
      DIMENSION          NUM(5),KCHEC(5),ND(5)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
  171 READ(2,88) M,ISTRE,Q,TFIN,PR,Z,CP
   88 FORMAT(4I4,3F4.2)
      CALL SIMO
      CALL GROU (DEM,5,1)
      CALL GROU(DEL,25,4)
      AREA=0
      CBO=0
      KOUN=0
   39 DO 46 K=1,5
   46 RA(K)=-1/(.01*K)
      LC=125
      LSS=25
      L=100
      CR=.3
      G=0
      NOPRC=0
      KA=-50
      REPTO=0
      NOTON=0
      ONORD=1
      VC=12
      NDEL=0
      CALL ENTI(SYPRO,2)
      CALL ENTI(EVENT,7)
      CALL ENTI(FIN,3)
      CLOCK=0
      SYPRO(STATE)=NOTON
      CALL SETT(FIN(NAME),TFIN)
      WRITE(I,88) M,ISTRE,Q,TFIN,PR,Z,CP
      CALL ADDL  (FIN(NAME),TIMES)
      CALL SETT (EVENT(NAME),0)
      DO 3523 K=1,5
 3523 LAM(K)=.01*K
      LAMT=.15
      F=M-LAMT*L+Q+.5
      CWSTK=F
      DO 217 J=1,5
      KCHEC(J)=0
      CALL PNOX
      IB=F*LAM(J)/LAMT+.5
      IF (B-IB) 200,200,201
  201 B=IB
  200 STOCK(J)=B
      IF (B-CWSTK)      465,465,466
```

```
  466 STOCK(J)=CWSTK
      CWSTK=0
      GO TO 888
  465 CWSTK=CWSTK-B
  217 CONTINUE
  888 DO 79 J=1,5
      CALL GENX
      ND(J)=0
      TRS(J)=0
      NOSTK(J)=TRS(J)+STOCK(J)
      NUM(J)=0
      DO 79 K=1,10
   79 TR(J,K)=0
      GO TO 4001
11111 CALL SCA (TIMES,MEMBE ,CLOCK)
      K=REFN  (MEMBE )
      CALL DELE  (MEMBE ,TIMES)
      CALL CALGX
  500 AREA=AREA+G*(CLOCK-TIMV  (EVENT(NAME)))
  800 CALL SETT  ,(EVENT(NAME),CLOCK)
  116 GO TO (1,2,3,4),K
C                           A.PHASE
C               DEMAND ON SUB-STORE
    1 J=MEMN  (MEMBE )
   69 STOCK(J)=STOCK(J)-1
      NOSTK(J)=NOSTK(J)-1
   90 FORMAT(3F7.3,I6)
  910 IF (STOCK(J)) 10,1000,1000
   10 CBO=CBO+.4
 1000 CALL GENX
      GO TO 4000
C               DELIVERY AT CENTRAL STORE
    2 CWSTK=CWSTK+Q
      SYPRO(STATE)=NOTON
      VC=12
      GO TO 4001
C             FINISH SIMULATION
    3 CALL CALGX
      AREA=AREA+G*(CLOCK-TIMV  (EVENT(NAME)))
      HC=.0004*PR*AREA
      TCOST=CP*NOPRC+ CR*REPTO+CBO+HC
      WRITE(1,262)NOPRC,REPTO,CBO,HC,  TCOST
  262 FORMAT(' NOPRC=',I4,' REPTO=',I4,' CBO=',F8.4,' HC  =',F10.2,' TCO
     1ST=',F12.4)
      GO TO 171
C         DELIVERY AT SUBSTORE
    4 NN=MEMN(MEMBE)
      J=DEL(NN,STATE)
      NUM(J)=NUM(J)+1
      K=NUM(J)
      IF (K-10) 91,104,91
  104 NUM(J)=0
   91 STOCK(J)=STOCK(J)+TR(J,K)
      TRS(J)=TRS(J)-TR(J,K)
      GO TO 4000
C                           C-PHASE
C     TEST FOR SUBSTORE ORDER OR SYSTEM ORDER
 4000 IF (SYPRO(STATE)-NOTON) 304,4001,304
 4001 CALL SYRX
```

```
304 IF (CWSTK) 11111,11111,305
305 DO 777 J=1,5
    IF (CWSTK) 11111,11111,306
306 IF (NOSTK(J)- 5) 111,111,777
111 CALL SSRX
497 IF (KAUNT-94)   777,338,339
338 CALL SFRX
    CALL PNOX
    IF (VC-12) 38,48,38
 48 IB=F*LAM(J)/LAMT+.5
    IF (B-IB) 38,38,49
 49 B=IB
    GO TO 38
339 CALL POOX
 38 B=B-NOSTK(J)
    IF (B-CWSTK) 8,8,7
  7 B=CWSTK
    GO TO 118
  8 IF (B) 777,777,118
118 ND(J)=ND(J)+1
    KT=ND(J)
    IF (KT-10)  222,223,222
223 ND(J)=0
222 TR(J,KT)=B
    TRS(J)=TRS(J)+B
    NOSTK(J)=NOSTK(J)+B
    REPTO=REPTO+1
    NDEL=NDEL+1
    CWSTK=CWSTK-B
    IF (NDEL-26) 55,56,55
 56 NDEL=1
 55 CALL SETT  (DEL(NDEL,NAME),(CLOCK+LSS))
    CALL ADDL  (DEL(NDEL,NAME),TIMES)
    DEL(NDEL,STATE)=J
175 FORMAT (12I5)
777 CONTINUE
    VC=0
    GO TO 11111
    END
```

```
// DUP
*STORE      WS   UA   MMMY
// FOR
* ONE WORD INTEGERS
      SUBROUTINE SFRX
      REAL LAM(5),LAMT
      REAL RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      DO 1 K=1,5
    1 S(K)=NOSTK(K)/LAM(K)
      RR=CWSTK
    2 I=1
      K=2
   11 IF (S(I)-S(K)) 3,3,8
    3 K=K+1
      IF (K-6) 11,7,7
    8 I=K
      GO TO 3
    7 IF (RR) 34,34,21
   21 S(I)=S(I)+1/LAM(I)
      RR=RR-1
      GO TO 2
   34 F=S(I)*LAMT
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SFRX
// FOR
* ONE WORD INTEGERS
      SUBROUTINE SYRX
      REAL LAM(5),LAMT
      REAL RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      DIMENSION S(5)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      DO 1 K=1,5
    1 S(K)=NOSTK(K)/LAM(K)
      RR=CWSTK
    2 I=1
      K=2
   11 IF (S(I)-S(K)) 3,3,8
    3 K=K+1
      IF (K-6) 11,7,7
    8 I=K
      GO TO 3
    7 IF (RR) 34,34,21
   21 S(I)=S(I)+1/LAM(I)
      RR=RR-1
      GO TO 2
   34 F=S(I)*LAMT
      IF (F-M) 80,80,111
   80 TOT=0
      DO 434 K=1,5
      ISDA=S(K)*LAM(K)+.5
      INDE=ISDA
      ILDA=LC
      RLAMD=LAM(K)
      CALL CALCX(ISDA,ILDA,RLAMD)
      VI=VA
      ISDA=INDE-1
      CALL CALCX(ISDA,ILDA,RLAMD)
  170 FORMAT(2F23.6)
      IF (ISDA) 707,434,434
  707 VA=VA+.4
  434 TOT=(VA-VI)*LAM(K)/LAMT+TOT
      IF (TOT-4.7/37.5) 111,180,180
  180 CALL SETT  (SYPRO (NAME),(LC-LSS+CLOCK))
      CALL ADDL  (SYPRO (NAME),TIMES)
      NOPRC=NOPRC+1
      SYPRO (STATE)=ONORD
      WRITE(1,9) CLOCK,NOPRC
    9 FORMAT(2I6)
  111 RETURN
      END
```

```
// DUP
*STORE       WS   UA   SYRX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE PNOX
      REAL RA(5)
      REAL LAM(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      RL=LAM(J)/LAMT
      SRB=F-M
      IF (SRB) 7,8,8
    7 SRB=0
    8 HI=SRB*RL+LAM(J)*LC
      B=HI+Z*SQRT(SRB*RL*RL+HI)+.5
      RETURN
      END
```

```
// DUP
*STORE        WS   UA   PNOX

// JOB

(=7 9K5A'7 1'Y2'X/*9X*(P*7P*7)P(X*9(P(PM=*(**7R*9)727'PX'7PX=7( 7P1X=/'/
// FOR
*ONE WORD INTEGERS
      SUBROUTINE GENX
      REAL LAM(5),RA(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
   21 FORMAT(F12.6,I4)
      CAM=ALOG(RANDY(ISTRE))
      INT=CAM*RA(J)+.5
 6666 INTA=CLOCK+INT
      CALL SETT  (DEM(J,NAME),INTA)
      CALL ADDL  (DEM(J,NAME),TIMES)
      RETURN
      END
```

```
// DUP
*STORE       WS   UA   GENX
// JOB
// DUP
*DELETE               MMMC
// JOB
// DUP
*DELETE               MMMY
// FOR
*ONE WORD INTEGERS
      SUBROUTINE SSRX
      REAL RA(5),LAM(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      INTEGER TIMV
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      C=LAM(J)
      IF (SYPRO(STATE)-ONORD) 76,41,41
   76 IF (C-.095) 7,6,6
    6 IR=8
      GO TO 21
    7 IF (C-.083) 5,4,4
    4 IR=7
      GO TO 21
    5 IF (C-.071) 2,1,1
    1 IR=6
      GO TO 21
    2 IF (C-.059)  11,12,12
   12 IR=5
      GO TO 21
   11 IF (C-.046) 13,14,14
   14 IR=4
      GO TO 21
   13 IF (C-.035) 15,16,16
   16 IR=3
      GO TO 21
   15 IF (C-.025) 17,18,18
   18 IR=2
      GO TO 21
   17 IF (LAM(J)-.015) 19,20,20
   20 IR=1
      GO TO 21
   19 IR=0
   21 IF (NOSTK(J)-IR) 26,26,27
   26 KAUNT=94
      RETURN
   27 KAUNT=0
      RETURN
   41 N=NOSTK(J)
   52 T=TIMV(SYPRO(NAME))-CLOCK
      IF (N) 42,42,43
   42 IF (T-.75/C) 26,95,95
   43 CALL CAIEX
      IF (T-IEX) 27,95,95
   95 KAUNT=95
   34 FORMAT(2I2)
```

```
RETURN
END
```

```
// DUP
*STORE      WS  UA  SSRX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE CAIEX
      REAL RA(5),LAM(5),LAMT
      INTEGER IE(12,11)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      IF (KA)  30,4444,4444
   30 DO 75  K=1,12
      DO 75 I=1,11
   75 IE(K,I)=11111
      IE(1,1)=160
      IE(2,1)=68
      IE(2,2)=145
      IE(3,1)=38
      IE(3,2)=74
      IE(3,3)=138
      IE(4,1)=26
      IE(4,2)=45
      IE(4,3)=82
      IE(4,4)=137
      IE(5,1)=19
      IE(5,2)=31
      IE(5,3)=53
      IE(5,4)=89
      IE(5,5)=138
      IE(6,1)=15
      IE(6,2)=22
      IE(6,3)=37
      IE(6,4)=62
      IE(6,5)=97
      IE(6,6)=140
      IE(7,1)=13
      IE(7,2)=17
      IE(7,3)=27
      IE(7,4)=45
      IE(7,5)=70
      IE(7,6)=103
      IE(8,1)=11
      IE(8,2)=14
      IE(8,3)=21
      IE(8,4)=34
      IE(8,5)=53
      IE(8,6)=78
      IE(8,7)=109
      IE(9,1)=10
      IE(9,2)=12
      IE(9,3)=17
      IE(9,4)=26
      IE(9,5)=40
      IE(9,6)=60
      IE(9,7)=85
      IE(9,8)=141
```

```
      IE(10,1)=9
      IE(10,2)=11
      IE(10,3)=14
      IE(10,4)=20
      IE(10,5)=31
      IE(10,6)=46
      IE(10,7)=67
      IE(10,8)=92
      KA=100
 4444 FLAM=100*LAM(J)
      ILAM=FLAM
      ILAM1=ILAM+1
      A=FLAM-ILAM
      IEX=A*IE(ILAM,N)+(1-A)*IE(ILAM1,N)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  CAIFX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE CALCX(ISDA,ILDA,RLAMD)
      REAL LAM(5),LAMT,RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      A=0
      CO=0
      RMUDA=RLAMD*ILDA
      TM=EXP(-RMUDA)
      GO TO 5
    3 A=A+1
      TM=TM*RMUDA/A
      IF (ISDA) 81,5,5
   81 ISDA=0
    5 IF (A-ISDA) 3,3,4
    4 CI=TM*(A-ISDA)*.4
      CO=CO+CI
      IF (CI-.01) 7,7,3
    7 VA=CO
   70 FORMAT(3F10.6,I6)
      RETURN
      END
```

```
// DUP
*STORE       WS   UA   CALCX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE POOX
      REAL LAM(5),LAMT,RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      DIMENSION S(5)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      U=LAM(J)*(T-1)
      B=NOSTK(J)+1
   76 D=0
      PTM=EXP(-U)
    1 JC=0
      VB=0
      IF (D) 31,31,773
  773 CALL CALBX(D)
   31 RR=CWSTK-(B+VB-NOSTK(J))
      DO 23 K=1,5
   23 S(K)=NOSTK(K)/LAM(K)
      S(J)=9999
  100 IF (RR) 91,91,21
   21 I=1
      K=2
   11 IF (S(I)-S(K)) 32,32,8
   32 K=K+1
      IF (K-5) 11,17,17
    8 I=K
      GO TO 3
   17 RR=RR-1
      S(I)=S(I)+1/LAM(J)
      GO TO 100
   91 VT=0
      DO 20 K=1,5
      IF (K-J) 7,20,7
    7 ILDA=T+LSS
      ISDA=S(K)*LAM(K)
      RLAMD=LAM(K)
      CALL CALCX(ISDA,ILDA,RLAMD)
      VT=VT+VA
   20 CONTINUE
      VA=VT
  183 TC=VA+JC*CR
      IF (D) 18,18,19
   18 SUM=TC*PTM
      SIGP=PTM
      GO TO 2
   19 PTM=U*PTM/D
      SIGP=SIGP+PTM
      SUM=SUM+TC*PTM
      IF (SIGP-.95) 2,3,3
    2 D=D+1
      GO TO 773
    3 IF (B-NOSTK(J)-1)  37,22,37
   22 SUMOL=SUM
```

```
 80 IF (B-NOSTK(J)-CWSTK)  75,75,81
 75 B=B+1
    GO TO 76
 37 IF (SUM-SUMOL) 22,81,81
 81 B=B-1
700 RETURN
    END
```

```
// DUP
*STORE      WS   UA   POOX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE CALGX
      REAL RA(5),LAM(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      INTEGER SUMS
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      SUMS=0.
      DO 2 K=1,5
      KS=NOSTK(K)-TRS(K)
      IF (KS) 2,2,8
    8 SUMS=SUMS+KS
    2 CONTINUE
      G=SUMS+TRS(1)+TRS(2)+TRS(3)+TRS(4)+TRS(5) +CWSTK
      RETURN
      END
```

```
// DUP
*STORE        WS   UA   CALGX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE CALBX(D)
      REAL RA(5),LAM(5),LAMT
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      DIMENSION IEN(12)
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      VB=0
      JC=0
      IBL=B-NOSTK(J)
      IW=NOSTK(J)
      IT=T
      DO 33 N=1,12
      CALL CAIEX
      IEN(N)=IEX
      IF (IEX-100) 33,33,34
   34 K=N-1
      GO TO 35
   33 CONTINUE
   35 IFL=0
   30 V=D/FLOAT(T-1)
    3 IF (IW) 130,130,13
  130 IF (V*(IT-.75/LAM(J))-IBL) 5,4,4
   13 IF (V*(IT-IEN(IW)) -IBL) 5,4,4
    5 IF (IW-IFL) 6,1,6
    6 IBL=IBL+1
      IW=IW-1
      GO TO 3
    4 IT=IT-FLOAT(IBL)/V+.5
      JC=JC+1
      IBL=(IFL+1)+(IT-1)*(FLOAT(B-IFL-1))/FLOAT(T-1)+.5-IW
      IF (IBL) 27,27,28
   27 IBL=1
   28 VB=VB+IBL
C          VB IS SUM OF LITTLE B
      IF (IW-IFL) 3,1,3
    1 RETURN
      END
```

```
// DUP
*STORE      WS  UA  CALBX
// FOR
*ONE WORD INTEGERS
      SUBROUTINE ENTI(N,M)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      DIMENSION N(1)
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(NIM-2)1051,1051,1050
  100 FORMAT(32H MASTER LIST EXHAUSTED IN ENTITY)
 1051 WRITE(3,100)
      CALL EXIT
 1050 N(1)=TAILM
      TAILM=MAST2(TAILM)
      J=MAST2(TAILM)
      NIM=NIM-2
      MAST1(TAILM)=M
      MAST2(TAILM)=TAILM
      TAILM=J
      IM=N(1)
      MAST1(IM)=-1
      WRITE(1,453)NIM
  453 FORMAT(' NIM=',I4)
      RETURN
      END
```

```
// DUP
*STORE        WS   UA   ENTI
// FOR
*ONE WORD INTEGERS
      SUBROUTINE SIMO
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2          ,STATE,NAME,TIMES,CLOCK,MEMBE
      I=0
    1 FORMAT(I4)
      DO 1001 J=1,199
 1001 MAST2(J)=J+1
      NIM=198
      TAILM=1
      WRITE(1,1) I
      STATE=2
      NAME=1
      WRITE(1,1) I
      CALL QUEUE(TIMES)
      WRITE(1,1) I
      RETURN
      END
```

```
// DUP
*STORE        WS   UA   SIMO
// FOR
*ONE WORD INTEGERS
      INTEGER FUNCTION MEMN(IE)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      K=MAST2(IE)
      MEMN=MAST2(K)
      RETURN
      END
```

```
// DUP
*STORE       WS   UA   MEMN
// FOR
*ONE WORD INTEGERS
      INTEGER FUNCTION SIZE(L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      SIZE=MAST1(L)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SIZE
// FOR
*ONE WORD INTEGERS
      SUBROUTINE QUEUE(L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF (NIM)1010,1010,1011
  100 FORMAT(29H MASTER LIST EXHAUSTED IN SET)
 1010 WRITE(3,100)
      CALL EXIT
 1011 L=TAILM
      J=L
      TAILM=MAST2(J)
      NIM=NIM-1
      MAST1(J)=0
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  QUFUE
// FOR
      INTEGER FUNCTION HEAD (L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(MAST1(L))3,1,2
    1 WRITE(3,100)
  100 FORMAT(18H HEADOF EMPTY LIST)
      CALL EXIT
    3 WRITE(3,200)
  200 FORMAT(14H HEADOF ENTITY)
      CALL EXIT
    2 K=MAST2(L)
      K=MAST2(K)
      HEAD=MAST1(K)
      RETURN
      END
```

```
// DUP
*STORE       WS   UA   HEAD
// FOR
*ONE WORD INTEGERS
      INTEGER FUNCTION TIMV(N)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE .
      TIMV=-MAST1(N)-1
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  TIMV
// FOR
*ONE WORD INTEGERS
      INTEGER FUNCTION REFN(IE)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      K=MAST2(IE)
      REFN=MAST1(K)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  REFN
// FOR
*ONE WORD INTEGERS
      SUBROUTINE BEHE(L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2         ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(MAST1(L))1030,1030,1031
  100 FORMAT(18H BEHEAD EMPTY LIST)
 1030 WRITE(3,100)
      CALL EXIT
 1031 K=MAST2(L)
      J=MAST2(K)
      MAST2(K)=MAST2(J)
      MAST2(J)=TAILM
      TAILM=J
      NIM=NIM+1
      MAST1(L)=MAST1(L)-1
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  BEHE
// FOR
*ONE WORD INTEGERS
      SUBROUTINE ADDL(M,L)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(NIM)1070,1070,1071
  100 FORMAT(33H MASTER LIST EXHAUSTED IN ADDLAST)
 1070 WRITE(3,100)
      CALL EXIT
 1071 J=TAILM
      TAILM=MAST2(J)
      NIM=NIM-1
      MAST1(J)=M
      IF(MAST1(L))1072,1073,1074
  101 FORMAT(18H ADDLAST TO ENTITY)
 1072 WRITE(3,101)
      CALL EXIT
 1073 MAST2(J)=J
      MAST2(L)=J
      MAST1(L)=1
      GO TO 1075
 1074 MAST1(L)=MAST1(L)+1
      K=MAST2(L)
      MAST2(J)=MAST2(K)
      MAST2(K)=J
      MAST2(L)=J
 1075 RETURN
      END
```

```
// DUP
*STORE      WS  UA  ADDL
// FOR
*ONE WORD INTEGERS
      SUBROUTINE DELE(M,IS)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
   40 J=MAST1(IS)
      DO 1080 I=1,J
   42 K=MAST2(IS)
   44 K=MAST2(K)
   46 IF(MAST1(K)-M)1080,1081,1080
 1080 MAST2(IS)=K
  100 FORMAT(29H MEMBER NOT PRESENT IN DELETE)
      WRITE(3,100)
      CALL EXIT
 1081 K=MAST2(IS)
      KK=MAST2(K)
      MAST2(K)=MAST2(KK)
      MAST2(KK)=TAILM
      TAILM=KK
      NIM=NIM+1
      MAST1(IS)=J-1
      IF(I-1)1084,1082,1084
 1084 I=J-I
      IF(I)1085,1082,1085
 1085 DO 1083 K=1,I
      KK=MAST2(IS)
 1083 MAST2(IS)=MAST2(KK)
 1082 RETURN
      END
```

```
// DUP
*STORE       WS   UA   DELF
// FOR
* ONE WORD INTEGERS
      SUBROUTINE SYRY
      REAL LAM(5),LAMT
      REAL RA(5)
      INTEGER KCHEC(5)
      INTEGER TAILM,STATE,TIMES,ONORD,SYPRO(2),CWSTK,DEM(5,2),G,B,VB,VC,
     1CLOCK,MAST1(200),MAST2(200),NOSTK(5),TRS(5),T
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      COMMON CBO,B,CWSTK,DEM,F,G,ISTRE,J,JC,KAUNT,LAM,LAMT,LC,LSS,M,KA,N
     1,NOSTK,ONORD,RA,SYPRO,VA,VB,VC,Y,Z,KCHEC,CR,TRS,    NOPRC,IEX,T
      F=NOSTK(1)+NOSTK(2)+NOSTK(3)+NOSTK(4)+NOSTK(5)+CWSTK
      IF (F-M) 80,80,111
   80 CALL SETT   (SYPRO (NAME),(LC-LSS+CLOCK))
      CALL ADDL   (SYPRO (NAME),TIMES)
      NOPRC=NOPRC+1
      SYPRO (STATE)=ONORD
      WRITE(1,1) CLOCK,NOPRC
    1 FORMAT(2I6)
  111 RETURN
      END
```

```
// DUP
*STORE       WS   UA   SYRY
// FOR
*ONE WORD INTEGERS
      SUBROUTINE SCA(IS,IM,IL)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2        ,STATE,NAME,TIMES,CLOCK,MEMBE
  456 NIS=MAST1(IS)
      IF(NIS)1090,1090,1091
  100 FORMAT(15H SCAN EMPTY SET)
 1090 WRITE(1,100)
      CALL EXIT
 1091 IP=MAST2(IS)
      IL=-29999
      DO 1092 J=1,NIS
      M=MAST1(IP)
      L=MAST1(M)
      IF(L-IL)1092,1092,1093
 1093 IL=L
      IM=M
 1092 IP=MAST2(IP)
      IL=-1-IL
      RETURN
      END
```

```
// DUP
*STORE       WS   UA   SCA
// FOR
*ONE WORD INTEGERS
      SUBROUTINE SETT(N,M)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      COMMON NIM,TAILM,MAST1,MAST2       ,STATE,NAME,TIMES,CLOCK,MEMBE
      MAST1(N)=-M-1
  213 FORMAT(3I10)
      RETURN
      END
```

```
// DUP
*STORE      WS  UA  SETT
// FOR
*ONE WORD INTEGERS
      SUBROUTINE GROU(IF,IN,IL)
      INTEGER NIM,TAILM,MAST1(200),MAST2(200),STATE,TIMES,CLOCK
      DIMENSION IE(1,1)
      COMMON NIM,TAILM,MAST1,MAST2      ,STATE,NAME,TIMES,CLOCK,MEMBE
      IF(NIM-2*IN)1150,1151,1151
 1150 WRITE(3,100)
  100 FORMAT(37H MASTER LIST EXHAUSTED IN GROUPENTITY)
      CALL EXIT
 1151 DO 1152 J=1,IN
      IE(J,1)=TAILM
      TAILM=MAST2(TAILM)
      K=MAST2(TAILM)
      MAST1(TAILM)=IL
      MAST2(TAILM)=J
      TAILM=K
      K=IE(J,1)
 1152 MAST1(K)=-1
      NIM=NIM-2*IN
      RETURN
      END
```

```
// DUP
*STORE        WS   UA   GROU
```