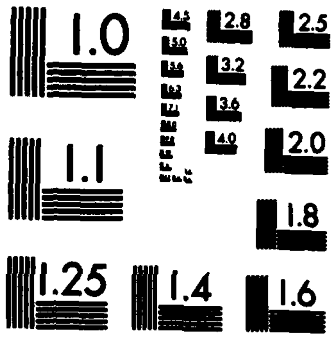
AD-A181 487 AN EVALUATION OF A JOINT REPLENISHMENT INVENTORY MODEL WITH RANDOM DEMANDS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W B KIM MAR 87 1/1

UNCLASSIFIED F/G 15/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A181 487

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC FILE COPY

DTIC
ELECTE
JUN 23 1987
S D
D

THESIS

AN EVALUATION
OF
A JOINT REPLENISHMENT INVENTORY MODEL
WITH RANDOM DEMANDS

by

Kim, Won Bong

March 1987

Thesis Advisor Francis R. Richards

Approved for public release; distribution is unlimited.

87

REPRODUCED AT GOVERNMENT EXPENSE

AD-A181 487

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
AN EVALUATION OF A JOINT REPLENISHMENT INVENTORY MODEL WITH RANDOM DEMANDS.

12 PERSONAL AUTHOR(S)
KIM, Won Bong

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1987 March	71

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Inventory, Joint Replenishment, Random Demands

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper considers a joint replenishment inventory problem with a continuous-review (S, c, s) policy for the backorder case with Poisson demands and constant procurement lead times.

Whenever item i's inventory level hits s_i (reorder point) or lower it triggers an order so as to raise item i's level to S_i (order up point). At the same time any other item j with inventory level at-or-below its can-order point c_j is included in the replenishment.

A Poisson demand model with a queueing description of the system's operation is analysed, and comparisions are conducted for joint versus individual orders in the case of multi-item problems, where joint replenishment of several items may reduce setup costs.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION
22a NAME OF RESPONSIBLE INDIVIDUAL Francis R. Richards	22b TELEPHONE (Include Area Code) 408-646-2543 / 22c OFFICE SYMBOL 55Rh

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release; distribution is unlimited.

An Evaluation of A Joint Replenishment Inventory Model with Random Demands

by

Kim, Won Bong
Lieutenant Commander, Republic of Korea Navy
B.S., R.O.K. Naval Academy, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1987

Author: 김 원 봉
Kim, Won Bong

Approved by: F R Richards
Francis R. Richards, Thesis Advisor

J. D. Esary
James D. Esary, Second Reader

Peter Purdue
Peter Purdue, Chairman,
Department of Operations Research

Kneale T. Marshall
Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

This paper considers a joint replenishment inventory problem with a continuous-review (S, c, s) policy for the backorder case with Poisson demands and constant procurement lead times.

sub i

Whenever item i's inventory level hits s_i (reorder point) or lower it triggers an order so as to raise item i's level to S_i (order up point). At the same time any other item j with inventory level at-or-below its can-order point c_j (sub j) is included in the replenishment.

A Poisson demand model with a queueing description of the system's operation is analysed, and comparisions are conducted for joint versus individual orders in the case of multi-item problems, where joint replenishment of several items may reduce setup costs. (Keywords: theses; mathematical models; computer programs.)

Accesion For	
NTIS CRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

QUALITY INSPECTED 4

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. DESCRIPTION OF THE PROBLEM

The purpose in constructing a mathematical model of an inventory system is to use it as an aid in developing a suitable operating doctrine for the system. The criterion most frequently used for selecting the operating doctrine is that of profit maximization or cost minimization. In some cases the task of determining the optimal operating doctrine is so difficult that it is either impossible or uneconomical to determine the optimal doctrine, and instead, one optimizes with respect to some subset of operating doctrines. Occasionally, the mathematical model may be so complicated that it is extremely difficult to do anything analytically. In such situations, simulations are used to study various operating doctrines.

We are concerned with finding optimal ordering policies for a multi-item inventory system in which the inventory position is under continuous review. In this thesis, the times at which demands occur are assumed to be generated by independent Poisson processes with intensities λ_i (annual demand rate for item i). We assume that all shortages are backordered and will be satisfied from the next shipment.

We assume that a holding cost of H_i per unit time is charged for each unit of item i in inventory. When an order is placed, a setup cost for an independent replenishment of item i is assumed to be $C_i = A + a_i$, where A is a fixed cost and a_i is a cost which depends on the item ordered. The cost of a joint replenishment of both item i and j is $A + a_i + a_j$. For instance, if item i triggers a replenishment, the setup cost of $A + a_i$ is applied to item i, and a_j to the jointly replenished item j.

This type of cost structure is particularly appropriate when a group of items is ordered from the same supplier, or if a group of items uses the same means of transportation, and transportation costs do not increase proportionally when the quantity transported is increased. Thus, when a replenishment is made, there is a major fixed cost independent of the number of items involved and a variable cost that depends upon the number of items [Ref. 1: pp. 278-83].

Because of this cost structure, it is clear that there may be an opportunity to reduce total variable costs if one takes advantage of joint replenishment.

B. SCOPE

Our interest in this problem stems from the "random joint order policy" proposed by Balintfy [Ref. 2], and from an (S,c,s) policy treated by Silver [Ref. 3]. For the backorder case with Poisson demands and constant lead times, and the specified customer service levels, we will compare the individual order policy versus joint order policy.

However, for selecting the control parameters of (S,c,s) in Joint Order Policy, we have considered slightly different procedures suggested by Schacck and Silver [Ref. 4], and Silver [Ref. 3].

Two programs were written in order to conduct the simulation experiments. The first program determines the control parameters of each item in the group for both independent and joint replenishment policies. A second program evaluates the model. Detailed user instructions and the program itself are contained in Appendix A and B, while examples of their use are presented in Chapter II.

The individual order policy is demonstrated only for the purpose of comparison. For the sake of simplicity, we compare the joint order policy to the individual order policy where the classical lot size formula is used for each item.

II. GENERAL FORMULATION OF THE MODEL

A. ASSUMPTIONS AND NOTATIONS

1. Assumptions

We are dealing with a group of items where the cost of replenishing two or more items at the same time is less than the total cost of replenishing the same number in separate individual replenishments. The assumptions of the model are:

1. There is fixed cost A associated with each replenishment, and variable cost a, associated with each item involved in the replenishment.
2. Demands for item i are Poisson distributed with parameter λ_i.
3. There are no quantity discounts.
4. Inventory holding costs are proportional to the average dollar value of inventory.
5. Service level is defined in one of two ways: probability of no shortage per replenishment cycle, and fraction of demand to be satisfied directly from the shelf.

2. Basic Notation

We use the following notation:

A : fixed cost per replenishment. (independent of the number of items and units involved)

a_i : fixed cost for including item i in a replenishment.

C_i : ordering cost per order: ($C_i = A + a_i$)

EC_i: expected relevant annual inventory cost for item i.

P_1 : desired customer service level, measured in terms of the probability of no shortage per replenishment cycle.

P_2 : desired customer service level, measured in terms of the fraction of demand satisfied directly from stock on hand.

L : replenishment lead time in years.

r : inventory holding cost rate.

v_i : standard unit price of item i.

s_i : reorder or must-order point of item i.

c_i : can-order point of item i.

S_i : order-up-to point of item i.

Q_i : order quantity of item i.

λ_i : Poisson demand rate for item i, in pieces per year.

I_i : average on hand inventory level of item i.

n : number of items in the replenishment group.

N_i : expected number of replenishments which involve item i per year.

NT_i : expected number of replenishments triggered by item i per year.

Other notation will be introduced as they needed.

B. INVENTORY THEORY BACKGROUND

Before considering the coordinated control problem it might be helpful to briefly discuss the related single item models whose solution will be a key element in determing the parameters of the (S, c, s) control system.

1. Economic Order Quantity Model

The E.O.Q. model is based on the assumption that the entire lot is added to stock at one time, and that the stock will be withdrawn at a constant rate, and no stockouts are permitted.

Total annual variable cost = Ordering cost + Holding cost

$$EC_i = \lambda_i C_i / Q_i + H_i Q_i / 2 \qquad (2.1)$$

The annual order cost is obtained by multiplying the number of orders per year (λ_i/Q_i) by the cost of placing an order (C_i).

The average annual holding cost for item i is the average inventory ($Q_i/2$) times the annual unit holding cost ($H_i = v_i r$).

To obtain the minimum cost lot size (E.O.Q.), take the first derivative of total annual cost with respect to Q_i and set it equal to zero. Solving this for Q_i, we get the E.O.Q. formula:

$$Q_i^* = \sqrt{[2C_i\lambda_i / H_i]} \qquad (2.2)$$

Once the economic order quantity is known, the expected number of orders placed during the year, n, can be determined:

$$n_i = \lambda_i / Q_i^*$$

2. Exact Model for the Backorders case with Poisson Demands and Constant Lead Times

If demands are random, the probability of stockouts exists, and we must include a stockout cost in the expression for expected annual variable costs. This expression is given in [Ref. 5: eqn. (4-61)] to be

$$EC_i = \lambda_i C_i / Q_i + H_i (Q_i / 2 + 1/2 + s_i - M_i) + \varphi_i E(Q_i, s_i) \tag{2.3}$$

$$+ (\tau_i + H_i) B(Q_i, s_i)$$

where

φ_i : the cost of backorder per unit quantity of backorder.

τ_i : the cost of backorder which is proportional to the length of backorder time.

M_i : expected lead time demand for item i in pieces, $M_i = \lambda_i L$.

H_i : holding cost of item i.

$E(Q_i, s_i)$: expected number of backorders incurred per year for item i.

$B(Q_i, s_i)$: the steady state expected number of backorders for item i.

It is not easy to derive the optimal Q_i and s_i by use of the above exact expression for EC_i. Fortunately, it turns out in practice that it is seldom necessary to use the exact formulation, except for the case when it costs very little to incur backorders. The approximation discussed in the next section has been shown to give good results when backorders are infrequent.

3. Approximate Form with Known Backorder Cost

The average annual variable cost equation can be approximated as follows [Ref. 6: pp. 59 - 62]:

$$EC_i = \lambda_i C_i / Q_i + H_i (Q_i / 2 + s_i - M_i) + \varphi_i (\lambda_i / Q_i) E(M_i > s_i) \tag{2.4}$$

where

φ_i : the cost of backorder per unit quantity of backorder.

$E(M_i > s_i)$:the expected number of backorders at the end of cycle.

To minimize EC_i, take the partial derivative of the EC_i with respect to Q_i and s_i and set these equal to zero.

$$dEC_i / dQ_i = - \lambda_i C_i / Q_i^2 + H_i / 2 + \varphi_i \lambda_i E(M_i > s_i) / Q_i^2 = 0 \tag{2.5}$$

$$dEC_i/ds_i = H_i + (\varphi_i\lambda_i / Q_i)dE(M_i > s_i)/ds_i = 0 \tag{2.6}$$

The first equation(2.5) yields the optimal Q_i for a given reorder point s_i:

$$Q_i^* = \sqrt{[2\lambda_i(C_i + \varphi_i E(M_i > s_i) / H_i]} \tag{2.7}$$

The second equation(2.6) yields :

$$dE(M_i > s_i)/ds_i = -\sum P_r(M_i) \tag{2.8}$$

where

$E(M_i > s_i)$: the expected number of backorders at the end of cycle.

$-\sum P_r(M_i)$: complementary cumulative distribution.

This is the complementary cumulative distribution of the Poisson random variable, X, evaluated at s_i. Therefore, solving the second equation for s_i in terms of Q_i gives the maximum allowable probability of a shortage during a lead time :

$$F'(s_i) = (H_i Q_i)/(\varphi_i\lambda_i) \tag{2.9}$$

To find the optimal pair (Q_i, s_i) that minimize EC_i, iterative procedures can be applied, such as described in [Ref. 6: p. 61]. Hower, we do not apply this kind of procedure to find out the values of control parameters in (S, c, s) policy, because the order size in the (S, c, s) policy is not fixed.

In (S, c, s) policy, the order size is determined as an order-up-to point minus a reorder point for the item which triggers the replenishment action. Order sizes for jointly replenishable items are order-up-to points minus on-hand levels which are less than or equal to can-buy points but more than reorder points.

4. Some Useful Formulas for Obtaining the Imputed Backorder Cost / Reorder Point /Safety Quantity in Fixed Order Size System

In any practical situation, it is very difficult to determine accurately backorder costs. They can include such factors as loss of customers' goodwill(i.e., in the future, he may take his business elsewhere), or in the military supply system, the cost of having some first line weapon system inoperative because of lack of parts. Other contributions to shortage cost can be somewhat easier to measure. However, these are usually small part of the total backorder costs [Ref. 5: p.18].

We shall assume here that there is a backorder cost associated with each unit backordered.

A complete cost minimization analysis of the optimal values of the model parameters can be made only if a cost can be assigned to each shortage. As an alternative to specifying the shortage cost, one can examine a given inventory policy and obtain the imputed cost of shortage within it [Ref. 7: p. 338]. Rather than explicitly costing backorders, we will specify a service level constraint and determine the imputed stockout cost.

Whenever an organization uses a service level to establish a reorder point, it effectually establishs a stockout cost. Associated with a given service level is an imputed or implicit stockout cost. It is a simple matter to determine the imputed stockout cost for a given service level from the probability of a stockout [Ref. 8: p. 162].

a. Probability of no Shortages per Replenishment Cycle

A service level based on the frequency of service per order interval or replenishment cycle will indicate the probability of not running out of stock during the replenishment. This approach does not concern itself with how large the shortage is, but only with how often it can occur during the lead time (replenishment cycle). It is defined as the fraction of the replenishment cycles without depletion of stock:

$$\text{Service level fraction per cycle} = 1 - \frac{\text{number of order periods with a stockout}}{\text{total number of order periods}}$$
$$= 1 - P_r(M_i > s_i)$$

$$P_r(M_i > s_i) = P_r(s_i)$$
$$= 1 - (\text{service level fraction per cycle})$$

The term $P_r(s_i)$ is the stockout level fraction per order cycle or the probability of at least one stockout while awaiting a supplier's delivery. It is also a measure of the fraction of lead time periods during which the demand will exceed the reorder point. The magnitude of the stockout is ignored with this approach.

The following formula is developed to compute the imputed backorder cost:

$$P_r(s_i) = (H_i Q_i) / (b_i \lambda_i) \qquad (2.10)$$

where

$P_r(s_i)$: probability of stockout for item i.

H_i : holding cost of item i, ($H_i = v_i r$).

b_i : imputed backorder cost of item i.

To obtain the appropriate stockout cost, solve eqn.(2.10) for b_i. An example will illustrate the procedure in a later section.

b. Fraction of Demands Satisfied Directly from the Shelf

The fraction of units demanded and immediately filled from the shelf can be defined as the ratio: (number of units supplied)/(total number of units demanded) which is equal to {1 - [(number of units short)/(total number of units demanded)]}.

The above relationships must be measured over some fixed time period such as the order interval.

For example, to obtain the stockout level fraction for units demanded:

$$\text{Stockout level fraction for units} = \frac{E(M_i > s_i)}{\text{quantity demanded during a cycle}}$$

$$E(M_i > s_i) \,/\, Q_i = (H_i Q_i) \,/\, (b_i \lambda_i) \tag{2.11}$$

where

$E(M_i > s_i)$: expected number of backorders for item i during the cycle.

H_i : holding cost of item i, ($H_i = v_i r$).

b_i : imputed backorder cost for item i.

c. Safety Stock

With backorders, there is no loss of sales since the customer waits for the arrival of the next order if stock is not available. The expected safety stock is defined as:

$$\text{Safety stock} = \sum (s - M)\, P_r(M) \tag{2.12}$$

$$= s \sum P_r(M) - \sum M\, P_r(M) = s - E(M)$$

where

$E(M)$: the expected lead time demand.

The number of backorders per lead time $= 0$, if $M < s$.

$= M - s$, if $M > s$.

So, the safety stock is simply the reorder point minus the average lead time demand. The reorder point is determined so that the customer service objective is satisfied.

C. MATHEMATICAL STATEMENT OF THE DECISION PROBLEM FOR THE JOINT REPLENISHMENT MODEL

1. Cost Equation

The basic cost equation in the joint replenishment model is as follow [Ref. 3]:

$$EC_i = I_i v_i r + NT_i A + N_i a_i \tag{2.13}$$

The first term represents the inventory holding costs and the other two represent the set up charges allocated to item i. The total expected costs per year of the group of items are given by

$$EC = \sum EC_i = \sum (I_i v_i r + NT_i A + N_i a_i) \tag{2.14}$$

2. Service Level Constraint

The service constraint takes one of two forms depending upon the measure of service used.

(1) Probability of no shortage per replenishment cycle

This is simply the probability that demand for item i during the lead time will exceed the reorder point.

$$P_r(s) = P_r(M_i > s_i) = \sum P_r(M_i) \tag{2.15}$$

where

$P_r(s)$: probability of stock out.

M_i : lead time demand for item i.

(2) Fraction of demand satisfied directly from the shelf

The expected stockout quantity during the lead time is given by

$$E(M > s_i) = \sum (M_i - s_i) P_r(M_i) \tag{2.16}$$

where

$E(M_i > s_i)$: expected number of stock out for item i in pieces during the lead time

M_i : lead time demand for item i .

A backorder occurs if and only if the lead time demand (which is Poisson with parameter M_i) is greater than the inventory level of the item when the order is placed.

D. INDIVIDUAL ORDER POLICY

Under the the individual order policy, there is a fixed order quantity for item i that is ordered every time the reorder point is reached .

Safety stock is needed to protect against stockouts between the time the reorder point is reached and the order received. As presented in [Ref. 3], the individual order policy (called independent control) is a special case of the joint replenishment policy (S, c, s) where $c_i = s_i$. In this case item i is ordered only when the inventory position hits its reorder point, and the order quantity is always of size $Q_i = S_i - s_i$. No other items will be included in the replenishment, hence a set up cost of $A + a_i$ is incurred with each replenishment.

As mentioned in the introduction, the main focus of this thesis is to compare joint replenishment with independent control to determine the cost savings that can be achieved.

Let us first consider independent control. Then, $c_i = s_i$, and the optimization problem is:

$$\text{Minimize } EC_i = (S_i - s_i + 1)/\,2 + (s_i - \lambda_i L)v_i r \qquad (2.17)$$

$$+ (A + a_i)\lambda_i \,/\, (S_i - s_i)$$

subject to

$$P_r\,(\,X \le s_i \mid \lambda_i L) \ge P_1 \qquad (2.18)$$

or

$$1 - E(\,X > s_i \mid \lambda_i L)\,/\,Q_i \ge P_2 \qquad (2.19)$$

where, X is the lead time demand with a Poisson distribution having mean $\lambda_i L$.

The left hand side of eqn.(2.18): $P_r(X \leq s_i | \lambda_i L)$ is the probability of no shortage per replenishment cycle.

In the left hand side of eqn.(2.19), the term $E(X > s_i)/Q_i$ is the stockout fraction of units demanded. Therefore $1 - E(X > s_i)/Q_i$ is the fraction of demand satisfied directly from the shelf.

The exact solution to this problem requires that the pair (S_i, s_i) which minimize eqn.(2.17) subject to either eqn.(2.18) or eqn.(2.19) be found for each item i. However, it have been observed by many researchers (see: for example Silver and Wilson [Ref. 9]) that a much simplier sequential approach does almost as well from the standpoint of minimizing costs. This sequential approximation approach is what we describe.

1. Applied Solution Procedure

Step1 : Let $Q_i = S_i - s_i$. Find the value of Q_i which minimizes EC_i ignoring the service level constraint. Then given this value of $S_i - s_i$, the constraint is used to find the lowest allowable value of s_i. This first step leads to the usual EOQ expression $Q_i = S_i - s_i = \sqrt{[\, 2(A + a_i)\lambda_i / H_i]}$.

Step2 : Determine the reorder points.

(1)For the service measure, probability of no shortage per replenishment cycle, find the smallest value of s_i, which satisfies the condition of eqn.(2.18):

$$P_r (X \leq s_i \mid \lambda_i L) \geq P_1$$

(2)For the service measure, fraction of demands satisfied directly from the shelf, find the smallest value of s_i, which satisfies the following inequality:

$$\lambda_i L - s_i - \lambda_i L\, P_r (X \leq s_i - 1 \mid \lambda_i L) + s_i P_r (X \leq s_i \mid \lambda_i L) \leq Q_i(1 - P_2)$$

Step3 : Set $S_i = s_i + Q_i$.

Step4 : Compute the average annual variable costs.

$EC_i = (S_i - s_i + 1)/ 2 + (s_i - \lambda_i L)v_i r + (A + a_i)\lambda_i / (S_i - s_i)$ using the values s_i and S_i estimated from Steps 2 and 3. A computer program for solving for these parameters is provided in Appendix A.

Note that we can now obtain imputed stockout costs.

Step5 : Compute the imputed stockout cost

(1)For the service measure, probability of no shortage per replenishment cycle: Use eqn.(2.10): $P_r(s) = (H_iQ_i) / (b_i\lambda_i)$ and solve for b_i, $b_i = H_iQ_i / (1 - P_1)\lambda_i$.

(2)For the service measure, fraction of demands satisfied directly from the shelf, it is necessary to solve for $E(M > s_i)$ and find the associated $P_r(s)$ before using eqn.(2.11). It can be done alternatively by finding the complementary cumulative distribution of the lead time demand, introduced by eqn. (2.8). Then eqn.(2.11) yields $b_i = H_iQ_i / [-\sum P_r(M_i)] / \lambda_i$.

2. Numerical Examples

Repeated from Silver's example in his paper [Ref. 3].

Example 1. Service Level = Probability of no shortage per replenishment cycle.

Consider the following example involving four items:

$A = \$50$

$a_i = \$10$ for $i = 1, 2, 3, 4$

$r = 0.2\ \$ / \$ / \text{yr}.$

$L = 1$ month (i.e. 1/12 year)

TABLE 1

INPUT DATA

item (i)	λ_i (pieces/yr)	v_i ($/piece)	P_1 (desired service level)
1	290	6.90	0.95
2	41	1.20	0.95
3	77	3.90	0.95
4	122	2.30	0.95

Use of eqn. (2.15), (2.17), (2.18) gives the results shown in Table 2.

Example 2. Service Level = Fraction of demand satisfied directly from the shelf.

Consider the same data as example 1, but with a new the measure of service level. Suppose the specified fraction of demands to be satisfied directly is 0.99. The results shown in Table 3.

TABLE 2
INDEPENDENT ORDER POLICY: SERVICE LEVEL 95% (P_1)

Item	LT demand	EOQ (integer)	Reorder	TVC(S)	S/O cost
1	24.167	158.800 (159)	33	232.024	15.113
2	3.417	143.178 (143)	7	35.343	16.762
3	6.417	108.840 (109)	11	88.860	22.051
4	10.167	178.399 (178)	16	84.977	13.453

TOTAL : 441.203

TABLE 3
INDEPENDENT ORDER POLICY: SERVICE LEVEL 99% (P_2)

Item	LT demand	EOQ (integer)	Reorder	TVC(S)	S/O cost
1	24.167	158.800 (159)	25	220.984	1.983
2	3.417	143.178 (143)	3	34.383	1.882
3	6.417	108.840 (109)	7	85.740	3.499
4	10.167	178.399 (178)	10	82.217	1.536

TOTAL : 423.323

E. JOINT ORDER POLICY

1. Introduction

Let us now consider the possibility of joint replenishments. As discussed earlier, we can reduce ordering setup costs by combining the orders of several items whenever an order is placed. This may result in increased holding costs but the savings in ordering costs may be greater than the increase in holding costs resulting in a net decrease in total costs. Further, it is likely that the occurance of stockouts will be reduced since we will frequently reorders some items before the inventory levels hit the normal reorder points for those items.

We do not seek to determine the "optimal" joint replenishment policy. Rather, we restrict attention to a simple, intuitively appealing, policy and seek to find the optimal values of parameters for the selected policy. The policy we consider is the (S, c, s), "can buy, must buy" policy which requires an order to be placed for item i up to the level S_i any time the inventory position for item i reaches or falls below s_i. In addition, however, we order any other item i whose inventory position is below the can buy point c_i, we always order up to the level S_i.

2. General Concepts for Obtaining the Values of Control Parameters

Suppose a single item i, is in a group of coordinated items and assume initially that the replenishment lead time is zero. The assumption of unit sized Poisson demands together with a zero lead time implies that $s_i = 0$.

Now item i, from time to time, will be faced with the opportunity of a replenishment at the reduced setup cost of only "a_i". Such an event is caused by another item triggering a replenishment. Occasionally item i will trigger an order when its inventory hits the zero level. Under such circumstances we would like to determine the (S, c) pair which minimizes the expected costs per unit time.

Silver shows that the opportunities to replenish at the reduced cost occur probabilistically according to a Poisson process with a rate μ per year, where μ is the expected number of orders triggered per year by all other items in the group.

Let NT_i be the number of orders triggered per year by item i. Then μ_j is given by $\mu_j = \sum NT_i$, where the summation is taken over all $i \neq j$.

For example, if there are four items in the group and we are considering item 2, we would have $\mu_2 = NT_1 + NT_3 + NT_4$.

It is very difficult to determine the value of μ_i algebraically. We will consider some heuristic methods to approximate this.

Silver [Ref. 3: eqn. 13] determines that the relevant cost equation (the subscripts i have been suppressed) is given by:

$$EC_i = \{S - c + \rho(1 - \rho^c)/(1 - \rho)\}^{-1} \{(S - c)(S + c + 1)\, vr/2 \qquad (2.20)$$

$$+ \rho[c - \rho(1 - \rho^c)/(1 - \rho)]\, vr/(1 - \rho) + \lambda \rho^c A + \lambda a\}$$

where

$$\rho = \lambda/(\lambda + \mu)$$

The algorithm suggested by Silver in [Ref. 3] is a simultaneous optimization procedure to minimize eqn. (2.17) with respect to S_i, c_i, s_i. The procedure is iterative.

Solution procedures to determine the control parameters of the (S, c, s) policy are explained in the next chapter.

III. DEVELOPMENT OF THE MODEL

In this chapter we consider the general joint replenishment problem with positive leadtimes. We obtain values of the parameters s_i, c_i, and S_i using an iterative procedure recommended in [Ref. 3]. Silver [Ref. 10] derives the following expression for the total expected costs per year for item i (we suppress the item subscript for simplification of notation):

$$EC_i = \{S - c + \rho (1 - \rho^c) / (1 - \rho)\}^{-1} \{(S - c)(S + c + 1) vr / 2 \tag{3.1}$$

$$+ \rho [c - \rho (1 - \rho^c)/(1 - \rho)]vr/(1 - \rho) + \lambda \rho^c A + \lambda a\}$$

where

$$\rho = \lambda/(\lambda + \mu)$$

ρ is the probability that a particular occurance is either a demand or an opportunity to replenish at reduced cost.

A. SOLUTION PROCEDURE

We recommend as a solution procedure a somewhat modified version of the procedure suggested by Silver.

1. Algorithm

Step 1 : Initialization

Use the values computed by the independent control to initialize. Let $k = 1$, where k = index of iteration.

$c_{min} = 0.$

$c_{max} = Q^*$

$EC(k)$ = EC(determined from the independent control)

$N(k) = \lambda / Q^*$

$\mu_j(k) = \sum N_i$, where all $i \neq j$, and $i, j = 1, ..., n$

$\rho(k) = \lambda / (\lambda + \mu)$

$c(k) = 0$

$S(k) = Q^*$

Step 2 : Let $k = k + 1$

$$c(k) = c(k\text{-}1) + (c_{max} - c_{min}) / 2, \text{ if } EC(k) < EC(k\text{-}1)$$
$$c(k) = c(k\text{-}1) - (c_{max} - c_{min}) / 2, \text{ if } EC(k) > EC(k\text{-}1)$$
$$S(k) = c(k) - \rho (1 - \rho^c)/(1 - \rho) + \sqrt{} \{2 \lambda (a + A \lambda^c)/vr$$
$$+ 2 c \lambda^{c+1}/(1 - \rho) - \rho (1 - \rho^c)(1 + \rho^{c+1})/(1 - \rho)^2\}$$
$$EC(k) = \{\rho (1 - \rho^c)/(1 - \rho)\}^{-1}\{\rho [c - \rho (1 - \rho^c)/(1 - \rho)]vr$$
$$/(1 - \rho) + \lambda \rho^c A + \lambda a\}$$

Step 3 : Test for convergence of c and S.

Round the c and S values to the nearest integer.

If $c(k) = c(k\text{-}1)$ and $S(k) = S(k\text{-}1)$, go to Step 4, else go to Step 2. And, repeat until S and c converge.

Step 4 : Repeat Steps 1, 2 and 3 for $i = 1, 2, ..., n$.

Step 5 : Let $N_i = \lambda \rho^c/\{S - c + \rho (1 - \rho^c)/(1 - \rho)\}$.

ρ values are from the previous iteration and S and c values are from step 4.

Step 6 : Repeat steps 1-5 until $|c(k) - c(k\text{-}1)| < \delta$ and $|S(k) - S(k\text{-}1)| < \delta$.

Step 7 : Now determine the must-buy points, s_i, for $i = 1, ..., n$ from the service constraints.

(a) For Service Measure 1: probability (P_1) of no shortage per cycle, find the smallest value s such that

$$(1/\rho)P_r(X \le s + c|\lambda L) - P_r(X = s + 1|\lambda L)$$
$$- \rho^{s+1} \sum P_r(X = x_0|\lambda L) (1/\rho)^{x_0} \ge P_1/\rho^c$$

(b) For Service Measure 2: fraction (P_2) of demand to be satisfied directly from shelf, find the smallest value s such that

$$\rho^c \{\lambda L - s \lambda L P_r (X \le s - 1 | \lambda L) + s P_r(X \le s|\lambda L)\}$$
$$+ (1 - \rho)^{c+s} \sum \rho^{-w_0} \{\lambda L - w_0 - \lambda L P_r(w_0 - 1|\lambda L)$$
$$+ w_0 P_r(X \le w_0)\}$$
$$\le \{S - c + \rho (1 - \rho^c)/(1 - \rho)\} (1 - P_2)$$

Step 8 : Recompute the values S_i and c_i.

$$S_i = S + s_i$$
$$c_i = c + s_i$$

Stop

2. Flowchart

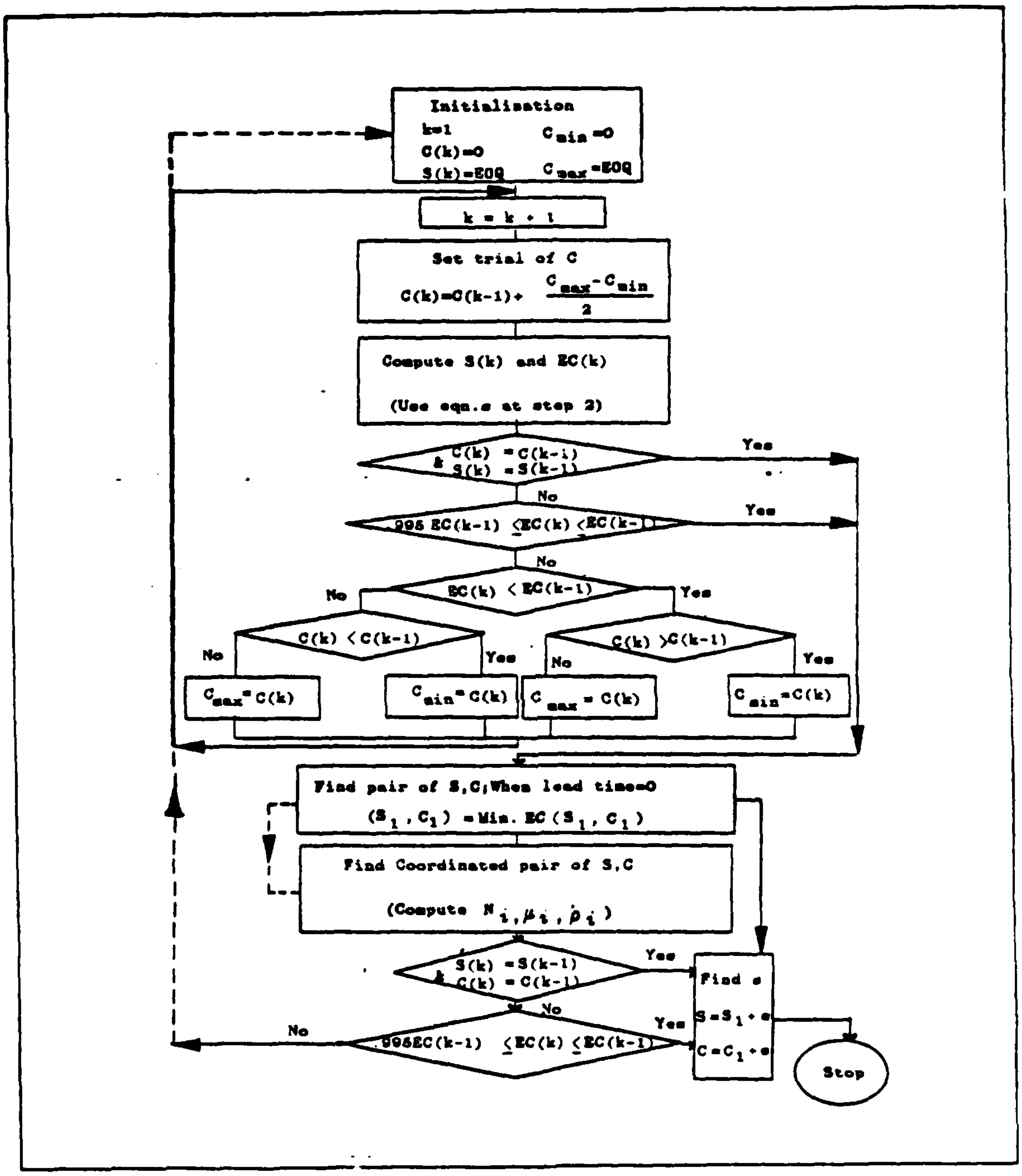


Figure 3.1 Determination of coordinated pair (S,c).

IV. SIMULATION STUDY

In order to determine how costs of ordering, holding, shortage, etc. are affected by joint replenishment, a simulation model was developed. The purpose of the simulation study was to test the cost savings by comparing the costs for the two inventory policies, independent and joint replenishment. The simulation was written in FORTRAN77.

A. ASSUMPTIONS FOR THE SIMULATION STUDY

1. Poisson demands (exponential interarrival times).
2. Items demanded one at a time.
3. Constant lead times (independent of the size of the order and the number of items in a replenishment).
4. For the joint replenishment case, place an order whenever any item hits its must-buy point s_i and order up to S_i. Also, include item $j \neq i$ if the inventory position for item j is below the can-buy point c_j. Order up to S_j.
5. For the independent control, order up to level S_i whenever the stock on hand for item i reaches s_i.
6. For a given item, there is never more than a single order outstanding in the joint replenishment case.

B. MODEL INPUTS

1. Number of items.
2. Demand rate λ_i for each item.
3. Holding cost rate v_i for each item.
4. Lead time (in months) L.
5. Type of service level constraint (Type1, or Type2)
6. Service level required.
7. Unit price UP_i.
8. Group ordering cost A.
9. Individual ordering cost a_i.
10. Stockout cost rate per unit stockout (percentage of unit price).
11. Time weighted stockout cost (percentage of unit price).
12. S_i, c_i, s_i for each item.
13. Model options (Independent control or Joint Replenishment, etc.)
14. Limits of simulation period, NSIM.

C. MODEL OUTPUTS

1. Achieved service level for each item.
2. Number of orders (independent and joint).
3. Total ordering cost for each item.
4. Total holding cost for each item.
5. Total unit years of backorders.
6. Total number of backorders.
7. Total unit years of stock held.
8. Total stockout cost (fixed and time weighted) for each item.
9. Total annual costs for each item.
10. Total annual variable costs for all items.
11. Standard deviation of total annual variable costs.
12. A complete audit trail of all orders placed (optional).

D. STRUCTURE OF THE PROGRAM

1. Algorithm

Step 1 : Read input data, process options and initialize variables.

Step 2 : Determine the earliest event time and type of event.

Let TDA_i be the time of the next demand for item i and TOA_i be the next time of the for item of an order for item i.

Set the master clock time, CT, to the minimum of $\{TDA_i, TOA_i\}$.

If the event type is a demand for item i then generate another demand time for that item.

If the event type is an order arrival then increament the on hand inventory for all items included in the order.

Step 3 : Determine if an order is required.

If the onhand level is less than or equal to the reorder point for that item and there are no outstanding orders for that item, place an order. If the ordering policy is independent, go to step 5.

Step 4 : Test for joint replenishments.

If the onhand level is less than or equal to the can order point for another item j in the replenishment set, place joint replenishment for the items to the level S_i.

Step 5 : Determine if the simulation is complete.

If CT $\geq$ NSIM stop. else go to step 2.

The program collects detailed statistics at each event time for unit years of stock held, number of stockouts, unit years of stockouts, number of orders, number of units ordered, and number of demands for each item.

At the conclusion of the simulation additional summary statistics are computed for achieved service levels and costs.

2. Flowchart

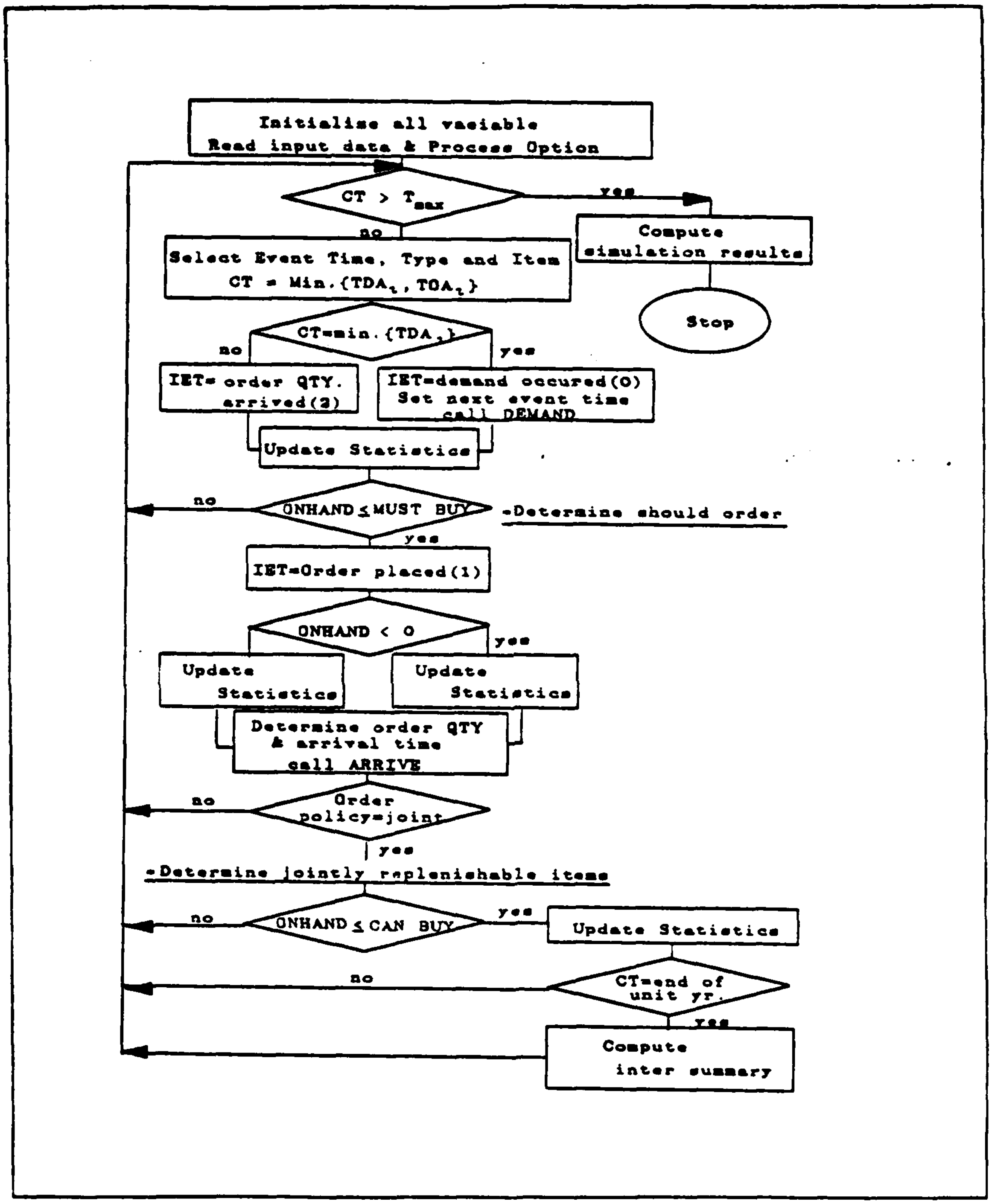


Figure 4.1 Simulation Flowchart.

V. EVALUATION OF THE MODEL

A. PERFORMANCE ANALYSIS

1. Basic Examples Tested

Table 4 shows the data used for the evaluation of the joint replenishment model. Several combinations (96 cases) of service level, service measure, and lead times were considered. The data include the test set considered in Silver's paper [Ref. 3].

TABLE 4

ITEM CHARACTERISTICS

group number	A (\$)	a (\$)	number of items	i	ITEM DATA λ_i (piece/yr.)	v_i (\$/piece)
1	50	10	4	1	290.000	6.90
				2	41.000	1.20
				3	77.000	3.90
				4	122.000	2.30
2	125	80	10	1	40.630	0.45
				2	4.090	36.41
				3	34.680	42.17
				4	4.240	4.46
				5	4.240	4.46
				6	4.090	36.41
				7	4.240	4.46
				8	28.780	3.92
				9	4.090	36.41
				10	4.090	36.41
3	50	5	4	1	290.000	6.90
				2	41.000	1.20
				3	77.000	3.90
				4	122.000	2.30
4	50	5	8	1	290.000	6.90
				2	41.000	1.20
				3	77.000	3.90
				4	122.000	2.30
				5	50.000	1.20
				6	154.000	3.90
				7	87.000	2.30
				8	25.000	1.20

2. Model Results

a. Convergence of Parameter Values and Costs

Table 5 shows the results of each iteration from Silver's model for joint replenishment. Observe that five iterations were required for convergence. This example considers service measure type 2 with a specified level 0.99. The lead time is one month.

TABLE 5
PARAMETERS AND COSTS

Item	Iter.	ρ	C	S	EC
1	1	0.9942	79	146	201.8498
	2	0.9972	84	152	210.4767
	3	0.9973	84	152	210.5555
	4	0.9974	87	152	210.8846
	5	0.9973	87	152	210.8239
2	1	0.9272	31	90	21.7035
	2	0.9499	36	97	23.4369
	3	0.9464	36	96	23.1011
	4	0.9469	36	96	23.1429
	5	0.9467	36	96	23.1271
3	1	0.9650	27	85	66.6963
	2	0.9768	48	89	70.0377
	3	0.9741	43	88	68.8164
	4	0.9744	43	88	68.9584
	5	0.9744	43	88	68.9309
4	1	0.9774	45	138	63.6090
	2	0.9850	71	145	66.8895
	3	0.9835	70	143	65.8678
	4	0.9835	70	143	65.9083
	5	0.9835	70	143	65.8689

b. Cost Savings: Joint Replenishment VS. Independent Control

Table 6 shows the cost savings achieved by the joint replenishment policy for both the modified Silver's method and our heuristic method.

The comparisions are provided for two cases:

1. Service measure 1, $P_1 = 0.95$
2. Service measure 2, $P_2 = 0.99$

Observe that both the heuristic method and Silver's method achieve cost savings greater than 13 % by using joint replenishment, and the heuristic method results in greater cost savings than does Silver's method.

3. Simulation Results

Figure 5.1 shows the time history of stock on hand for items 1 to 4 generated by one simulation. The upper graph represents individual control and the lower graph represents joint replenishment.

TABLE 6
COMPARISION OF TWO METHODS

1. modified Silver's method : $P_1 = 0.95$, $L = 1$ month

item	ρ	s	C	S	EC(JP)	EC(IP)	Save(%)
1	0.9973	32	119	184	221.6339	232.024	4.48
2	0.9467	4	40	100	23.2671	35.343	34.17
3	0.9744	9	52	97	70.9459	88.860	20.16
4	0.9835	13	83	156	67.1722	84.977	20.95
				total cost	383.0190	441.203	13.19

2. heuristic method : $P_1 = 0.95$, $L = 1$ month

item	ρ	s	C	S	EC(JP)	EC(IP)	Save(%)
1	0.9942	31	110	177	211.2798	232.024	8.94
2	0.9272	3	34	93	21.6035	35.343	38.87
3	0.9650	9	36	94	68.7113	88.860	22.67
4	0.9774	14	59	152	65.3723	84.977	23.07
				total cost	351.8467	441.203	16.83

3. modified Silver's method : $P_2 = 0.99$, $L = 1$ month

item	ρ	s	C	S	EC(JP)	EC(IP)	Save(%)
1	0.9973	25	112	177	211.9739	220.984	4.08
2	0.9467	-1	35	95	22.0671	34.383	35.82
3	0.9744	5	48	93	67.8259	85.740	20.89
4	0.9835	7	77	150	64.4122	82.217	21.66
				total cost	366.2788	423.323	13.48

4. heuristic method : $P_2 = 0.99$, $L = 1$ month

item	ρ	s	C	S	EC(JP)	EC(IP)	Save(%)
1	0.9942	25	104	171	202.9998	220.984	8.14
2	0.9272	-1	30	89	20.6435	34.383	39.96
3	0.9650	5	32	90	65.5912	85.740	23.50
4	0.9774	8	53	146	62.6123	82.217	23.85
				total cost	351.8467	423.323	16.88

The simulation results for several combinations of leadtime, service measure and service level are represented in Tables 7 , 8 , 9 .

When lead time is less than 1 month, the heuristic method yields more savings in total cost and also satisfies the required service level. But, as the lead time becomes larger, it sometimes fails to meet the required service level, although it does continue to generate greater cost savings than Silver's method with the cost parameters examined in this thesis. Both methods generate substantial savings over independent control. In short lead times, both methods tend to overprotect against stockouts. They generally produce service levels in excess of what is required.

One explanation for the better performance of the joint replenishment models for the short lead time examples is the fact that a key step in the joint replenishment algorithm for determining the S, c, s parameters assumes that lead time is zero. Adjustments are then made to those values to account for the nonzero lead times but, apparently those adjustments are not adequate. Additional work is required to examine this issue and to explore other methods of adjustment.

We should keep in mind the fact that the model satisfies all requirements in the experiment does not mean that it is a correct model -- merely that it is a plausible one which has not been found inadequate by the data or experiment.

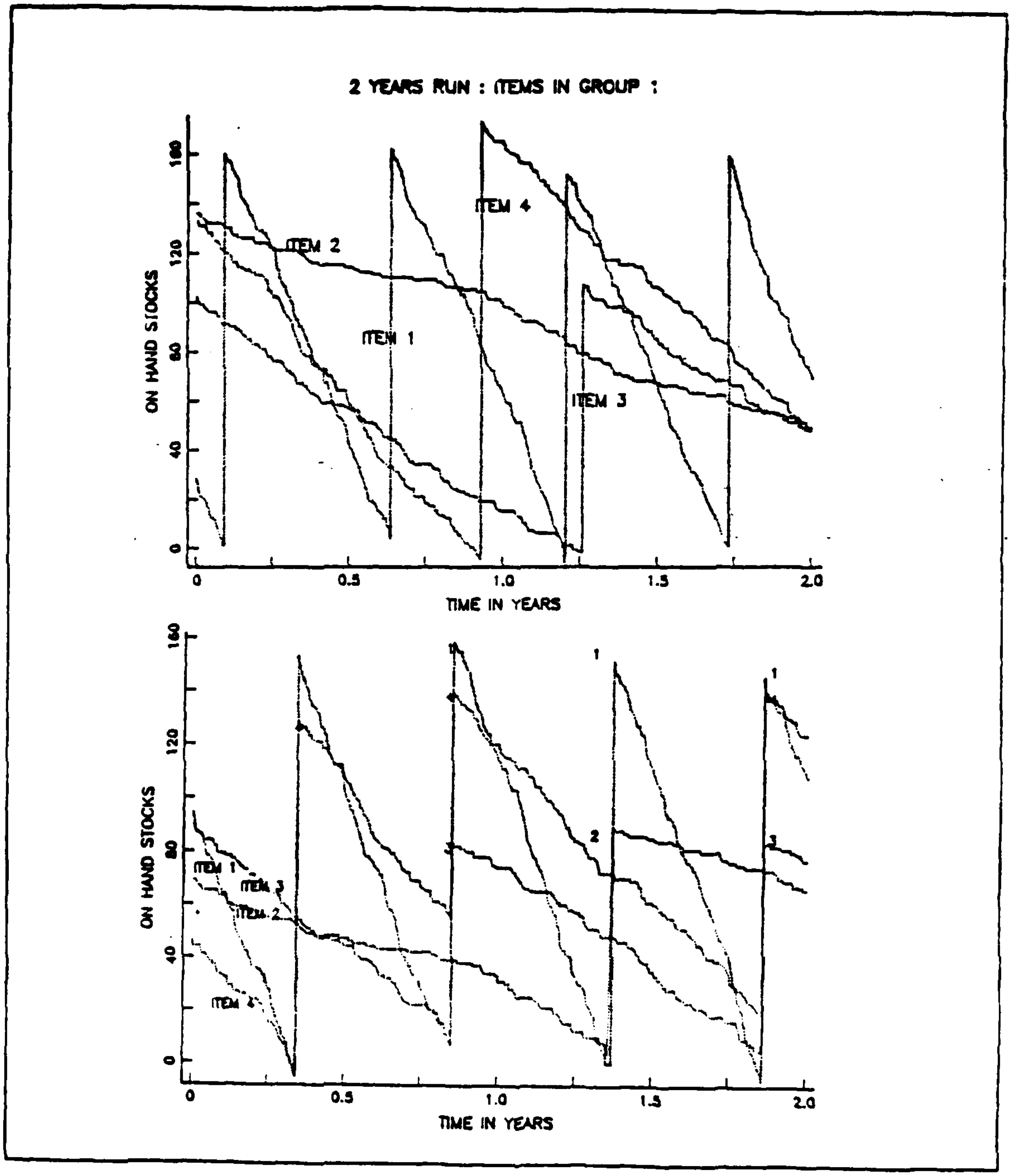


Figure 5.1 Stock on Hand – Independent Control and Joint Replenishment.

TABLE 7

COST SAVINGS BY JOINT REPLENISHMENT: WHEN L = 1 MONTH

Group number	req. SVL (%)	AVG. Total Variable COsts($)				Cost Savings(%) by Joint Replenishment		AVG. SVL (%) achieved by SIM.
		INDEPENDENT		JOINT				
		MODEL	SIM	MODEL	SIM	MODEL	SIM	
1. PARAMETERS FROM MODIFIED SILVER'S METHOD								
1	P1=95 P1=99	441.20 448.06	443.04 445.77	383.02 392.74	361.48 368.70	13.19 12.35	18.41 17.29	99.49 99.74
	P2=95 P2=99	403.20 423.32	414.71 431.00	343.31 366.28	341.12 348.38	14.86 13.48	17.75 19.17	97.91 99.71
2	P1=95 P1=99	1111.43 1153.33	1073.85 1107.24	998.32 1037.70	953.78 956.40	10.18 10.03	11.22 13.62	99.27 100.00
	P2=95 P2=99	1032.91 1088.91	1029.70 1066.90	878.37 971.82	888.82 926.96	14.96 10.75	13.68 13.12	97.70 99.64
3	P1=95 P1=99	423.30 430.16	422.05 425.74	356.89 365.83	345.93 354.15	15.69 14.95	18.04 16.82	99.76 100.00
	P2=95 P2=99	386.08 406.80	396.72 409.84	315.71 340.01	315.35 333.05	18.23 16.42	20.51 18.74	98.44 99.76
4	P1=95 P1=99	676.10 686.94	677.55 683.89	513.14 528.48	490.92 508.80	24.10 23.07	27.54 25.60	99.21 99.80
	P2=95 P2=99	620.56 650.70	627.49 650.80	458.34 490.60	433.05 459.33	26.14 24.60	30.99 29.42	99.01 99.83
2. PARAMETERS FROM HEURISTIC METHOD								
1	P1=95 P1=99	441.20 448.06	443.04 445.77	366.97 376.93	365.30 368.73	16.83 15.88	17.55 17.28	96.51 99.04
	P2=95 P2=99	403.20 423.32	414.71 431.02	329.23 351.85	329.36 347.15	18.35 16.88	20.58 19.46	97.34 99.46
2	P1=95 P1=99	1111.43 1153.33	1073.85 1107.24	949.18 999.68	960.01 979.11	14.60 13.32	10.60 11.57	97.43 99.44
	P2=95 P2=99	1032.91 1088.91	1029.70 1066.90	849.28 903.69	845.07 893.75	17.78 17.01	17.93 16.23	97.54 99.08
3	P1=95 P1=99	423.30 430.16	422.05 425.74	332.99 343.17	342.74 350.13	21.33 20.22	18.79 17.76	98.60 99.77
	P2=95 P2=99	386.08 406.80	396.72 409.84	296.95 319.11	314.70 328.87	23.09 21.56	20.67 19.76	98.01 99.64
4	P1=95 P1=99	676.10 686.94	677.55 683.89	447.50 463.88	455.68 471.45	33.81 32.47	32.75 31.06	97.19 99.51
	P2=95 P2=99	620.56 650.70	627.49 650.80	392.74 426.24	394.17 420.22	36.71 34.50	37.18 35.43	97.54 99.29

TABLE 8

COST SAVINGS BY JOINT REPLENISHMENT: WHEN L = 3 MONTH

Group number	req. SVL (%)	AVG. Total Variable COsts($) INDEPENDENT MODEL	INDEPENDENT SIM	JOINT MODEL	JOINT SIM	Cost Savings(%) by Joint Replenishment MODEL	Cost Savings SIM	AVG. SVL (%) achieved by SIM.
1. PARAMETERS FROM MODIFIED SILVER'S METHOD								
1	P1=95 P1=99	453.70 466.64	451.55 464.38	394.04 408.46	368.34 383.34	13.15 12.47	18.43 17.45	98.97 99.74
	P2=95 P2=99	404.20 431.48	407.47 433.39	340.27 373.06	331.60 350.99	15.82 13.54	18.62 19.01	96.91 99.59
2	P1=95 P1=99	1173.94 1224.36	1135.01 1177.53	1019.79 1078.73	938.40 994.54	13.13 11.89	17.32 15.54	98.83 99.33
	P2=95 P2=99	1047.78 1109.34	1054.40 1097.29	893.13 992.50	855.98 917.02	14.76 10.53	18.82 16.42	98.34 99.78
3	P1=95 P1=99	435.80 448.74	439.10 446.89	365.75 382.33	358.36 371.04	16.07 14.80	18.39 16.97	96.73 99.51
	P2=95 P2=99	386.76 413.82	390.60 410.99	314.71 347.03	309.32 337.40	18.63 16.14	20.81 17.91	97.88 99.57
4	P1=95 P1=99	694.66 713.84	690.92 711.89	525.31 549.89	491.81 516.17	24.38 22.97	28.82 27.49	99.32 99.80
	P2=95 P2=99	619.70 660.82	623.99 660.38	446.77 498.85	418.44 467.03	27.91 24.51	32.94 29.28	98.70 99.84
2. PARAMETERS FROM HEURISTIC METHOD								
1	P1=95 P1=99	453.70 466.64	451.55 464.38	377.98 392.64	363.73 378.46	16.69 15.86	19.45 18.50	97.35 99.76
	P2=95 P2=99	404.20 431.48	407.47 433.39	326.78 357.32	318.15 346.98	19.15 17.19	21.92 19.94	96.75 99.44
2	P1=95 P1=99	1173.94 1224.36	1135.01 1177.53	978.83 1041.41	966.47 1015.42	16.62 14.94	14.85 13.77	*93.78 99.17
	P2=95 P2=99	1047.78 1109.34	1054.40 1097.29	854.13 950.04	860.07 923.59	18.48 14.36	18.43 15.83	97.94 99.44
3	P1=95 P1=99	435.80 448.74	439.10 446.89	340.91 358.43	344.09 360.88	21.77 20.13	21.64 19.25	99.30 99.77
	P2=95 P2=99	386.76 413.82	390.60 410.99	291.57 324.41	301.49 327.00	24.61 21.61	22.82 20.44	97.70 99.74
4	P1=95 P1=99	694.66 713.84	690.92 711.89	456.20 483.56	447.29 474.04	34.33 32.26	35.26 33.41	95.34 98.97
	P2=95 P2=99	619.70 660.82	623.99 660.38	381.62 432.52	400.09 425.76	38.42 34.55	35.88 35.53	95.55 99.10

TABLE 9

COST SAVINGS BY JOINT REPLENISHMENT: WHEN L = 6 MONTH

Group number	req. SVL (%)	AVG. Total Variable COsts($)				Cost Savings(%) by Joint Replenishment		AVG. SVL (%) achieved by SIM.
		INDEPENDENT		JOINT				
		MODEL	SIM	MODEL	SIM	MODEL	SIM	
1. PARAMETERS FROM MODIFIED SILVER'S METHOD								
1	P1=95	465.47	461.92	404.32	374.87	13.14	18.85	97.53
	P1=99	485.50	482.24	425.84	396.24	12.29	17.83	99.24
	P2=95	406.25	407.04	343.22	320.37	15.51	21.29	96.23
	P2=99	441.41	440.34	381.50	354.81	13.57	19.42	99.34
2	P1=95	1217.73	1174.03	1068.46	983.25	12.26	16.25	97.90
	P1=99	1285.89	1240.05	1131.83	1045.79	11.98	15.67	99.33
	P2=95	1046.00	1056.59	897.43	851.39	14.20	19.42	98.07
	P2=99	1161.47	1132.14	1014.36	933.91	12.67	17.51	99.77
3	P1=95	447.56	442.91	375.58	359.91	16.08	18.74	94.80
	P1=99	467.60	464.06	397.10	380.81	15.08	17.94	99.14
	P2=95	390.96	392.25	317.44	310.39	18.81	20.87	97.19
	P2=99	423.50	418.99	355.48	342.46	16.06	18.27	99.29
4	P1=95	712.21	711.50	536.58	505.25	24.66	28.99	96.47
	P1=99	740.67	737.77	571.58	529.95	22.83	28.17	99.04
	P2=95	624.71	629.18	447.04	411.24	28.44	34.64	97.19
	P2=99	674.81	674.72	507.72	465.69	24.76	30.98	99.44
2. PARAMETERS FROM HEURISTIC METHOD								
1	P1=95	465.47	461.92	387.43	379.63	16.77	17.82	*92.64
	P1=99	485.50	482.24	409.79	401.59	15.60	16.72	98.74
	P2=95	406.25	407.04	328.41	328.97	19.16	19.18	95.33
	P2=99	441.41	440.34	366.23	363.02	17.03	17.56	99.08
2	P1=95	1217.73	1174.03	1030.96	991.45	15.34	15.55	98.92
	P1=99	1285.89	1240.05	1090.96	1049.95	15.16	15.33	100.00
	P2=95	1046.00	1056.59	848.43	878.96	18.89	16.81	96.53
	P2=99	1161.47	1132.14	962.32	940.00	17.15	16.97	99.34
3	P1=95	447.56	442.91	351.19	341.63	21.53	22.87	*91.90
	P1=99	467.60	464.06	372.95	362.02	20.24	21.99	*96.14
	P2=95	390.96	392.25	292.41	298.23	25.21	23.97	95.57
	P2=99	423.50	418.99	331.47	330.47	21.73	21.13	98.48
4	P1=95	712.21	711.50	466.69	532.09	34.47	25.21	*86.62
	P1=99	740.67	737.77	503.17	559.74	32.07	24.13	*94.34
	P2=95	624.71	629.18	381.93	404.55	38.36	35.70	*92.36
	P2=99	674.81	674.72	442.41	452.60	34.44	32.92	*97.70

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

For each of several test combinations, we determined the independent and coordinated control parameter values and the associated expected costs. We simulated several years of operations with the derived joint replenishment ordering parameters to evaluate the effectiveness of the models as compared to independent control models. We also use the simulation to compare a simple heuristic joint replenishment model. With the simulation we were able to accumulate costs (ordering, holding and stockout) and effectiveness as measured by two service levels.

In summary, the following statements can be made:

1. Coordinated control results in total costs significantly lower than under independent control. The average cost savings are about 18.7 % .
2. Coordinated control provides better service than independent control. In a few cases, the actual service for an individual item is slightly lower than required: when this occurs, the item involved is the one that triggers the majority of the replenishments in the group.
3. Cost savings of coordinated control increases as the ratio a/A decreas es.
4. Cost savings improve as the number of items in replenishment set increases: the more items there are in the group, the more beneficial coordinated replenishment becomes.
5. For a given service measure and lead time, the cost savings increase as the desired service level decreases.

B. RECOMMENDATIONS

The procedure to determine the parameters for coordinated control requires only the data for independent control. We observe that the joint replenishment models perform best for shorter lead times. This is probably the result of the fact that the joint replenishment algorithms initially assume zero lead times and then adjust the derived parameters heuristically to account for nonzero lead times. This adjustment procedure may be inadequate. Additional work is needed to consider other adjustment methods.

APPENDIX A

"PARA":PROGRAM TO SELECT PARAMETERS

```
C          **    VARIABLE DEFINITION   **
C        ==================================
C        DM       : ANNUAL DEMAND
C        MR       : REORDER POINT
C        S/S1     : ORDER - UP POINT
C        C/C1     : CAN ORDER POINT
C        IPC      : OPTIMAL PAIR OF C
C        IPS      : OPTIMAL PAIR OF S
C        Q        : ECNOMIC ORDER QUANTITY
C        ECI      : AVERAGE ANNUAL TOTAL VARIABLE COST(INDEPENDENT ODER)
C        EC/EC1   : AVERAGE ANNUAL TOTAL VARIABLE COST(JOINT ORDER)
C        TLM      : LEAD TIME(IN MONTH)
C        TLT      : LEAD TIME(IN YEARS)
C        TL       : AVERAGE LEAD TIME(IN YEAR)
C        DLT      : LEAD TIME DEMAND
C        NIT      : NO. OF TOTAL ITEM IN THE GROUP
C        TN       : NO. OF TRIGGERING AN ORDER BY ITEM I IN THE GROUP
C        R / RHO  : OPPORTUNITY TO REPLENISH AT REDUCED COST
C        F        : HOLDING COST RATE TO PROCUREMENT COST
C        UP       : UNIT PRICE
C        OCI      : INDIVIDUAL ORDERING COST
C        OCG      : GROUP  ORDERING COST
C        SVL      : SERVICE LEVEL
C        SCU      : IMPUTED UNIT STOCKOUT COST GIVEN SVL.
C        PSO      : MAX. ALLOWABLE ROBABILITY OF STOCK OUT
C        STON     : EXPECTED NUMBER OF STOCKOUT DURING A ORDER CYCLE
C
C      ------------------------------------------------------------------
       DIMENSION ITM(100),DM(100),UP(100),H(100),OCI(100),ECI(100),Q(100)
      &,C(100),TN(100),S(100),IS(100),R(100),U(100),EC(100),MR(100),
      &IQ(100),DLT(100),SCU(100),IC(100),RHO(50,100),EC1(50,100),
      &CI(50,100),S1(50,100),IC1(50,100),IS1(50,100),IPC(100),IPS(100),
      &MR1(100),TL(100)
C
C      * DATA INITIALIZATION
        STN=0.
        TVC=0.
C
C      * READ PROCESS OPTIONS
C
        CALL OPTION(ID,LT,KEY,LTT,SVL,MSRCH)
          READ(ID,110)NIT,OCG,F
          TLT=LT/12.
          DO 11 I=1,NIT
            READ(ID,111)ITM(I),DM(I),UP(I),OCI(I)
            H(I)=UP(I)*F
       IF(LTT.EQ.1)TL(I)=TLT
       IF(LTT.EQ.2)THEN
        CALL LEAD(TLT,TLY)
        TL(I)=TLY
       END IF
 11    CONTINUE
C
C      * PRINT SELECTED OPTIONS
C
       WRITE(7,100)KEY,SVL,LTT,ID,MSRCH
C
C      *  CALCULATE E.O.Q
C
       DO 1 I =1,NIT
        DLT(I)=DM(I)*TL(I)
```

```
      OC=OCG+OCI(I)
      Q(I)=SQRT(2*DM(I)*OC/H(I))
      IQ(I)=INT(Q(I)+.5)
C
C     * COMPUTE REORDER POINT:IND. ORDER POLICY
C
C   KEY = 1 : PROB.OF NO SHORTAGE PER REPLENISHMENT CYCLE
C   KEY = 2 : FRACTION OF DEMAND TO BE SATISFIED DIRECTLY FROM SHELF
C
      CALL POISON(KEY,DLT(I),Q(I),SVL,MX,M,MAX,CCDF1)
        MR1(I)=M
      IF(KEY.EQ.1)THEN
        SCU(I)=H(I)*Q(I)/((1.- SVL)*DM(I))
      END IF
      IF(KEY.EQ.2)THEN
        SCU(I)=H(I)*Q(I)/(CCDF1*DM(I))
      END IF
      ECI(I)=H(I)*((Q(I)+1.)/2.+(MR1(I)-DLT(I)))+OC*(DM(I)/Q(I))
      WRITE(7,501)ITM(I),DLT(I),TL(I),Q(I),IQ(I),MR1(I),ECI(I),SCU(I)
      TVC=TVC+ECI(I)
      TN(I)=DM(I)/Q(I)
      STN=STN+TN(I)
 1    CONTINUE
C
      WRITE(7,502)TVC
C
C     *  DETERMINE THE PARAMETERS FOR JOINT ORDER POLICY
C            (REORDER / CAN ORDER / ORDER-UP POINT)
C
      DO 2 I=1,NIT
 2    U(I)=STN-TN(I)
      DO 3 I=1,NIT
 3     R(I)=DM(I)/(DM(I)+U(I))
C
C     * SEARCH LOOP FOR DETERMINIG A PAIR (S,C) : ITEM BY ITEM
C
      II=1
      SEC=0.
      SEC1=0.
 41   DO 4  I=1,NIT
        RHO(I,II)=R(I)
        K=1
        CMIN=0.
        CMAX=Q(I)
        C(K)=0.
        S(K)=Q(I)
        EC(K)=ECI(I)
        TEST=10.**10
        ICASE=1
        LOOP=0
 5    K=K+1
      IF(ICASE.EQ.1)THEN
       C(K)=C(K-1)+(CMAX-CMIN)/2
      END IF
      IF(ICASE.EQ.2)THEN
       C(K)=C(K-1)-(CMAX-CMIN)/2
      END IF
      S(K) = C(K)-R(I)*(1-R(I)**C(K))/(1-R(I))+SQRT(2*DM(I)*(OCI(I)
     &              +OCG*R(I)**C(K))/H(I)+2*C(K)*R(I)**(C(K)+1)/(1-R(I))
     &              -R(I)*(1-R(I)**C(K))*(1+R(I)**(C(K)+1))/(1-R(I))**2)
      EC(K)=(1./(S(K)-C(K)+R(I)*(1.-R(I)**C(K))/(1.-R(I))))
     &*((S(K)-C(K))*(S(K)+C(K)+1)*H(I)/2+R(I)*(C(K)-R(I)*(1.-R(I)**C(K))
     &/(1.-R(I)))*H(I)/(1.-R(I))+DM(I)*R(I)**C(K)*OCG+DM(I)*OCI(I))
      IS(K)=INT(S(K)+.5)
      IC(K)=INT(C(K)+.5)
      IF(C(K).EQ.C(K-1).AND.S(K).EQ.S(K-1))GO TO 6
      IF(EC(K).GT.(0.9995*EC(K-1)).AND.EC(K).LE.EC(K-1)) GO TO 6
      IF(EC(K).LT.EC(K-1)) THEN
              IF(C(K).GT.C(K-1))THEN
```

```
                 CMIN=C(K)
                 ICASE=1
                 GO TO 5
              ELSE
                 CMAX=C(K)
                 ICASE=2
                 GO TO 5
              END IF
      ELSE
              IF(C(K).LE.C(K-1))THEN
                 CMIN=C(K)
                 ICASE=1
                 GO TO 5
              ELSE
                 CMAX=C(K)
                 ICASE=2
                 GO TO 5
              END IF
      END IF
 6    IF(LOOP.EQ.0)THEN
        CMIN=0
        CMAX=C(2)
        ICASE=2
        LOOP=1
        GO TO 5
      END IF
C
C    * FIND OUT THE PAIR (S,C), MINIMIZES EC
C
      DO 7 J=1,K
        IF(EC(J).LE.TEST)THEN
         TEST=EC(J)
         EC1(I,II)=EC(J)
         C1(I,II)=C(J)
         S1(I,II)=S(J)
        END IF
 7     CONTINUE
C
        IC1(I,II)=INT(C1(I,II)+.5)
        IS1(I,II)=INT(S1(I,II)+.5)
        SEC1=SEC1+EC1(I,II)
 4    CONTINUE
      SHAVE=(TVC-SEC1)/TVC
C
      WRITE(7,507)II
      DO 20 I=1,NIT
       WRITE(7,508)I,RHO(I,II),IC1(I,II),IS1(I,II),EC1(I,II)
 20   CONTINUE
       WRITE(7,509)SEC1,SHAVE
C
      IF(MSRCH.EQ.2) GO TO 42
C
C    *  COMPUTATIONS NEEDED ONLY FOR S. METHOD 1
C
      IF(II.GE.2)THEN
      DO 21 I=1,NIT
 21   IF(C1(I,II).EQ.C1(I,II-1).AND.S1(I,II).EQ.S1(I,II-1)) GO TO 42
      END IF
      IF(II.GE.2)THEN
      DO 22 I=1,NIT
 22   IF(II.GT.NIT.OR.EC1(I,II).GT.(0.9995*EC1(I,II-1))
     &                         .AND.EC1(I,II).LE.EC1(I,II-1)) GO TO 42
      END IF
C     IF (II.GE.7) GO TO 42
      DO 23 I=1,NIT
         TN(I) = DM(I) * R(I)**C1(I,II) / ( S1(I,II) - C1(I,II) + R(I)
     &                *(1 - R(I)**C1(I,II)) / (1 -R(I)))
 23    CONTINUE
      SUM=0.
```

```
      DO 24 I=1,NIT
24     SUM=SUM+TN(I)
      DO 25 I=1,NIT
25    U(I)=SUM-TN(I)
      DO 26 I=1,NIT
26    R(I)=DM(I)/(DM(I)+U(I))
       II=II+1
       SEC1=0.
       GO TO 41
C
42    WRITE(7,503)
      DO 27 I=1,NIT
       C(I)=C1(I,II)
       S(I)=S1(I,II)
       IPC(I)=INT(C(I)+.5)
       IPS(I)=INT(S(I)+.5)
27     WRITE(7,504)I,IPC(I),IPS(I)
      WRITE(7,200)
C
C     * COMPUTE REORDER POINT:JOINT ORDER POLICY
C
      DO 28 I=1,NIT
       R(I)=RHO(I,II)
      IF(KEY.EQ.1)THEN
         CALL MUST1(R(I),C(I),DLT(I),SVL,MBUY)
            MR(I)=MBUY
            C(I)=C(I)+MR(I)
            S(I)=S(I)+MR(I)
            IC(I)=INT(C(I)+.5)
            IS(I)=INT(S(I)+.5)
            EC(I)=EC1(I,II)+(MR(I)-DLT(I))*H(I)
      END IF
      IF(KEY.EQ.2)THEN
         CALL MUST2(R(I),C(I),S(I),DLT(I),SVL,MBUY)
            MR(I)=MBUY
            C(I)=C(I)+MR(I)
            S(I)=S(I)+MR(I)
            IC(I)=INT(C(I)+.5)
            IS(I)=INT(S(I)+.5)
            EC(I)=EC1(I,II)+(MR(I)-DLT(I))*H(I)
      END IF
       WRITE(7,505)I,R(I),MR(I),IC(I),IS(I),EC(I)
28     SEC=SEC+EC(I)
      SHAVE=(TVC-SEC)/TVC
      WRITE(7,506)SEC,SHAVE
      STOP
C
100   FORMAT(1X,/,'**  SELECTED OPTIONS :',/,1X,'(* KEY =',I2,2X,'* SVL.
     & =',F4.2,2X,'* LEAD TIME TYPE =',I2,2X,'* DATA=',I2,
     &2X,'* S.METHD=',I2,1X,')',//,' * KEY= 1  : SVL. BY PROB. OF NO SHOR
     &TAGE PER REPLENISHMENT CYCLE',/,'          2 : SVL. BY FRACT. OF DEM
     ^ TO BE SATISFIED DIRECTLY FROM SHELF ',/,' * LTT= 1  : CONSTANT
     &LEADTIME         2 : RANDOM (UNIFORM W/  MEAN) ',/,' * DATA=1  : DATA
     &1   2 : DATA2    3 : DATA3    4 : DATA4 ',/,' * SEARCH METHOD =1 :
     & MODIFIED SILVER METHOD   2 : HEURISTIC',
     &///,20X,'***  INDEPENDENT ORDER POLICY   ***'
     &,//,1X,'ITEM',3X,'LT DEMAND(LT.)',2X,'EOQ(N-INT.)',2X,'(INTEGER)',
     &2X,'REORDER',2X,'EXP. TVC($)',2X,'S/O COST')
110   FORMAT(I3,2X,F6.2,2X,F4.2)
111   FORMAT(I3,2X,F7.3,3X,F5.2,2X,F5.2)
200   FORMAT(//,20X,'***  JOINT ORDER POLICY   ***',/,2X,'ITEM',5X,'RHO'
     &,8X,'REORDER',5X,'CANBUY',5X,'ORDERUP',5X,'EXPECTED COST')
501   FORMAT(1X,I3,4X,F7.3,'(',F5.3,')',3X,F8.3,6X,I4,7X,I3,4X,F9.3,3X,
     83)
502   FORMAT(31X,'SUM OF AVG. ANNUAL TVC($)=',F9.3)
503   FORMAT('',//,15X,'**  OPTIMAL PAIR (C,S)  ** ',/,
     &16X,'ITEM',7X,'OPT. C',7X,'OPT. S')
504   FORMAT(16X,I3,9X,I4,9X,I4)
505   FORMAT(2X,I3,4X,F6.4,7X,I4,9X,I4,7X,I4,7X,F10.4)
```

```
506   FORMAT(45X,'TOTAL COST =',F10.4,'(',F6.4,')')
507    FORMAT('',//,3X,'* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT'
     &,/,15X,'(WHEN LEADTIME = 0)',/,15X,'* AT ITERATION :',I2,/,
     &1X,'ITEM',4X,'RHO',4X,'OPT.C',2X,'OPT.S',3X,'EXP.COST')
508    FORMAT(2X,I2,3X,F6.4,4X,I3,4X,I3,4X,F8.4)
509    FORMAT(23X,'TOTAL =',F9.4,'(',F6.4,')')
      END
C
      SUBROUTINE POISON (KEY,DLT,Q,SVL,MX,M,MAX,CCDF1)
C
      PSO=1-SVL
      IF(KEY.EQ.1) THEN
        CALL SERV1(DLT,PSO,MAX,M)
      RETURN
      END IF
      IF(KEY.EQ.2) THEN
       CALL SERV2(Q,DLT,PSO,MAX,M,CCDF1)
      RETURN
      END IF
      END
C
          SUBROUTINE MAXD(DLT,MAX)
C
      DIMENSION PMF(750)
      I=1
      PMF(1)=EXP(-DLT)
      CDF=PMF(1)
      GO TO 2
 1    PMF(I)=(DLT/(I-1))*PMF(I-1)
      CDF=CDF+PMF(I)
 2    IF(CDF.GE.0.9995) GO TO 3
      I=I+1
      GO TO 1
 3    MAX = I+1
      RETURN
      END
C
          SUBROUTINE SERV1(DLT,PSO,MAX,M)
C
      DIMENSION PMF(200),CDF(200),CCDF(200)
      PMF(1)=EXP(-DLT)
      CDF(1)=PMF(1)
      CCDF(1)=1-CDF(1)
      CALL MAXD(DLT,MAX)
      DO 1 I=2,MAX+1
        PMF(I)=(DLT/(I-1))*PMF(I-1)
        CDF(I)=CDF(I-1)+PMF(I)
 1      CCDF(I)=1-CDF(I)
      DO 2 I=1,MAX+1
 2      IF(PSO.LT.CCDF(I).AND.PSO.GT.CCDF(I+1))GO TO 3
 3    M=I
      RETURN
      END
C
          SUBROUTINE SERV2(Q,DLT,PSO,MAX,M,CCDF1)
C
      DIMENSION PMF(200),CDF(200),CCDF(200)
      STON=Q*PSO
      PMF(1)=EXP(-DLT)
      CDF(1)=PMF(1)
      CCDF(1)=1.-CDF(1)
      CALL MAXD(DLT,MAX)
      DO 1 I=2,MAX+1
        PMF(I)=(DLT/(I-1))*PMF(I-1)
        CDF(I)=CDF(I-1)+PMF(I)
        CCDF(I)=1-CDF(I)
 1    CONTINUE
      I=0
 2    IF(I.EQ.0)THEN
```

```
      PST=DLT
    ELSE
      PST=DLT-I-DLT*CDF(I)+I*CDF(I+1)
    END IF
    IF(PST.LT.STON) THEN
      M=I
      CCDF1=CCDF(M+1)
    ELSE
     I=I+1
     GO TO 2
    END IF
    RETURN
    END
C
      SUBROUTINE LEAD(TLT,TLY)
C
C
      CALL LRND(719325,U,1,1,0)
      TLY=2*TLT*U
      RETURN
      END
C
      SUBROUTINE MUST1(R,C,DLT,SVL,MBUY)
C
      DIMENSION PMF(400),CDF(400)
      IC=INT(C+.5)
      RC=(1./R)**IC
      RHS=SVL/(R**IC)
      PMF(1)=EXP(-DLT)
      IF(PMF(1).LT.1.0E-70) PMF(1)=0.
      CDF(1)=PMF(1)
      DO 1 I=2,400
        PMF(I)=(DLT/(I-1))*PMF(I-1)
        IF(PMF(I).LT.1.0E-70) PMF(I)=0.
        CDF(I)=CDF(I-1)+PMF(I)
 1    CONTINUE
      JS=-2
 3    SUM=0.
      KF=JS+2
      KL=JS+IC
      DO 2 IX=KF,KL
 2    SUM=SUM+PMF(IX+1)*((1/R)**IX)
      TEST=RC*CDF(KL+1)-PMF(JS+2)-(R**(JS+1))*SUM
      IF(TEST.GE.RHS) THEN
         MBUY=JS
      ELSE
        JS=JS+1
        GO TO 3
      END IF
      RETURN
      END
C
      SUBROUTINE MUST2(R,C,S,DLT,SVL,MBUY)
C
      DIMENSION PMF(400),CDF(400)
      IC=INT(C+.5)
      RHS=(S-C+R*(1-R**C)/(1-R))*(1-SVL)
      PMF(1)=EXP(-DLT)
      IF(PMF(1).LT.1.0E-70) PMF(1)=0.
      CDF(1)=PMF(1)
      DO 1 I=2,400
        PMF(I)=(DLT/(I-1))*PMF(I-1)
        IF(PMF(I).LT.1.0E-70) PMF(I)=0.
        CDF(I)=CDF(I-1)+PMF(I)
 1    CONTINUE
      JS=-1
 3    SUM=0.
      DO 2 IW=JS+1,JS+IC
         IF(IW.LT.1) THEN
```

```
             VCDF=0.
             IF( IW.EQ.0)VCDF1=CDF(1)
           ELSE
             VCDF=CDF(IW)
             VCDF1=CDF(IW+1)
             END IF
 2    SUM = SUM+(1/R**IW)*(DLT-IW-DLT*VCDF+IW*VCDF1)
      IF(JS.LT.1) THEN
         VCDF=0.
         IF( JS.EQ.0)VCDF1=CDF(1)
      ELSE
         VCDF=CDF(JS)
         VCDF1=CDF(JS+1)
      END IF
      TEST=R**C*(DLT-JS-DLT*VCDF+JS*VCDF1)+(1-R)*R**(C+JS)*SUM
      IF(TEST.LE.RHS) THEN
         MBUY=JS
      ELSE
        JS=JS+1
        GO TO 3
      END IF
      RETURN
      END
C
       SUBROUTINE OPTION (ID,LT,KEY,LTT,SVL,MSRCH)
C
        WRITE(6,50)
        READ(5,51) ID
        WRITE(6,53)
        READ(5,51) KEY
        WRITE(6,55)
        READ(5,57) SVL
        WRITE(6,52)
        READ(5,51) LT
        WRITE(6,54)
        READ(5,51) LTT
        WRITE(6,56)
        READ(5,51) MSRCH
      RETURN
C
 50   FORMAT('1',' * ENTER OPTION FOR DATA FILE ( 1- 4) ! ',/,'     1 =
     &DATA1  2 = DATA2   3 =DATA3   4 = DATA4')
 51   FORMAT(I1)
 52   FORMAT('1',' * ENTER THE LEADTIME IN MONTH ! ',/,5X, '1 =  1 MONTH
     &',/,5X,'3 =  3 MONTH',/,5X,'6 =  6 MONTH')
 53   FORMAT('1',' * ENTER OPTION FOR THE METHOD OF SVL. MEASURE ! ',/,'
     &      1 =  PROB. OF NO SHORTAGE PER REPLENISHMENT CYCLE ',/,4X,' 2
     &=  FRACTION OF DEMAND TO BE SATISFIED DIRECTLY FROM SHELF ')
 54   FORMAT('1',' * ENTER THE LEADTIME TYPE     ! ',/,5X, '1 = CONSTANT
     & ',/,5X,'2 = RANDOM (UNIFORM W/ MEAN)')
 55   FORMAT('1',' * ENTER THE DESIRED SERVICE LEVEL   ! ',/,4X,'(SHOULD
     &BE IN THE FORM OF "F4.2" : EX. ; 0.90 OR 0.99 ...ETC.)')
 56   FORMAT('1',' * ENTER OPTION FOR SEARCH METHOD (1-2)!',/,'     1 =
     &MODIFIED SILVER METHOD    2 = HEURISTIC')
 57   FORMAT(F4.2)
      END
```

----

1. INPUT FORMAT

```
004  050.00  0.20
  1  290.000     6.90   10.00
  2   41.000     1.20   10.00
  3   77.000     3.90   10.00
  4  122.000     2.30   10.00
```

---

2. EXEC PROGRAM

```
&TRACE OFF
&FN = PARA
&FNO = &CONCAT OF &FN OUTPUT
&TYPE Do you need to compile your program ? (Y)
&READ VAR &R_COMPILE
&IF &R_COMPILE NE Y &GOTO -RUN
-H FORTVS &FN
&IF &RC EQ 0 &GOTO -RUN
&TYPE Your program did not compile; check for errors.
&TYPE Do you wish to view the program LISTING file? (Y)
&READ VAR &RSP1
&IF &RSP1 EQ Y BROWSE &FN LISTING A
&TYPE Do you wish to XEDIT the program file? (Y)
&READ VAR &RESP1
&IF &RESP1 NE Y &EXIT 1
&COMMAND XEDIT &FN FORTRAN A
&TYPE Do you wish to run the program again? (Y)
&READ VAR &RESP2
&IF &RESP2 EQ Y &GOTO -H
&EXIT 1
-RUN
FILEDEF 01 DISK &FN DATA1 A1
FILEDEF 02 DISK &FN DATA2 A1
FILEDEF 03 DISK &FN DATA3 A1
FILEDEF 04 DISK &FN DATA4 A1
FILEDEF 07 DISK &FN OUTPUT A1 (LRECL 133
LOAD &FN (START
&IF &RC EQ 0 &SKIP 9
&TYPE Your program did not run correctly; check for errors.
&TYPE Do you wish to XEDIT the program file? (Y)
&READ VAR &RESP3
&IF &RESP3 NE Y &EXIT 2
&COMMAND XEDIT &FN FORTRAN A
&TYPE Do you wish to run the program again? (Y)
&READ VAR &RESP4
&IF &RESP4 EQ Y &GOTO -H
&EXIT 2
&TYPE YOUR OUTPUT IS IN THE FILE &FN OUTPUT A
&TYPE Do you wish  to BROWSE your output? (Y)
&READ VAR &RESP
&IF &RESP EQ Y &COMMAND BROWSE &FN OUTPUT A
&TYPE Print your output file?  (Y)
&READ VAR &RESP7
&IF &RESP7 EQ Y &COMMAND PRINT &FN OUTPUT A
-REDO
&TYPE Do you wish to XEDIT the program file? (Y/N)
&READ VAR &RESP5
&IF &RESP5 EQ Y XEDIT &FN FORTRAN A
&TYPE Do you wish to run the program again? (Y)
&READ VAR &RESP6
&RESP56 = &CONCAT OF &RESP5 &RESP6
&IF &RESP56 EQ YY &GOTO -H
&IF &RESP6 EQ Y &GOTO -RUN
&EXIT
```

3. RUN EXAMPLE

```
para
Do you need to compile your program ? (Y)
y
VS FORTRAN COMPILER ENTERED.  11:58:02
**MAIN** END OF COMPILATION 1 ******
**POISON** END OF COMPILATION 2 ******
**MAXD** END OF COMPILATION 3 ******
**SERV1** END OF COMPILATION 4 ******
**SERV2** END OF COMPILATION 5 ******
**LEAD** END OF COMPILATION 6 ******
**MUST1** END OF COMPILATION 7 ******
**MUST2** END OF COMPILATION 8 ******
**OPTION** END OF COMPILATION 9 ******
VS FORTRAN COMPILER EXITED.   11:58:10

EXECUTION BEGINS...
 * ENTER OPTION FOR DATA FILE ( 1- 4) !
   1 =  DATA1 2 = DATA2  3 =DATA3  4 = DATA4
1
 * ENTER THE LEADTIME IN MONTH !
    1 =  1 MONTH
    3 =  3 MONTH
    6 =  6 MONTH
1
 * ENTER OPTION FOR THE METHOD OF SVL. MEASURE !
    1 =  PROB. OF NO SHORTAGE PER REPLENISHMENT CYCLE
    2 =  FRACTION OF DEMAND TO BE SATISFIED DIRECTLY FROM SHELF
2
 * ENTER THE LEADTIME TYPE     !
    1 = CONSTANT
    2 = RANDOM (UNIFORM W/ MEAN)
1
 * ENTER THE DESIRED SERVICE LEVEL  !
   (SHOULD BE IN THE FORM OF "F4.2" : EX. ; 0.90 OR 0.99 ...ETC.)
0.99
 * ENTER OPTION FOR SEARCH METHOD (1-2)!
   1 =  MODIFIED SILVER METHOD   2 = HEURISTIC
1
YOUR OUTPUT IS IN THE FILE PARA OUTPUT A
Do you wish to BROWSE your output? (Y)
y
Print your output file? (Y)
n
Do you wish to XEDIT the program file? (Y/N)
n
Do you wish to run the program again? (Y)
n
R; T=1.47/2.39 11:58:39
```

4. OUTPUT 1 (P2 = 0.99)

** SELECTED OPTIONS :
(* KEY = 2 * SVL. =0.99 * LEAD TIME TYPE = 1 * DATA= 1 * S.METHD= 1)

* KEY= 1 : SVL. BY PROB. OF NO SHORTAGE PER REPLENISHMENT CYCLE
  2 : SVL. BY FRACT. OF DEMAND TO BE SATISFIED DIRECTLY FROM SHELF
* LTT= 1 : CONSTANT LEADTIME 2 : RANDOM (UNIFORM W/ MEAN)
* DATA=1 : DATA1 2 : DATA2 3 : DATA3 4 : DATA4
* SEARCH METHOD =1 : MODIFIED SILVER METHOD 2 : HEURISTIC

*** INDEPENDENT ORDER POLICY ***

ITEM	LT DEMAND(LT.)	EOQ(N-INT.)	(INTEGER)	REORDER	EXP. TVC($)	S/O COST
1	24.167(0.083)	158.800	159	25	220.984	1.983
2	3.417(0.083)	143.178	143	3	34.383	1.882
3	6.417(0.083)	108.840	109	7	85.740	3.499
4	10.167(0.083)	178.399	178	10	82.217	1.536
			SUM OF AVG. ANNUAL TVC($)=		423.323	

* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT
(WHEN LEADTIME = 0)
* AT ITERATION : 1

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9942	79	146	201.8498
2	0.9272	31	90	21.7035
3	0.9650	27	85	66.6963
4	0.9774	45	138	63.6090
			TOTAL =	353.8584(0.1641)

* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT
(WHEN LEADTIME = 0)
* AT ITERATION : 2

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9972	84	152	210.4767
2	0.9499	36	97	23.4369
3	0.9768	48	89	70.0377
4	0.9850	71	145	66.8895
			TOTAL =	370.8406(0.1240)

* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT
(WHEN LEADTIME = 0)
* AT ITERATION : 3

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9973	84	152	210.5555
2	0.9464	36	96	23.1011
3	0.9741	43	88	68.8164
4	0.9835	70	143	65.8678
			TOTAL =	368.3406(0.1299)

* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT
(WHEN LEADTIME = 0)
* AT ITERATION : 4

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9974	87	152	210.8846
2	0.9469	36	96	23.1429
3	0.9744	43	88	68.9584
4	0.9835	70	143	65.9083
			TOTAL =	368.8940(0.1286)

*** PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT**
(WHEN LEADTIME = 0)
*** AT ITERATION : 5**

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9973	87	152	210.8239
2	0.9467	36	96	23.1271
3	0.9744	43	88	68.9309
4	0.9835	70	143	65.8689

TOTAL = 368.7505(0.1289)

** OPTIMAL PAIR (C,S) **

ITEM	OPT. C	OPT. S
1	87	152
2	36	96
3	43	88
4	70	143

*** JOINT ORDER POLICY ***

ITEM	RHO	REORDER	CANBUY	ORDERUP	EXPECTED COST
1	0.9973	25	112	177	211.9739
2	0.9467	-1	35	95	22.0671
3	0.9744	5	48	93	67.8259
4	0.9835	7	77	150	64.4122

TOTAL COST = 366.2788(0.1348)

5. OUTPUT 2 (P1 = 0.95)

** SELECTED OPTIONS :
(* KEY = 1 * SVL. =0.95 * LEAD TIME TYPE = 1 * DATA= 1 * S.METHD= 1)

```
* KEY= 1 : SVL. BY PROB. OF NO SHORTAGE PER REPLENISHMENT CYCLE
       2 : SVL. BY FRACT. OF DEMAND TO BE SATISFIED DIRECTLY FROM SHELF
* LTT= 1 : CONSTANT LEADTIME        2 : RANDOM (UNIFORM W/  MEAN)
* DATA=1 : DATA1    2 : DATA2    3 : DATA3    4 : DATA4
* SEARCH METHOD =1 : MODIFIED SILVER METHOD    2 : HEURISTIC
```

*** INDEPENDENT ORDER POLICY ***

ITEM	LT DEMAND(LT.)	EOQ(N-INT.)	(INTEGER)	REORDER	EXP. TVC($)	S/O COST
1	24.167(0.083)	158.800	159	33	232.024	15.113
2	3.417(0.083)	143.178	143	7	35.343	16.762
3	6.417(0.083)	108.840	109	11	88.860	22.051
4	10.167(0.083)	178.399	178	16	84.977	13.453

SUM OF AVG. ANNUAL TVC($)= 441.203

* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT
(WHEN LEADTIME = 0)
* AT ITERATION : 1

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9942	79	146	201.8498
2	0.9272	31	90	21.7035
3	0.9650	27	85	66.6963
4	0.9774	45	138	63.6090

TOTAL = 353.8584(0.1980)

* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT
(WHEN LEADTIME = 0)
* AT ITERATION : 2

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9972	84	152	210.4767
2	0.9499	36	97	23.4369
3	0.9768	48	89	70.0377
4	0.9850	71	145	66.8895

TOTAL = 370.8406(0.1595)

* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT
(WHEN LEADTIME = 0)
* AT ITERATION : 3

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9973	84	152	210.5555
2	0.9464	36	96	23.1011
3	0.9741	43	88	68.8164
4	0.9835	70	143	65.8678

TOTAL = 368.3406(0.1651)

* PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT
(WHEN LEADTIME = 0)
* AT ITERATION : 4

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9974	87	152	210.8846
2	0.9469	36	96	23.1429
3	0.9744	43	88	68.9584
4	0.9835	70	143	65.9083

TOTAL = 368.8940(0.1639)

*** PARAMETERS & EX.COSTS FOR JOINT REPLENISHMENT**
(WHEN LEADTIME = 0)
*** AT ITERATION : 5**

ITEM	RHO	OPT.C	OPT.S	EXP.COST
1	0.9973	87	152	210.8239
2	0.9467	36	96	23.1271
3	0.9744	43	88	68.9309
4	0.9835	70	143	65.8689
			TOTAL =	368.7505(0.1642)

** OPTIMAL PAIR (C,S) **

ITEM	OPT. C	OPT. S
1	87	152
2	36	96
3	43	88
4	70	143

*** JOINT ORDER POLICY ***

ITEM	RHO	REORDER	CANBUY	ORDERUP	EXPECTED COST
1	0.9973	32	119	184	221.6339
2	0.9467	4	40	100	23.2671
3	0.9744	9	52	97	70.9459
4	0.9835	13	83	156	67.1722
				TOTAL COST =	383.0190(0.1319)

APPENDIX B

"SIM":PROGRAM TO TEST THE MODEL

```
C
C         VARIABLE DEFINITIONS
C        =====================
C
C
C     KEY  = OPTIONS FOR ORDER POLICY
C     ISTA = OPTIONS FOR START INVENTORY LEVEL
C     IPRINT= OPTIONS FOR PRINTOUT
C     NIT  = NO.OF ITEMS IN THE GROUP TO BE EVALUATED
C     NSIM = SIMULATION PERIOD
C     SIMT = SIMULATION INTERVAL TIME (1YEAR)
C-------------------------------------------------------------------
C     DM   = ANNUAL DEMAND RATE
C     ME   = ORDER-UP POINT
C     IEQ  = ECONOMIC ORDER QUANTITY(INTEGER)
C     MC   = CAN ORDER POINT (CAN BUY)
C     MR   = REORDER POINT(M1=IND.:M2=JOINT)
C     TDA  = TIME OF DEMAND OCCURING TIME
C     TOA  = ORDER QTY. ARRIVAL TIME
C     M    = INDEX OF EVENT FOR DEMAND OCCURED ITEM
C     IA   = INDEX OF EVENT FOR OREDR QTY. ARRIVED ITEM
C     IET  = INDEX OF EVENT TYPE
C     CT   = MASTER CLOCK TIME IN YRS(THE EARLIEST EVENT OCCURING TIME)
C     DA   = TIME INTERVAL BETWEEN 2 SUCCESSIVE DEMANDS
C     ARV  = UNIFORM RANDOM VARIABLE FOR LEADTIME WITH MEAN  "ULT"
C     ULT  = EXPECTED LEADTIME
C     IQ   = ORDER QTY.
C     IT   = INDEX FOR REPLENISH TRIGGER ITEM
C     JT   = INDEX FOR JOINT REPLENISHED ITEM
C     IOH  = ONHAND INVENTORY
C     IOHS = START LEVEL
C     UOH  = UNIT YEARS ONHAND QTY
C     UBO  = UNIT YEARS BACKORDER QTY
C     NBQ  = NO. OF BACKORDERS (QTY.)
C     NTBQ = TOTAL NUMBER OF BACKORDERS
C     BOT  = BACKORDER TIME
C     TBOT = TOTAL BACKORDER TIME
C     NOD  = TOATL NUMBER OF ORDERS FOR EACH ITEM
C     NDM  = TOTAL NUMBER OF DEMANDS FOR EACH ITEM
C     NTDM = TOTAL NUMBER OF DEMANDS DURING TMAX
C     NJTO = NUMBER OF JOINT REPLENISHMENTS
C-------------------------------------------------------------------
C     UP   = UNIT PRICE
C     F    = HOLDING COST RATE
C     HC   = HOLDING COST
C     OC   = ORDER & REVIEW  COST
C     OCI  = INDIVIDUAL ORDERING COST
C     OCG  = JOINT (GROUP) ORDERING COST
C     SCU  = BACKORDER COST PER BACKORDER UNIT
C     SCT  = BACKORDER COST (TIME DEPENDENT FACTOR)
C     TVC  = AVG. ANNUAL TOTAL VARIABLE COSTS
C
C---------------------------------------------------------------------
      DIMENSION ISEED(10),DM(10),MR(10),MC(10),ME(10),IOH(10),ASVL(10),
     &TDA(100),IT(10,100),JT(10,100),IQ(100),DA(100),TOA(100),IEQ(10),
     &H(10),OCI(10),UP(10),SCU(10),SCT(10),F(10),NJTO(100),OCT(100),
     &UOH(10,100),AUOH(100),UBO(10,100),AUBO(100),NOD(10,100),KB(10,100)
     &,ANOD(100),NDM(10,100),ANDM(100),NBQ(10,100),NTBQ(10,100),KBT(10),
     &BOT(10,100),TBOT(10,100),FRC(10,100),FRQ(10,100),SVL(10,100),
     &AFRQ(100),OC(10,100),AOC(100),COST(10,100),TCOST(100),NTOD(100),
     &HCOST(10,100),SCOST(10,100),SCTU(10,100),SCTT(10,100),NTDM(100),
     &AHCOST(100),ASCTU(100),ASCTT(100),ABOT(10),ABOQ(100),ACOST(100),
```

```
     &THCOST(100),TSCTU(100),TSCTT(100),CCTT(100),M1(10),M2(10),
     &TUOH(100),TUBO(100),TFRT(10),TFRQ(100),NTDMS(100),NTODS(100),
     &ANB(10)
      COMMON ICASE,ISEED,DM,DA,ARV,ULT,I,ISTP,MR,ME,KEY,IEQ
C
C -----   DATA INITIALIZATION ------
C
           SIMT   = 1.
           TMAX   = 0.
           IC     = 0
           ISD    = 293715
           CT     = 0.
           DT     = 0.
           AT     = 0.
           K      = 1
           N      = 0
           TVC    = 0.
           NTJTO  = 0
           TANDM  = 0.
           TANOD  = 0.
           TAOC   = 0.
           TCST   = 0.
           STV    = 0.
           ANJ    = 0.
C
C ----- READ PROCESS OPTIONS -----
C
      CALL OPTION(ID,RATE1,RATE2,KEY,LTOPT,NSIM,ISTA,IP)
C
C ----- READ INPUT DATA  ----
C
       READ(ID,101)NIT,OCG,ISV,MSV,LT
         ULT=LT/12.
      DO 401 I=1,NIT
      READ(ID,100)DM(I),M1(I),M2(I),MC(I),ME(I),IEQ(I),OCI(I),UP(I),F(I)
        IF(KEY.EQ.1)MR(I)=M1(I)
        IF(KEY.EQ.2)MR(I)=M2(I)
        SCU(I)=RATE1*UP(I)
        SCT(I)=RATE2*UP(I)
        H(I)=UP(I)*F(I)
 401  CONTINUE
C
      WRITE(7,198)ID,KEY,NSIM,ISTA,IP,LTOPT,MSV,ISV,ULT,LT
C
      DO  1 I=1,NIT
         ISEED(I)=ISD+I**3
         KBT(I)  = 0
         ANB(I)  = 0.
         TFRT(I) = 0.
         ABOT(I) = 0.
         ASVL(I) = 0.
         TOA(I)  = 10.**10
 1       TDA(I)  = 0.
      DO  2 I=1,NIT
       DO  2 J=1,100
           DA(J)        = 0.
           IQ(J)        = 0
           SVL(I,J)     = 0.
           UOH(I,J)     = 0.
           TUOH(J)      = 0.
           AUOH(J)      = 0.
           COST(I,J)    = 0.
           TCOST(J)     = 0.
           ACOST(J)     = 0.
           UBO(I,J)     = 0.
           TUBO(J)      = 0.
           AUBO(J)      = 0.
           NOD(I,J)     = 0
           NTOD(J)      = 0
```

```
         NTODS(J)      = 0
         ANCD(J)       = 0.
         NDM(I,J)      = 0
         NTDM(J)       = 0
         NTDMS(J)      = 0
         ANDM(J)       = 0.
         NBQ(I,J)      = 0
         NTBQ(I,J)     = 0
         BOT(I,J)      = 0.
         TBOT(I,J)     = 0.
         ABOQ(J)       = 0.
         FRQ(I,J)      = 0.
         FRC(I,J)      = 0.
         TFRQ(J)       = 0.
         AFRQ(J)       = 0.
         OC(I,J)       = 0.
         OCT(J)        = 0.
         OCTT(J)       = 0.
         AOC(J)        = 0.
         NJTO(J)       = 0
         HCOST(I,J)    = 0.
         THCOST(J)     = 0.
         AHCOST(J)     = 0.
         SCOST(I,J)    = 0.
         SCTU(I,J)     = 0.
         TSCTU(J)      = 0.
         ASCTU(J)      = 0.
         SCTT(I,J)     = 0.
         TSCTT(J)      = 0.
         ASCTT(J)      = 0.
         KB   (I,J)    = 0
         IT   (I,J)    = 0
 2       JT   (I,J)    = 0
      DO 3  I=1,NIT
         IF(ISTA.EQ.1.AND.KEY.EQ.1)THEN
          IOH(I) = IEQ(I)
         END IF
         IF(ISTA.EQ.1.AND.KEY.EQ.2) THEN
          IOH(I)  = ME(I)
         END IF
         IF(ISTA.EQ.2) THEN
          CALL START
          IOH(I)=ISTP
         END IF
 3       CONTINUE
C
C   ----  PRINTOUT INPUT DATA  ----
C
      DO  4 I=1,NIT
 4    WRITE(7,199) I,DM(I),IEQ(I),MR(I),MC(I),ME(I),IOH(I),ULT
      WRITE(7,97)
      DO 5  I=1,NIT
      WRITE(7,98) I,UP(I),F(I),H(I),OCG,OCI(I),SCU(I),SCT(I)
           IC = IC + 1
           IF (IC.GE. 65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
             IC = 0
             WRITE(7,200)
           END IF
 5    CONTINUE .
C
C  ----  START SIMULATION ----
C
C
      IF (IP.EQ.1.OR.IP.EQ.2)THEN
      WRITE(7,200)
      END IF
C
      DO 777 ITER=1,NSIM
      IF (ITER.GT.1.AND.IP.EQ.1.OR.ITER.GT.1.AND.IP.EQ.2) THEN
```

```
      IC = 0
      WRITE(7,200)
      END IF
      TMAX=TMAX+SIMT
C
 20   IF(CT.GT.TMAX) GO TO 77
          N = N+1
          JOINT = 0
C
C ----  DETERMINE THE INITIAL DEMAND OCCURANCE TIMES FOR EACH ITEM ----
C
       IF(N.LE.1) THEN
         DO  6  I=1,NIT
            ICASE=I
            CALL DEMAND
            TDA(I)=DA(ICASE)
 6          UOH(I,ITER)=UOH(I,ITER)+IOH(I)*DA(ICASE)
C
C ----   DETERMINE THE EARLIEST EVENT  -----
C    (DEMMAND OCCURED / ORDER PLACED / ORDERED QTY. ARRIVED)
C
            IF(TDA(1).LE.TDA(2)) THEN
              TD=TDA(1)
            ELSE
              TD=TDA(2)
            END IF
            DO 7 I=1,NIT
            IF(TDA(I).LE.TD) THEN
              TD=TDA(I)
              M=I
            END IF
 7          CONTINUE
C
            IET=0
            CT=TDA(M)
            IOH(M)=IOH(M)-1
            IF(MSV.EQ.1.AND.IOH(M).EQ.-1) KB(M,ITER)=KB(M,ITER)+1
            NDM(M,ITER)=NDM(M,ITER)+1
C
C  ---- SET NEXT DEMAND OCCURING TIME FOR THE ITEM :'M'  ----
C
            ICASE=M
            CALL DEMAND
            TDA(M)=CT+DA(ICASE)
            GO TO 90
       END IF
C
      IF(TDA(1).LE.TDA(2)) THEN
          TD=TDA(1)
      ELSE
          TD=TDA(2)
      END IF
      DO 8 I=1,NIT
        IF(TDA(I).LE.TD) THEN
           TD=TDA(I)
           M=I
        END IF
 8    CONTINUE
C
      IF(TOA(1).LE.TOA(2)) THEN
          TA=TOA(1)
      ELSE
          TA=TOA(2)
      END IF
      DO 9 I=1,NIT
        IF(TOA(I).LE.TA) THEN
           TA=TOA(I)
           IA=I
        END IF
```

```
 9    CONTINUE
C
      IF(TA.LE.TD)THEN
        GO TO 93
      ELSE
        IET=0
        CT=TD
        ICASE=M
        CALL DEMAND
        TDA(M)=CT+DA(ICASE)
        IOH(M)=IOH(M)-1
        IF(MSV.EQ.1.AND.IOH(M).EQ.-1) KB(M,ITER)=KB(M,ITER)+1
        NDM(M,ITER)=NDM(M,ITER)+1
      END IF
C
C ----  DETERMINE SHOULD ORDER  ----
C
 90   IF(IOH(M).GT.MR(M))THEN
         UOH(M,ITER)=UOH(M,ITER)+IOH(M)*DA(M)
         IF(IP.EQ.1)THEN
         WRITE(7,501)CT,IET,M,UOH(M,ITER),(IOH(I),I=1,4)
         END IF
         IF(IP.EQ.2)THEN
         WRITE(7,601)CT,IET,M,UOH(M,ITER),IOH(M)
         END IF
            IC = IC + 1
           IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
              IC = 0
              WRITE(7,200)
            END IF
         GO TO 20
      END IF
C
      IF(K.EQ.1)THEN
        IET=1
        K=K+1
        IT(M,K)=M
         IF(IOH(M).LT.0) THEN
           NBQ(M,ITER)=-IOH(M)
           IF(KEY.EQ.1) IQ(M) = IEQ(M)+NBQ(M,ITER)
           IF(KEY.EQ.2) IQ(M) = ME(M)+NBQ(M,ITER)
           IF(LTOPT.EQ.1)TOA(M) = CT + ULT
           IF(LTOPT.EQ.2)THEN
            CALL ARRIVE
            TOA(M)=CT+ARV
           END IF
           BOT(M,ITER)=TOA(M) - CT
           IF(NBQ(M,ITER).EQ.1)TBOT(M,ITER)=TBOT(M,ITER)+BOT(M,ITER)
           UBO(M,ITER)=UBO(M,ITER)+BOT(M,ITER)
         IF(IP.EQ.1)THEN
           WRITE(7,502)CT,IET,M,NBQ(M,ITER),BOT(M,ITER),UBO(M,ITER),
     &                 IQ(M),TOA(M),(IOH(I),I=1,4)
         END IF
         IF(IP.EQ.2)THEN
           WRITE(7,602)CT,IET,M,NBQ(M,ITER),BOT(M,ITER),UBO(M,ITER),
     &                 IQ(M),TOA(M),IOH(M)
         END IF
            IC = IC + 1
           IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
              IC = 0
             WRITE(7,200)
           END IF
         ELSE
           IF(KEY.EQ.1) IQ(M) = IEQ(M)
           IF(KEY.EQ.2) IQ(M)=ME(M)-IOH(M)
           IF(LTOPT.EQ.1)TOA(M)=CT+ULT
           IF(LTOPT.EQ.2)THEN
             CALL ARRIVE
             TOA(M)=CT+ARV
```

```
          END IF
        IF(IP.EQ.1)THEN
          WRITE(7,503)CT,IET,M,IQ(M),TOA(M),(IOH(I),I=1,4)
        END IF
        IF(IP.EQ.2)THEN
          WRITE(7,603)CT,IET,M,IQ(M),TOA(M),IOH(M)
        END IF
           IC = IC + 1
          IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
             IC = 0
            WRITE(7,200)
          END IF
        END IF
      NOD(M,ITER) = NOD(M,ITER)+1
      OC(M,ITER) = OC(M,ITER)+OCG+OCI(M)
C
      IF(KEY.EQ.1)GO TO 20
      IF(KEY.EQ.2)GO TO 92
C
      END IF
C
C ----  DON'T ORDER UNTIL PREVIOUS ORDER ARRIVES ---
C
      IF(M.EQ.IT(M,K))THEN
       IET=0
       IF(IOH(M).LT.0) THEN
        NBQ(M,ITER)=-IOH(M)
        BOT(M,ITER)=TOA(M)-CT
        IF(NBQ(M,ITER).EQ.1)TBOT(M,ITER)=TBOT(M,ITER)+BOT(M,ITER)
        UBO(M,ITER)=UBO(M,ITER)+BOT(M,ITER)
        IF(IP.EQ.1)THEN
        WRITE(7,504)CT,IET,M,NBQ(M,ITER),BOT(M,ITER),UBO(M,ITER),
     &               (IOH(I),I=1,4)
        END IF
        IF(IP.EQ.2)THEN
        WRITE(7,604)CT,IET,M,NBQ(M,ITER),BOT(M,ITER),UBO(M,ITER),IOH(M)
        END IF
           IC = IC + 1
          IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
             IC = 0
            WRITE(7,200)
          END IF
       ELSE
        UOH(M,ITER)=UOH(M,ITER)+IOH(M)*DA(M)
        IF(IP.EQ.1)THEN
        WRITE(7,501)CT,IET,M,UOH(M,ITER),(IOH(I),I=1,4)
        END IF
        IF(IP.EQ.2)THEN
        WRITE(7,601)CT,IET,M,UOH(M,ITER),IOH(M)
        END IF
           IC = IC + 1
          IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
             IC = 0
            WRITE(7,200)
          END IF
       END IF
       GO TO 20
      ELSE
        DO 10 J=1,K
 10     IF(IQ(IT(M,J)).GT.0.OR.IQ(JT(M,J)).GT.0)GO TO 92
            IET=1
            K=K+1
            IT(M,K)=M
          IF(IOH(M).LT.0)THEN
            NBQ(M,ITER)  = -IOH(M)
            IF(KEY.EQ.1)IQ(M) = IEQ(M)+NBQ(M,ITER)
            IF(KEY.EQ.2)IQ(M) = ME(M)+NBQ(M,ITER)
            IF(LTOPT.EQ.1)TOA(M) = CT + ULT
            IF(LTOPT.EQ.2) THEN
```

```
            CALL ARRIVE
            TOA(M)=CT+ARV
          END IF
          BOT(M,ITER) = TOA(M) - CT
          IF(NBQ(M,ITER).EQ.1)TBOT(M,ITER)=TBOT(M,ITER)+BOT(M,ITER)
          UBO(M,ITER)=UBO(M,ITER)+BOT(M,ITER)
        IF(IP.EQ.1)THEN
          WRITE(7,502)CT,IET,M,NBQ(M,ITER),BOT(M,ITER),UBO(M,ITER),
     &                IQ(M),TOA(M),(IOH(I),I=1,4)
        END IF
        IF(IP.EQ.2)THEN
         WRITE(7,602)CT,IET,M,NBQ(M,ITER),BOT(M,ITER),UBO(M,ITER),
     &               IQ(M),TOA(M),IOH(M)
        END IF
          IC = IC + 1
         IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
           IC = 0
          WRITE(7,200)
         END IF
         ELSE
          IF(KEY.EQ.1)IQ(M)=IEQ(M)
          IF(KEY.EQ.2)IQ(M)=ME(M)-IOH(M)
          IF(LTOPT.EQ.1)TOA(M)=CT+ULT
          IF(LTOPT.EQ.2)THEN
            CALL ARRIVE
            TOA(M)=CT+ARV
          END IF
        IF(IP.EQ.1)THEN
          WRITE(7,503)CT,IET,M,IQ(M),TOA(M),(IOH(I),I=1,4)
        END IF
        IF(IP.EQ.2)THEN
         WRITE(7,603)CT,IET,M,IQ(M),TOA(M),IOH(M)
        END IF
          IC = IC + 1
         IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
           IC = 0
          WRITE(7,200)
         END IF
         END IF
          NOD(M,ITER) = NOD(M,ITER)+1
          OC(M,ITER) = OC(M,ITER)+OCG+OCI(M)
      END IF
      IF(KEY.EQ.1)GO TO 20
C
C  ---- DETERMINE THE JOINTLY REPLENISHABLE ITEM ----
C
 91   DO 11 I=1,NIT
      IF (IOH(I).GT.MR(I).AND.IOH(I).LE.MC(I).AND.IQ(I).EQ.0)THEN
         JT(I,K) = I
         IQ(I) = ME(I)-IOH(I)
         IF(LTOPT.EQ.1)TOA(I)=CT+ULT
         IF(LTOPT.EQ.2)THEN
          CALL ARRIVE
          TOA(I)=CT+ARV
         END IF
         NOD(I,ITER)= NOD(I,ITER)+1
         OC(I,ITER) = OC(I,ITER)+OCI(I)
         JOINT = 1
         IF(IP.EQ.1.OR.IP.EQ.2)THEN
         WRITE(7,506)I,IQ(I),TOA(I)
         END IF
          IC = IC + 1
         IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
           IC = 0
          WRITE(7,200)
         END IF
      END IF
 11   CONTINUE
        NJTO(ITER) = NJTO(ITER) +JOINT
```

```
          GO TO 20
   92   IF(IOH(M).LT.0) THEN
          NBQ(M,ITER)=-IOH(M)
          UBO(M,ITER)=UBO(M,ITER)+BOT(M,ITER)
          BOT(M,ITER)=TOA(M)-CT
          IF(NBQ(M,ITER).EQ.1)TBOT(M,ITER)=TBOT(M,ITER)+BOT(M,ITER)
          IF(IP.EQ.1)THEN
          WRITE(7,504)CT,IET,M,NBQ(M,ITER),BOT(M,ITER),UBO(M,ITER),
     &               (IOH(I),I=1,4)
          END IF
          IF(IP.EQ.2)THEN
          WRITE(7,604)CT,IET,M,NBQ(M,ITER),BOT(M,ITER),UBO(M,ITER),IOH(M)
          END IF
             IC = IC + 1
            IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
               IC = 0
              WRITE(7,200)
            END IF
        ELSE
          UOH(M,ITER)=UOH(M,ITER)+IOH(M)*DA(M)
          IF(IP.EQ.1)THEN
          WRITE(7,501)CT,IET,M,UOH(M,ITER),(IOH(I),I=1,4)
          END IF
          IF(IP.EQ.2)THEN
          WRITE(7,601)CT,IET,M,UOH(M,ITER),IOH(M)
          END IF
             IC = IC + 1
            IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
               IC = 0
              WRITE(7,200)
            END IF
        END IF
        GO TO 20
C
C ----  SET NEW ON HAND LEVEL FOR THE ITEM :'IA' ----
C
   93   CT=TA
        IET=2
        IOH(IA)=IOH(IA)+IQ(IA)
        DO 12 I=1,K
          IF(IA.EQ.IT(IA,I))THEN
            IT(IA,I)=0
            IQ(IA)=0
            NTBQ(IA,ITER)=NTBQ(IA,ITER)+NBQ(IA,ITER)
            BOT(IA,ITER)=0.
            NBQ(IA,ITER)=0
            TOA(IA)=10.**10
          END IF
        IF(IA.EQ.JT(IA,I))THEN
          JT(IA,K)=0
          IQ(IA)=0
          NTBQ(IA,ITER)=NTBQ(IA,ITER)+NBQ(IA,ITER)
          BOT(IA,ITER)=0.
          NBQ(IA,ITER)=0
          TOA(IA)=10.**10
        END IF
   12   CONTINUE
          IF(IP.EQ.1)THEN
           WRITE(7,500)CT,IET,IA,(IOH(I),I=1,4)
          END IF
          IF(IP.EQ.2)THEN
           WRITE(7,600)CT,IET,IA,IOH(IA)
          END IF
             IC = IC + 1
            IF (IC.GE.65.AND.IP.EQ.1.OR.IC.GE.65.AND.IP.EQ.2) THEN
               IC = 0
              WRITE(7,200)
            END IF
        GO TO 20
```

```
C
C  ---- COMPUTE INTER-SUMMARY (ANNUAL) ----
C
 77    IF(IP.NE.4)THEN
       WRITE(7,300)ITER
       END IF
       DO 15 I=1,NIT
       NTDM(ITER)=NTDM(ITER)+NDM(I,ITER)
       NTOD(ITER)=NTOD(ITER)+NOD(I,ITER)
       OCT(ITER)=OCT(ITER)+OC(I,ITER)
       IF(MSV.EQ.1)THEN
         IF(KB(I,ITER).GT.0)THEN
           FRC(I,ITER)=KB(I,ITER)*1.0/NOD(I,ITER)
         ELSE
           FRC(I,ITER)=0.
         END IF
          SVL(I,ITER)=(1.-FRC(I,ITER))*100.
       ELSE
         IF(NTBQ(I,ITER).GT.0)THEN
           FRQ(I,ITER)=NTBQ(I,ITER)*1.0/NDM(I,ITER)
         ELSE
           FRQ(I,ITER)=0.
         END IF
           SVL(I,ITER)=(1.-FRQ(I,ITER))*100.
       END IF
 15    CONTINUE
       IF(IP.NE.4)THEN
       DO 16 I=1,NIT
          WRITE(7,507)I,NDM(I,ITER),NOD(I,ITER),UOH(I,ITER),UBO(I,ITER),
      &NTBQ(I,ITER),TBOT(I,ITER),SVL(I,ITER),IOH(I)
 16    CONTINUE
       WRITE(7,508)NTDM(ITER) ,NTOD(ITER)
       WRITE(7,509)NJTO(ITER)
       END IF
C
C ----  COMPUTE COSTS  ----
C
       DO 17 I=1,NIT
        HCOST(I,ITER)=UOH(I,ITER)*H(I)
        SCTT(I,ITER)=UBO(I,ITER)*(SCT(I)+H(I))
        SCTU(I,ITER)=NTBQ(I,ITER)*SCU(I)
        SCOST(I,ITER)=SCTU(I,ITER)+SCTT(I,ITER)
        COST(I,ITER)=HCOST(I,ITER)+SCOST(I,ITER)+OC(I,ITER)
 17     TCOST(ITER)=TCOST(ITER)+COST(I,ITER)
       IF(IP.NE.4)THEN
        WRITE(7,514)
        DO 18 I=1,NIT
 18     WRITE(7,515) I,HCOST(I,ITER),SCTU(I,ITER),SCTT(I,ITER),
      &                OC(I,ITER),COST(I,ITER)
       WRITE(7,516)OCT(ITER),TCOST(ITER)
       END IF
       DO 19 I=1,NIT
 19       KBT(I)=KBT(I)+KB(I,ITER)
 777   CONTINUE
C
C  ---- COMPUTE FINAL-RESULTS (MEAN AND STD. DEVIATIONS) ----
C
       WRITE(7,510)NSIM
       DO 21 I=1,NIT
       DO 21 J=1,ITER-1
       NTDMS(I)=NTDMS(I)+NDM(I,J)
       NTODS(I)=NTODS(I)+NOD(I,J)
       TUOH(I)=TUOH(I)+UOH(I,J)
       TUBO(I)=TUBO(I)+UBO(I,J)
       TFRT(I)=TFRT(I)+TBOT(I,J)
       TFRQ(I)=TFRQ(I)+NTBQ(I,J)*1.0
       OCTT(I)=OCTT(I)+OC(I,J)
       THCOST(I)=THCOST(I)+HCOST(I,J)
       TSCTT(I)=TSCTT(I)+SCTT(I,J)
```

```
21    TSCTU(I)=TSCTU(I)+SCTU(I,J)
      DO 22 I=1,NIT
      ANDM(I)=NTDMS(I)/TMAX
      ANOD(I)=NTODS(I)/TMAX
      AUOH(I)=TUOH(I)/TMAX
      AUBO(I)=TUBO(I)/TMAX
      ABOT(I)=TFRT(I)/TMAX
      ABOQ(I)=TFRQ(I)/TMAX
      ANB(I) =KBT(I)*1.0/NTODS(I)
      AFRQ(I)=TFRQ(I)/NTDMS(I)
      IF(MSV.EQ.1) ASVL(I)=(1.-ANB(I))*100.
      IF(MSV.EQ.2) ASVL(I)=(1.-AFRQ(I))*100.
      AOC(I)=OCTT(I)/TMAX
      AHCOST(I)=THCOST(I)/TMAX
      ASCTU(I)=TSCTU(I)/TMAX
      ASCTT(I)=TSCTT(I)/TMAX
      ACOST(I)=AHCOST(I)+ASCTU(I)+ASCTT(I)+AOC(I)
22    TVC=TVC+ACOST(I)
      DO 23 I=1,ITER-1
23    NTJTO=NTJTO+NJTO(I)
      IF(NTJTO.GT.0)ANJ = NTJTO/TMAX
      DO 24 I=1,ITER-1
24     STV=STV+(TCOST(I)-TVC)**2
       STDVC=SQRT(STV/(ITER-2))
      DO 25 I=1,NIT
25    WRITE(7,511)I,ANDM(I),ANOD(I),AUOH(I),AUBO(I),ABOQ(I),ABOT(I),
     &           ASVL(I)
      DO 26 I=1,NIT
       TANDM=TANDM+ANDM(I)
       TANOD=TANOD+ANOD(I)
       TAOC=TAOC+AOC(I)
26    CONTINUE
      WRITE(7,512)TANDM,TANOD
      WRITE(7,513)ANJ
      WRITE(7,517)
       DO 27 I=1,NIT
27     WRITE(7,515) I,AHCOST(I),ASCTU(I),ASCTT(I),AOC(I),ACOST(I)
      WRITE(7,516)TAOC,TVC
      WRITE(7,518)STDVC
      STOP
97    FORMAT(2X,' ',  ///,' * INPUT DATA FOR COSTS:',//,' ITEM',2X,'UNIT
     & PRICE',2X,'H.RATE',2X,'H.COST',3X,'  ORDER COST',9X,'B/O COST',
     &/,36X,'GROUP',6X,'IND.',4X,'QTY.',5X,'TIME')
98    FORMAT(2X,I2,4X,7(F6.2,3X))
100   FORMAT(F7.3,2X,5(I3,2X),3(F6.2,2X))
101   FORMAT(I3,3X,F6.2,3X,I2,3X,2(I1,3X))
198   FORMAT(//,5X,'** INPUT DATA :',/,3X,/,12X,'* USED DATA FILE NO. ='
     &,I3,/,12X,'* ORDER POLICY OPTION =',I3,/,12X,'* SIMIMULATION LIMIT
     &(YRS) =',I3 , /,12X,'* START LEVEL OPTION =',I3,/,12X,'* PRINTOUT
     &OPTION =',I3,/,12X,'* LEAD TIME TYPE  =',I3,/,12X,///,7X,'* ORDER
     &POLICY OPTIONS',//,10X,'1 : INDIVIDUAL ORDER POLICY',/,10X,'2 : JO
     &INT ORDER POLICY',//,7X,'* START LEVEL OPTIONS',//,10X,'1 : REGULA
     &R (IND. POLICY W/ E.O.Q ; JOINT POLICY W/  ORDER-UP QTY.)',/,10X,'
     &2 : RANDOM (REORDER POINT + 1; ORDER-UP POINT)',//,7X,'* PRINTOUT
     &OPTIONS',//,10X,'1 : PRINT ALL ITEM',/,10X,'2 : PRINT ONLY EVENT
     &ITEM',/,10X,'3 : PRINT ONLY SUMMARY',/,10X,'4 : PRINT ONLY FINAL R
     &ESULT',//,7X,'* LEAD TIME TYPES ',//,10X,'1 : COSTANT LEADTIME',/,
     &10X,'2 : RANDOM LEADTIME (UNIFORM W/ MEAN)',//,6X,'* SERVICE LEVEL
     & : P',I1, '=',I3,1X,'%',5X,'* AVG. LEAD TIME : ',F6.3,' YRS.','(',
     &I2,' MON.)',///,3X,'ITEM',3X,'YR DEMAND',4X,'EOQ',6X,'MUST BUY',
     &3X,'CAN BUY',3X,'ORD. UP QTY',3X,'START LEVEL',3X,'EXP. LT.')
199   FORMAT(3X,I2,6X,F7.3,5X,4(I3,8X),3X,I3,7X,F7.3)
200   FORMAT('1',//,35X,'**  SIMULATION RESULTS  **',//,20X,'(EVENT TYPE
     &    0 : DEMAND OCCURED    1 : ORDER PLACED    2 : ORDER QTY ARRIVED)'
     &,//,5X,'EVENT',4X,'ITEM #',2X,'BACK ORDER',3X,'UNIT YRS(QTY)',4X,'
     &ON ORDER',10X,'ON HAND',9X,'JOINT REPLENISHMENT',/,1X,'TIME',2X,
     &'*TYPE',10X,'QTY.',2X,'TIME',3X,'ON HAND',2X,'B/O.S',3X,'QTY.',2X,
     &'AR/T.',4X,'1',4X,'2',4X,'3',4X,'4',2X,'ITEM#',2X,'OD. QTY.',2X,'O
     &D.AR/T.')
```

```
300  FORMAT('1',///,20X,'***  INTER - SUMMARY  (ANNUAL)  ***',//, 20X,
    &'AT THE YEAR  :',2X,I3,//,3X,'ITEM #',3X,'# DEMANDS',3X,'# ORDERS'
    &,10X,' UNIT YEAR ',7X,'BACKORDER',8X,'SERVICE',6X,'F/ OH',/,40X,
    &'ON HAND',3X,' B/O  ',2X,'QTY.',3X,'TIME',5X,'LEVEL(%)')
500  FORMAT(' ',F6.4,2X,I2,3X,I2,48X,4(I4,1X))
600  FORMAT(' ',F6.4,2X,I2,3X,I2,54X,I4)
501  FORMAT(' ',F6.4,2X,I2,3X,I2,18X,F8.2,22X,4(I4,1X))
601  FORMAT(' ',F6.4,2X,I2,3X,I2,18X,F8.2,28X,I4)
502  FORMAT(' ',F6.4,2X,I2,3X,I2,5X,I3,3X,F6.3,11X,F6.2,3X,I3,1X,
    & F6.3,1X,4(I4,1X))
602  FORMAT(' ',F6.4,2X,I2,3X,I2,5X,I3,3X,F6.3,11X,F6.2,3X,I3,1X,
    & F6.3,7X,I4)
503  FORMAT(' ',F6.4,2X,I2,3X,I2,37X,I3,1X,F6.3,1X,4(I4,1X))
603  FORMAT(' ',F6.4,2X,I2,3X,I2,37X,I3,1X,F6.3,7X,I4)
504  FORMAT(' ',F6.4,2X,I2,3X,I2,5X,I3,3X,F6.3,11X,F6.2,14X,4(I4,1X))
604  FORMAT(' ',F6.3,2X,I2,3X,I2,5X,I3,3X,F6.3,11X,F6.2,20X,I4)
506  FORMAT(86X,I2,6X,I3,5X,F6.3)
507  FORMAT(5X,I2,9X,I4,7X,I3,9X,F9.4,2X,F6.3,2X,I3,3X,F6.3,5X,F6.2,
    &      9X,I4)
508  FORMAT(1X,'TOTAL =',7X,I5,5X,I5)
509  FORMAT(1X,'TOTAL NUMBER OF JOINT REPLENISHMENT =',I5)
510  FORMAT('1',///,20X,'** FINAL RESULTS (AVG.) :',10X,'DURING',I3,2X,
    &'YEARS   ***',//,3X,'ITEM #',3X,'# DEMANDS',3X,'# ORDERS',9X,'AVG
    &. UNIT YEAR',4X,'AVG. B/O',5X,'SERVICE' ,/,40X,
    &'ON HAND',3X,'B/O.S',4X,'QTY.',2X,'TIME',3X,'LEVEL','(','%',')')
511  FORMAT(5X,I2,6X,F6.2,5X,F6.2,10X,F7.3,1X,F7.3,1X,F6.2,2X,F7.3,
    &2X,F6.2)
512  FORMAT(1X,'TOTAL AVG.=',F7.2,5X,F6.2)
513  FORMAT(1X,'AVG. NUMBER OF JOINT REPLENISHMENT =',F6.3)
514  FORMAT(1X,///,10X,' * COMPUTED ANNUAL COSTS ($)',//,2X,
    &'ITEM',3X,'HOLDING',9X,'BACKORDER',9X,'ORDER',6X,'TOTAL',/,22X,'QT
    &Y.',7X,'TIME')
515  FORMAT(2X,I2,5X,5(F7.3,4X))
516  FORMAT(1X,/,33X,'TOTAL = ',2(F8.3,3X))
517  FORMAT(1X,///,10X,' * COMPUTED AVG. ANNUAL COSTS ($)',//,2X,
    &'ITEM',3X,'HOLDING',9X,'BACKORDER',9X,'ORDER',6X,'TOTAL',/,22X,'QT
    &Y.',7X,'TIME')
518  FORMAT(1X,/,33X,'STAND. DEVIATION = ',F8.3)
     END
C
                    SUBROUTINE DEMAND
C
     DIMENSION ISEED(10),DM(10),DA(100),MR(10),ME(10),IEQ(10)
     COMMON ICASE,ISEED,DM,DA,ARV,ULT,I,ISTP,MR,ME,KEY,IEQ
     CALL LRND (ISEED(ICASE),U,1,1,0)
       DA(ICASE)=-(1./DM(ICASE))*ALOG(U)
     RETURN
     END
C
C
                    SUBROUTINE ARRIVE
C
     DIMENSION ISEED(10),DM(10),DA(100),MR(10),ME(10),IEQ(10)
     COMMON ICASE,ISEED,DM,DA,ARV,ULT,I,ISTP,MR,ME,KEY,IEQ
     CALL LRND(ISEED(I),U,1,1,0)
     ARV= 2.*ULT*U
     RETURN
     END
C
C
                    SUBROUTINE START
C
     DIMENSION ISEED(10),DM(10),DA(100),MR(10),ME(10),IEQ(10)
     COMMON ICASE,ISEED,DM,DA,ARV,ULT,I,ISTP,MR,ME,KEY,IEQ
     IF(KEY.EQ.1) THEN
         CALL LRND(ISEED(I),U,1,1,0)
         IS= (IEQ(I)-(MR(I)+1))*U
         ISTP= IS+MR(I)+1
         RETURN
```

```
      END IF
      IF(KEY.EQ.2) THEN
         CALL LRND(ISEED(I),U,1,1,0)
           IS= (ME(I)-(MR(I)+1))*U
           ISTP= IS+MR(I)+1
         RETURN
      END IF
      END
C
      SUBROUTINE OPTION(ID,RATE1,RATE2,KEY,LTOPT,NSIM,ISTA,IP)
C
          WRITE(6,35)
          READ(5,34) ID
          WRITE(6,39)
          READ(5,40)RATE1,RATE2
          WRITE(6,31)
          READ(5,34) KEY
          WRITE(6,37)
          READ(5,34) LTOPT
          WRITE(6,36)
          READ(5,30) NSIM
          WRITE(6,32)
          READ(5,34) ISTA
          WRITE(6,33)
          READ(5,34) IP
      RETURN
 30   FORMAT(I2)
 31   FORMAT('1','   * ENTER OPTION FOR ORDER POLICY (1 - 2) ! ',/,5X, '
     &1 = INDIVIDUAL ORDER POLICY',/, 5X,'2 = JOINT ORDER POLICY ')
 32   FORMAT('1',' * ENTER OPTION  FOR START LEVEL (1 - 2) ! ',/,5X,'1
     &= REGULAR (IND.ORDER POLICY : E.O.Q / JOINT ORDER POLICY : ORDER U
     &P QTY.)',/,5X,'2 = RANDOM BETWEEN (MUST ORDER LEVEL+1) AND (ORDER
     &UP LEVEL)')
 33   FORMAT('1','  * ENTER OPTION FOR PRINTOUT(1 - 4) ! ',/,5X,'1 = PRI
     &NT ALL ITEM',/,5X,'2 = PRINT ONLY EVENT  ITEM',/,5X,'3 = PRINT ONL
     &Y SUMMARY ',/,5X,'4 = PRINT ONLY FINAL RESULT')
 34   FORMAT(I1)
 35   FORMAT('1','  * ENTER OPTION FOR DATA FILE (1 - 4)    ! ',/,
     &5X,'1 = DATA1',3X,'2 = DATA2',3X,'3 = DATA3',3X,'4 = DATA4')
 36   FORMAT('1','  * ENTER SIMULATION LIMIT YEARS     !',/,5X,'01 = SIMU
     &LATION LIMIT IS 1 YEAR',/,5X,'02 = SIMULATION LIMIT IS 2 YEARS'
     & ,/,7X,'  ....  ',/,5X,'99 = SIMULATION LIMIT IS 99 YEARS ')
 37   FORMAT('1','  * ENTER THE TYPE OF LEADTIME(1 - 2) !',/,5X,'1 = CON
     &STANT LEADTIME',/,5X,'2 = RANDOM (UNIFORM  WITH MEAN)')
 39   FORMAT('1','  * ENTER RATE OF STOCKOUT COST TO UNIT PRICE !',/,5X
     &,'1.25 0.91 = STOCKOUT COST PER UNIT IS 125% OF UNIT PRICE',/,17X,
     &'STOCKOUT COST PER TIME IS 91% OF UNIT PRICE')
 40   FORMAT(2(F4.2,1X))
      END
```

```
   ---------------------------------------------------
   1. DATA FORMAT ( P2 = 0.99, for items in group 1 )
   ---------------------------------------------------

004     050.00   99    2    1
290.000  025    025   112  177  159  010.00  006.90  000.20
041.000  003     -1   035  095  143  010.00  001.20  000.20
077.000  007    005   048  093  109  010.00  003.90  000.20
122.000  010    007   077  150  178  010.00  002.30  000.20
```

```
----------------
2. EXEC PROGRAM
----------------
&TRACE OFF
&FN = SIM
&FNO = &CONCAT OF &FN OUTPUT
&TYPE Do you need to compile your program ? (Y)
&READ VAR &R_COMPILE
&IF &R_COMPILE NE Y &GOTO -RUN
-H FORTVS &FN
&IF &RC EQ 0 &GOTO -RUN
&TYPE Your program did not compile; check for errors.
&TYPE Do you wish to view the program LISTING file? (Y)
&READ VAR &RSP1
&IF &RSP1 EQ Y BROWSE &FN LISTING A
&TYPE Do you wish to XEDIT the program file? (Y)
&READ VAR &RESP1
&IF &RESP1 NE Y &EXIT 1
&COMMAND XEDIT &FN FORTRAN A
&TYPE Do you wish to run the program again? (Y)
&READ VAR &RESP2
&IF &RESP2 EQ Y &GOTO -H
&EXIT 1
-RUN
FILEDEF 01 DISK &FN DATA1 A1
FILEDEF 02 DISK &FN DATA2 A1
FILEDEF 03 DISK &FN DATA3 A1
FILEDEF 04 DISK &FN DATA4 A1
FILEDEF 07 DISK &FN OUTPUT A1 (LRECL 133
LOAD &FN (START
&IF &RC EQ 0 &SKIP 9
&TYPE Your program did not run correctly; check for errors.
&TYPE Do you wish to XEDIT the program file? (Y)
&READ VAR &RESP3
&IF &RESP3 NE Y &EXIT 2
&COMMAND XEDIT &FN FORTRAN A
&TYPE Do you wish to run the program again? (Y)
&READ VAR &RESP4
&IF &RESP4 EQ Y &GOTO -H
&EXIT 2
&TYPE YOUR OUTPUT IS IN THE FILE &FN OUTPUT A
&TYPE Do you wish  to BROWSE your output? (Y)
&READ VAR &RESP
&IF &RESP EQ Y &COMMAND BROWSE &FN OUTPUT A
&TYPE Print your output file?  (Y)
&READ VAR &RESP7
&IF &RESP7 EQ Y &COMMAND PRINT &FN OUTPUT A
-REDO
&TYPE Do you wish to XEDIT the program file? (Y/N)
&READ VAR &RESP5
&IF &RESP5 EQ Y XEDIT &FN FORTRAN A
&TYPE Do you wish to run the program again? (Y)
&READ VAR &RESP6
&RESP56 = &CONCAT OF &RESP5 &RESP6
&IF &RESP56 EQ YY &GOTO -H
&IF &RESP6 EQ Y &GOTO -RUN
&EXIT
```

3. RUN EXAMPLE

```
sim
Do you need to compile your program ? (Y)
y
VS FORTRAN COMPILER ENTERED.  12:11:11

**MAIN** END OF COMPILATION 1 ******

**DEMAND** END OF COMPILATION 2 ******

**ARRIVE** END OF COMPILATION 3 ******

**START** END OF COMPILATION 4 ******

**OPTION** END OF COMPILATION 5 ******
VS FORTRAN COMPILER EXITED.   12:11:23

EXECUTION BEGINS...

  * ENTER OPTION FOR DATA FILE (1 - 4)    !
    1 = DATA1    2 = DATA2    3 = DATA3    4 = DATA4
1

  * ENTER RATE OF STOCKOUT COST TO UNIT PRICE !
    1.25 0.91 = STOCKOUT COST PER UNIT IS 125% OF UNIT PRICE
                STOCKOUT COST PER TIME IS 91% OF UNIT PRICE
0.40 0.01

   * ENTER OPTION FOR ORDER POLICY (1 - 2) !
    1 = INDIVIDUAL ORDER POLICY
    2 = JOINT ORDER POLICY
1

  * ENTER THE TYPE OF LEADTIME(1 - 2) !
    1 = CONSTANT LEADTIME
    2 = RANDOM (UNIFORM  WITH MEAN)
1

  * ENTER SIMULATION LIMIT YEARS     !
    01 = SIMULATION LIMIT IS 1 YEAR
    02 = SIMULATION LIMIT IS 2 YEARS
      ....
    99 = SIMULATION LIMIT IS 99 YEARS
50

  * ENTER OPTION  FOR START LEVEL (1 - 2) !
    1 = REGULAR(IND.ORDER POLICY: E.O.Q / JOINT ORDER POLICY: OD.UP QTY.)
    2 = RANDOM BETWEEN (MUST ORDER LEVEL+1) AND (ORDER UP LEVEL)
2

  * ENTER OPTION FOR PRINTOUT(1 - 4) !
    1 = PRINT ALL ITEM
    2 = PRINT ONLY EVENT  ITEM
    3 = PRINT ONLY SUMMARY
    4 = PRINT ONLY FINAL RESULT
4
YOUR OUTPUT IS IN THE FILE SIM OUTPUT A
Do you wish to BROWSE your output? (Y)
y
Print your output file? (Y)
n
Do you wish to XEDIT the program file? (Y/N)
n
Do you wish to run the program again? (Y)
n
R; T=4.63/5.82 12:11:54
```

4. OUTPUT 1 (individual order policy)

** INPUT DATA :

* USED DATA FILE NO. = 1
* ORDER POLICY OPTION = 1
* SIMIMULATION LIMIT(YRS) = 50
* START LEVEL OPTION = 2
* PRINTOUT OPTION = 4
* LEAD TIME TYPE = 1

* ORDER POLICY OPTIONS

1 : INDIVIDUAL ORDER POLICY
2 : JOINT ORDER POLICY

* START LEVEL OPTIONS

1 : REGULAR (IND. POLICY - E.O.Q ; JOINT POLICY - OD.UP QTY.)
2 : RANDOM (REORDER POINT + 1; ORDER-UP POINT)

* PRINTOUT OPTIONS

1 : PRINT ALL ITEM
2 : PRINT ONLY EVENT ITEM
3 : PRINT ONLY SUMMARY
4 : PRINT ONLY FINAL RESULT

* LEAD TIME TYPES

1 : COSTANT LEADTIME
2 : RANDOM LEADTIME (UNIFORM W/ MEAN)

* SERVICE LEVEL: P2= 99 %   * AVG. LEAD TIME: 0.083 YRS.(1 MON.)

ITEM	YR DEMAND	EOQ	MUST BUY	CAN BUY	ORD. UP	START	EXP. LT.
1	290.000	159	25	112	177	65	0.083
2	41.000	143	3	35	95	45	0.083
3	77.000	109	7	48	93	38	0.083
4	122.000	178	10	77	150	60	0.083

* INPUT DATA FOR COSTS:

ITEM	UNIT PRICE	H.RATE	H.COST	ORDER COST		B/O COST	
				GROUP	IND.	QTY.	TIME
1	6.90	0.20	1.38	50.00	10.00	2.76	0.07
2	1.20	0.20	0.24	50.00	10.00	0.48	0.01
3	3.90	0.20	0.78	50.00	10.00	1.56	0.04
4	2.30	0.20	0.46	50.00	10.00	0.92	0.02

** FINAL RESULTS (AVG.) : DURING 50 YEARS ***

ITEM #	# DEMANDS	# ORDERS	AVG. UNIT YEAR		AVG. B/O		SERVICE
			ON HAND	B/O.S	QTY.	TIME	LEVEL(%)
1	292.24	1.84	79.151	0.025	2.92	0.009	99.00
2	41.58	0.30	69.217	0.008	0.34	0.006	99.18
3	77.00	0.72	54.064	0.004	0.46	0.003	99.40
4	120.50	0.68	88.760	0.014	0.84	0.006	99.30
TOT AVG.;	531.32	3.54					

AVG. NO. OF JOINT REP; 0.000

* COMPUTED AVG. ANNUAL COSTS ($)

ITEM	HOLDING	BACKORDER QTY.	BACKORDER TIME	ORDER	TOTAL
1	109.228	8.059	0.036	110.400	227.723
2	16.612	0.163	0.002	18.000	34.777
3	42.170	0.718	0.003	43.200	86.090
4	40.830	0.773	0.007	40.800	82.409
			TOTAL =	212.400	430.999
			STAND. DEVIATION =		51.942

5. OUTPUT 2 (joint order policy)

**** INPUT DATA :**

- * USED DATA FILE NO. = 1
- * ORDER POLICY OPTION = 2
- * SIMIMULATION LIMIT(YRS) = 50
- * START LEVEL OPTION = 2
- * PRINTOUT OPTION = 4
- * LEAD TIME TYPE = 1

* ORDER POLICY OPTIONS

1 : INDIVIDUAL ORDER POLICY
2 : JOINT ORDER POLICY

* START LEVEL OPTIONS

1 : REGULAR (IND. POLICY - E.O.Q ; JOINT POLICY - OD. UP QTY.)
2 : RANDOM (REORDER POINT + 1; ORDER-UP POINT)

* PRINTOUT OPTIONS

1 : PRINT ALL ITEM
2 : PRINT ONLY EVENT ITEM
3 : PRINT ONLY SUMMARY
4 : PRINT ONLY FINAL RESULT

* LEAD TIME TYPES

1 : COSTANT LEADTIME
2 : RANDOM LEADTIME (UNIFORM W/ MEAN)

* SERVICE LEVEL: P2= 99 %   * AVG. LEAD TIME: 0.083 YRS.(1 MON.)

ITEM	YR DEMAND	EOQ	MUST BUY	CAN BUY	ORD. UP	START	EXP. LT.
1	290.000	159	25	112	177	71	0.083
2	41.000	143	-1	35	95	28	0.083
3	77.000	109	5	48	93	32	0.083
4	122.000	178	7	77	150	50	0.083

* INPUT DATA FOR COSTS:

ITEM	UNIT PRICE	H.RATE	H.COST	ORDER COST		B/O COST	
				GROUP	IND.	QTY.	TIME
1	6.90	0.20	1.38	50.00	10.00	2.76	0.07
2	1.20	0.20	0.24	50.00	10.00	0.48	0.01
3	3.90	0.20	0.78	50.00	10.00	1.56	0.04
4	2.30	0.20	0.46	50.00	10.00	0.92	0.02

** FINAL RESULTS (AVG.) :   DURING 50 YEARS   ***

ITEM #	# DEMANDS	# ORDERS	AVG. UNIT YEAR		AVG. B/O		SERVICE
			ON HAND	B/O.S	QTY.	TIME	LEVEL(%)
1	292.24	1.94	75.843	0.021	2.56	0.007	99.12
2	41.58	0.60	56.035	0.000	0.00	0.000	100.00
3	77.00	1.18	51.541	0.001	0.04	0.001	99.95
4	120.50	1.06	80.055	0.003	0.30	0.001	99.75
TOT AVG.;	531.32	4.78					

AVG. NO. OF JOINT REP; 1.620

*** COMPUTED AVG. ANNUAL COSTS ($)**

ITEM	HOLDING	BACKORDER QTY.	BACKORDER TIME	ORDER	TOTAL
1	104.663	7.066	0.030	113.400	225.159
2	13.448	0.000	0.000	6.000	19.448
3	40.202	0.062	0.001	12.800	53.065
4	36.825	0.276	0.001	13.600	50.703
			TOTAL =	145.800	348.375
			STAND. DEVIATION =		21.055

LIST OF REFERENCES

1. Ignall, Edward, "Optimal Continuous review Policies for Two Product Inventory Systems with Joint setup costs." *Management Science* Vol. 15, NO. 5, 1969.

2. Baltify, Joseph L., "On a Basic Class of Multi-Item Inventory Problems." *Management Science* Vol. 10, NO. 2, 1964.

3. Silver, Edward A., "A control System for Coordinated Inventory Replenishment." *Int. J, Prod. Res.* Vol. 12, NO. 6, 1974.

4. Schaack, J. P., and Silver, Edward A., "A Procedure, Inventory Simulation, for Selecting the Control Variables of an (S,c,s) Joint Ordering Strategy." *Infor* Vol. 10, NO. 2, June 1972.

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

6. Johnson, Lynwood A. and Montgomery Douglas C., *Operations Research in Production Planning, Scheduling, and Inventory Control*, John Wiley &Sons, 1975.

7. Buchan, J., and Koenigsberg, E., *Scientific Inventory Management*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

8. Tersine, Richard J., *Principles of Inventory and Materials Management*, North-Holland, 1982.

9. Silver, Edward A., and Wilson, T. G., "Cost Penalities of Simplified Procedures for Selecting Reorder Points and Order Quanties." *Procedings of 15th Annual International Conference of the American Production and Inventory Control Society* 219, 1972.

10. Silver, Edward A., "Three Ways of Obtaining The Average Cost Expression in a Problem Related to Joint Replenishment Inventory Control." *Nav. Res. Logist. Q.*, Vol. 20, 1972.

11. Taylor, Howard M. and Karlin, Samuel, *An Introduction to Stochastic Modeling*, Academic Press, Inc., 1984.

12. Graves, Stephen C., "The Application of Queueing Theory to Continious perishable Inventory Systems." *Management Science* Vol. 28, NO. 4, 1982.

13. Morgan, Byron J. T., *Elements of Simulation*, Chapman and Hall, 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Professor Francis R. Richards, Code 55Rh Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	1
4.	Professor James D. Esary Code 55Ey Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor Peter Purdue, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	1
6.	ADCNO for Education & Training Division P.O.Box 201-17 Youngdeongpo Gu Singil 7 Dong Seoul Korea	2
7.	Choi, seok c. 1103 Sonoma Ave. APT. #3 Seaside, California 93955	1
8.	Professor Baek, Jong Hyun Sin-Soo Dong 1 So-Gang university 121 Ma-Po Gu, Seoul Korea	2
9.	Kim, Won Bong Sin-Rim 11 Dong 746-43 Gun-Young APT., Ga Dong 409 Ho 152 Young-Dung-Po Gu, Seoul Korea	12

END

7-87

DTIC