ANALYSIS OF INVENTORY/RESUPPLY SUB-SYSTEM MODELS IN OPTIMAL MATERIEL DISTRIBUTION SYSTEM DESIGN

Arthur Z. Kovacs

April 1978

School of Engineering
UNIVERSITY OF DAYTON
DAYTON, OHIO
ANALYSIS OF INVENTORY/RESUPPLY SUB-SYSTEM MODELS IN OPTIMAL MATERIEL DISTRIBUTION SYSTEM DESIGN

Final Report
by
Arthur Z. Kovacs

April 30, 1978
30 Apr 78

Engineering Services
School of Engineering
University of Dayton
Dayton, Ohio 45469

for

Department of the Air Force
Air Force Office of Scientific Research
Bolling Air Force Base, D.C. 20332

Directorate of Mathematical and Information Sciences
FOREWARD

The work reported in this document was accomplished by Dr. Arthur Z. Kovacs under Grant AFOSR-77-3266 awarded to the University of Dayton by the Department of the Air Force, Air Force Office of Scientific Research, Directorate of Mathematical and Information Sciences, Bolling Air Force Base. The author thanks the Air Force Office of Scientific Research which made this work possible and the Air Force Logistics Command which introduced him to this problem area during the 1976 United States Air Force/American Society for Engineering Education Summer Faculty Research Program at Wright-Patterson Air Force Base, Ohio. In particular, he thanks Douglas Allen, a student in the Department of Mechanical Engineering at the University of Dayton, for his excellent work in writing the computer programs used to perform the analyses described in this report. Finally, special thanks is given to Dr. David C. Kraft, who as Dean of the School of Engineering, provided valuable support toward the completion of this project.
ABSTRACT

The purpose of the Air Force freight distribution system is to meet spare parts demand requirements at minimum cost. Budgetary constraints have suggested that total expected backorder level for items at user installations be minimized, subject to a given dollar expenditure level for inventory investment. LOGAIR (a dedicated Air Force air transport service) is a major transportation sub-system to support spare parts delivery requirements of users of high priority items. A two-echelon inventory system for spare parts delivery exists with centralized, specialized inventories at the Air Logistics Centers (ALCs) and decentralized, broad profile inventories located at user installations. LOGAIR provides transport with low order and ship time to reduce resupply time in the maintenance of inventory safety levels at user bases. A systems approach is used to formulate a cost/benefit model, which recognizes the impact of the Air Force resupply system upon the total spare parts distribution system in terms of total inventory investment level, total system cost, and backorder level. Given a total expenditure level available for allocation between inventory investment and transportation, the problem is to determine the optimal fractional allocation to be made to transportation (the remaining fraction to be allocated to inventory investment) such that total expected backorder level is minimized. A conceptual framework for trade-off analysis for minimizing total system cost in terms of inventory investment level and resupply time level for a given backorder level is presented. This conceptual framework also allows for the minimization of backorder level in terms of inventory investment level and resupply time level for a given total dollar expenditure level. Queueing models exhibiting various demand and resupply processes are explored and compared to determine the impact of inventory investment level and resupply time level upon backorder level. Specific solution procedures are developed for and are applied to the trade-off analyses mentioned above.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COVER PAGE</td>
<td>i</td>
</tr>
<tr>
<td>FOREWARD</td>
<td>ii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iv</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1-1</td>
</tr>
<tr>
<td>2 GENERAL MATHEMATICAL STATEMENT OF PROBLEM</td>
<td>2-1</td>
</tr>
<tr>
<td>3 DEMAND AND RESUPPLY PROCESSES IN INVENTORY MODELING</td>
<td>3-1</td>
</tr>
<tr>
<td>4 CONSERVATIVE PARALLEL SERVER INVENTORY MODEL</td>
<td>4-1</td>
</tr>
<tr>
<td>5 POISSON MODEL</td>
<td>5-1</td>
</tr>
<tr>
<td>6 CONSERVATIVE SINGLE SERVER INVENTORY MODEL</td>
<td>6-1</td>
</tr>
<tr>
<td>7 EXAMPLE COST FUNCTION</td>
<td>7-1</td>
</tr>
<tr>
<td>8 PROBLEM FORMULATION</td>
<td>8-1</td>
</tr>
<tr>
<td>9 PROBLEM SOLUTION</td>
<td>9-1</td>
</tr>
<tr>
<td>10 A GRADIENT TECHNIQUE TO SOLVE A SET OF COUPLED EQUATIONS</td>
<td>10-1</td>
</tr>
<tr>
<td>11 SENSITIVITY ANALYSIS</td>
<td>11-1</td>
</tr>
<tr>
<td>12 RESULTS AND CONCLUSIONS</td>
<td>12-1</td>
</tr>
<tr>
<td>13 REFERENCE LIST AND BIBLIOGRAPHY</td>
<td>13-1</td>
</tr>
<tr>
<td>14 DOCUMENTS INTERNAL TO AFLC</td>
<td>14-1</td>
</tr>
<tr>
<td>APPENDIX A PROGRAM LISTINGS</td>
<td>A-1</td>
</tr>
<tr>
<td>APPENDIX B EXAMPLE NUMERICAL RESULTS</td>
<td>B-1</td>
</tr>
</tbody>
</table>
SECTION 1
INTRODUCTION

The purpose of the Air Force freight distribution system is to meet spare parts demand requirements of user installations at minimum cost. Budgetary constraints have suggested that total expected backorder level for items at user installations be minimized subject to a given dollar expenditure level for inventory investment. LOGAIR (a dedicated Air Force air transport service) is a major transportation sub-system to support spare parts delivery requirements to users of high priority items. A two-echelon inventory support system for spare parts delivery exists with centralized, specialized inventories at the Air Logistics Centers (ALCs) and decentralized, broad profile inventories located at user installations. LOGAIR provides transport with low order and ship time to reduce resupply time in the maintenance of inventory safety levels at the user bases.

The total systems problem is to determine an optimal logistics support system considering the cost trades-off which are possible among inventory, repair and maintenance, and transportation sub-systems as well as management policies regarding procurement and level of service. Within this framework, the transportation system is to be developed to utilize available transportation modes in servicing requirements in a timely, cost-effective manner. Any mode which is dedicated to Air Force needs (LOGAIR, truck) must be defined in detail to include route specification, vehicle schedules, and managerial procedures. Thus, a systems approach is proposed to explore formulation of a cost/benefit model which recognizes the impact of the Air Force resupply system upon the total spare parts distribution system in terms of total inventory investment level, total system cost, and backorder level. This approach recognizes that the purpose of expenditures for transportation is to reduce resupply time in the support of inventory policy designed to meet user needs.
Implicit in the approach taken by any inventory model is the assumption of the existence of a transportation system capable of delivery of an order and ship time (O&ST) profile which makes up part of the resupply time required for input to the model. The Air Force freight transportation system is supposed to provide timely delivery of inventory items to users from the supply depots and to ship items from the user installations back to the depots for repair. A transportation system with low average order and ship time levels should yield correspondingly low levels of required inventory investment at a cost which is small compared to any resultant savings in inventory investment. Another purpose of the transportation system is to minimize resupply time in the event of a stockout.

The problem then becomes:

Given a total expenditure level available for allocation between inventory investment and transportation, what is the optimal fractional allocation to be made to transportation (the remaining fraction to be allocated to inventory investment) such that the total expected back-order level at the user installations is minimized?

In this allocation, it is assumed that expenditures are to be expended in an optimal fashion. Inventory models which can treat consumable as well as recoverable items are to optimally allocate total inventory investment among supply depots and user installations subject to a given input O&ST profile. A transportation network and mode selection model can be required to optimally utilize a given transportation expenditure level to yield an optimum O&ST profile that minimizes average O&ST level experienced within the network. Or, conversely, a transportation network and optimal mode selection model might be designed to meet specific O&ST standards for all shipments in such a fashion as to minimize total transportation cost.

The above approach ignores maintenance and repair costs and other costs associated with other segments in the resupply cycle. Expenditure
levels in these areas are assumed to be optimal in relation to the total expenditure level allocated for inventory investment and transportation. In principle, a methodology could be developed to include optimal allocation of resources to other segments of the resupply cycle.
SECTION 2
GENERAL MATHEMATICAL STATEMENT OF PROBLEM

Let $[N_B]$ be the expected number of backorders for a given single investment (recoverable) item. $[N_B]$ can be written

$$ [N_B] = [N_B(N, \rho)] ,$$

(2.1)

where

$$N = \text{total number of recoverable items including those installed},$$

$$\rho = \frac{\lambda}{\mu} = \lambda \tau, \quad \text{ratio of item demand rate to item resupply rate},$$

$$\lambda = \frac{1}{\tau \lambda}, \quad \text{item demand rate},$$

$$\tau \lambda = \text{mean time between demands},$$

$$\mu = \frac{1}{\tau}, \quad \text{resupply rate},$$

$$\tau \mu = \text{mean resupply time}.$$

The total cost associated with maintaining the inventory/resupply system can be written

$$C = C(N, \rho) .$$

(2.2)

Suppose it is desired to maintain a given "safe" backorder level so that

$$[N_B] = N_B^{(o)} ,$$

(2.3)

where $N_B^{(o)}$ is a specified constant, and the objective is to minimize system cost. Then the problem becomes

minimize

$$C = C(N, \rho)$$

subject to

$$[N_B(N, \rho)] = N_B^{(o)}$$

(2.4)
To facilitate discussion, assume \( N \) and \( \rho \) to be continuous variables; the generalization to the case where \( N \) is discrete can be achieved through usage of first difference operations for discrete variables. Thus,

\[
\delta C = \left( \frac{\partial C}{\partial N} \right)_\rho \delta N + \left( \frac{\partial C}{\partial \rho} \right)_N \delta \rho = 0 \quad ; \quad (2.5)
\]

\[
\delta [N_B] = \left( \frac{\partial [N_B]}{\partial N} \right)_\rho \delta N + \left( \frac{\partial [N_B]}{\partial \rho} \right)_N \delta \rho = 0 \quad ; \quad (2.6)
\]

\[
\delta N = - \left( \frac{\partial [N_B]}{\partial \rho} \right)_N \delta \rho \quad . \quad (2.7)
\]

For physical systems \( \left( \frac{\partial [N_B]}{\partial \rho} \right)_N > 0 \), \( \left( \frac{\partial [N_B]}{\partial N} \right)_\rho < 0 \), \( \left( \frac{\partial C}{\partial N} \right)_\rho > 0 \), and \( \left( \frac{\partial C}{\partial \rho} \right)_N < 0 \).

Note that

\[
\left( \frac{\partial N}{\partial \rho} \right) [N_B] = - \frac{\left( \frac{\partial [N_B]}{\partial \rho} \right)_N}{\left( \frac{\partial [N_B]}{\partial N} \right)_\rho} > 0 \quad , \quad (2.8)
\]

and

\[
\left( \frac{\partial N}{\partial \rho} \right)_C = - \frac{\left( \frac{\partial C}{\partial \rho} \right)_N}{\left( \frac{\partial C}{\partial N} \right)_\rho} > 0 \quad . \quad (2.9)
\]

2-2
Thus,  
\[ \delta N = + \left( \frac{\partial N}{\partial \rho} \right)_{N_B} \delta \rho \]  
(2.10)

and  
\[ \delta N = + \left( \frac{\partial N}{\partial \rho} \right)_{C} \delta \rho \]  
(2.11)

which implies
\[ f(N, \rho) = \left( \frac{\partial N}{\partial \rho} \right)_{C} - \left( \frac{\partial N}{\partial \rho} \right)_{N_B} = 0 \]  
(2.12)

If equation (2.12) and equation (2.4) are solved simultaneously, then solution values of \((N, \rho)\) are obtained which minimize system cost \(C(N, \rho)\). If, on the other hand, backorder level is to be minimized subject to a given total dollar investment level so that
\[ C(N, \rho) = C_o \]  
(2.13)

then equation (2.12) and equation (2.13) are solved simultaneously to obtain the desired values of \((N, \rho)\).

In order to determine the optimal (minimum) system expenditure level to be allocated between inventory investment level \((N)\) and resupply time level \((\tau = \rho \tau)\), the following functions must be determined:
\[ [N_B] > 0 \]  
(2.14)

\[ g_1(N, \rho) = \left( \frac{\partial [N_B]}{\partial N} \right)_{\rho} < 0 \]  
(2.15)

\[ g_2(N, \rho) = \left( \frac{\partial [N_B]}{\partial \rho} \right)_{N} > 0 \]  
(2.16)

2-3
The determination of relations (2.14) through (2.17) (STUDY NO. 1) is a study of the impact of inventory investment level \( N \) and resupply time level \( \tau_\mu = \rho_\lambda \tau_\mu \) upon backorder level. The determination of relations (2.18) through (2.21) (STUDY NO. 2) is a study of the impact of inventory investment level and resupply time level upon total system cost. The results of STUDY NO. 1 and STUDY NO. 2 can be used to determine optimum allocation of resources between inventory investment level and resupply time level as outlined above.

Note that the resupply time \( \tau_\mu \) is given by

\[
\tau_\mu = \text{(repair time)} + \text{(transportation time)} + \text{(administrative time for processing shipments)}.
\]

If the subject of interest is the impact of transportation time upon total system cost and required inventory investment level, then the cost function
$C(N, \rho)$ need only include costs associated with inventory investment level and transportation ship time level, where a differential change in
\[ \rho \left( \delta = \lambda \delta \tau \right) \] is considered to be due to a differential change in transportation ship time level.

The above problem formulation is generalized to include base level repair and depot level repair (which results in the inclusion of transportation time as part of the resupply time), the interaction of more than one base with a single depot, and can be modified to include the interactions due to movement of many investment and consumable items in the transportation pipelines among user bases and supply depots. If desired, the impact of administrative processing time and/or of repair time upon total system cost in relation to resulting required inventory investment level can also be determined using the conceptual procedure outlined above.
SECTION 3
DEMAND AND RESUPPLY PROCESSES IN INVENTORY MODELING

When dealing with failures of units in operation it is reasonable to assume that

1. The probability for a given unit in operation to fail in any particular operational time interval of arbitrary size is the same for all identical operational units;

2. The fact that a given unit in operation has failed in a given operational time interval does not affect the probability that other identical units in operation may fail in the same operational time interval (independence of failures);

3. The probability for a given unit in operation to fail during a given operational time interval is the same for all operational time intervals of equal arbitrary size (the mean operational time to failure is long compared to the total operational time period of observation) no matter when in time the failure takes place (time independence of failure rate).

The above conditions are necessary and sufficient to define the probability for failure of a given unit in operation during a given finite operational time interval \( t \). Thus for small \( \Delta t \), let

\[
\lambda \Delta t = \text{probability for failure of an item in operation during the operational time interval } \Delta t ,
\]

and

\[
1 - \lambda \Delta t = \text{probability of a given unit in operation for not failing during the operational time interval } \Delta t .
\]

If \( P(t) \) is the probability that a given unit in operation does not fail during a finite operational time interval \( t \), then

\[
P(t + \Delta t) = (1 - \lambda \Delta t) P(t) ,
\]
so that

\[
\frac{dP(t)}{dt} = \lim_{\Delta t \to 0} \frac{P(t + \Delta t) - P(t)}{\Delta t} = -\lambda P(t) ; \quad (3.4)
\]

\[
\frac{dP(t)}{dt} = -\lambda e^{-\lambda t} ; \quad (3.5)
\]

\[
P(t) = e^{-\lambda t} . \quad (3.6)
\]

The above results apply to a single unit in operation and are independent of the number of units in operation in the system. If, in addition, one requires that

4. The total number of units in operation and the total number of equal arbitrarily chosen operational time intervals are large (making statistical averages significant), then the failure process is Poisson. If the probability for observing \( m \) failures in an operational time \( t \) is given by \( P_m(t) \), then

\[
P_m(t + \Delta t) = P_m(t)(1 - \lambda \Delta t) + P_{m-1}(t)\lambda \Delta t , \quad \infty > m \geq 1 ; \quad (3.7)
\]

\[
P_0(t + \Delta t) = P_0(t)(1 - \lambda \Delta t) , \quad m = 0 . \quad (3.8)
\]

Thus when \( \Delta t \to 0 \), one obtains

\[
\frac{\partial P_0(t)}{\partial t} = -\lambda P_0(t) , \quad m = 0 ; \quad (3.9)
\]

\[
\frac{\partial P_m(t)}{\partial t} = -\lambda P_m(t) + \lambda P_{m-1}(t) , \quad \infty > m \geq 1 . \quad (3.10)
\]
Application of standard techniques obtains

\[ P_m(t) = \frac{(\lambda t)^m e^{-\lambda t}}{m!} , \quad 0 \leq m < \infty . \] (3.11)

It is observed that the Poisson process only applies if the number of units in operation is very large, or if as soon as a unit fails, it is immediately replaced (resupply time \( \equiv 0 \)) by an operational unit. If no backorders occur (i.e., there always exists sufficient servicable units in stock available for necessary immediate replacement of failed units), then the Poisson failure process applies; when backorders are allowed to exist for time intervals of significant duration, or if replacement time is significant, then the Poisson failure process does not apply. Thus while exponential operational failure times may be assumed, in general a Poisson failure process cannot be assumed without careful investigation into the nature of actual resupply times.

Assume that probability to complete a service on a given unit satisfies the following conditions.

1. The probability for a given unit in service to complete service in a particular time interval is the same for all identical units in service being served by identical servers;

2. The fact that a given unit in service has completed service in a given time interval does not affect the probability that other units in service in other servers may complete service in the same time interval (independence of service completion times);

3. The probability for a given unit in service to complete service during a given time interval is the same for all time intervals of equal size (the mean time to complete service is long compared to the total period of observation) no matter when the service completion takes place (time independence of service rate);

4. The number of units in service does not affect the probability for completion of service for an item in service.
The above conditions are necessary and sufficient to define the probability for completion of service for an item in service during an arbitrary given time interval $t$. Thus, for small $\Delta t$ let

$$\mu \Delta t = \text{probability for completing service for an item in service during } \Delta t,$$

$$1 - \mu \Delta t = \text{probability for not completing service for an item in service during } \Delta t.$$  \hfill (3.12)  \hfill (3.13)

An argument similar to that given above for failures obtains exponentially distributed service times. (Note that if the number of units awaiting service for a given single server were always non-zero, then the number completing service in an arbitrary time interval would be Poisson distributed.)

In applying servers with exponential service times in modeling the resupply process, the nature of the server configuration is extremely important. A few possible models for resupply are given below.

If a single level resupply system function is assumed, then resupply can be described by a singly indexed resupply state so that

$$\mu_n = n\mu, \quad 0 \leq n \leq n_{\text{max}},$$

$$\mu_n = n_{\text{max}}\mu, \quad N \geq n \geq n_{\text{max}},$$

$$1 \leq n_{\text{max}} \leq N,$$  \hfill (3.14)  \hfill (3.15)  \hfill (3.16)

where

$$\mu_n = \text{resupply rate when } n \text{ units are in resupply},$$

$$\mu = \text{constant = resupply rate when a single unit is in resupply},$$

$$n_{\text{max}} = \text{number of identical parallel servers assumed},$$

$$N = \text{total number of units in the system (including installed, stock, and resupply items)}.$$
The single index for resupply is \( n \), the number of units in resupply.

Note that when \( n_{\text{max}} = 1 \), then a single server model is assumed, and when \( n_{\text{max}} > 1 \), then a multiple server model is assumed. If \( n_{\text{max}} \) is chosen so that \( n_{\text{max}} = N \), then no service queue is formed; if \( n_{\text{max}} < N \), then service queues do form.

For a dual level resupply system (two level resupply; e.g., base level and depot level) two server models may be postulated. The first is described by a resupply state with a single index so that

\[
\mu_n = n\mu, \quad 0 \leq n \leq n_{\text{max}}, \quad (3.17)
\]

\[
\mu_n = n_{\text{max}}\mu, \quad n_{\text{max}} \leq n \leq N, \quad (3.18)
\]

\[
1 \leq n_{\text{max}} \leq N \quad (3.19)
\]

\[
\mu = f_1\mu_1 + f_2\mu_2 \quad (3.20)
\]

where

\[
f_1 = \text{probability for level 1 resupply (e.g., base level resupply)},
\]

\[
f_2 = \text{probability for level 2 resupply (e.g., depot level resupply)},
\]

\[
f_1 + f_2 = 1,
\]

\[
\mu_1 = \text{resupply rate for one unit in level 1 resupply},
\]

\[
\mu_2 = \text{resupply rate for one unit in level 2 resupply}.
\]

Again the single index for resupply is \( n \), the total number of units in resupply. The form for the above dual level resupply is identical to that for single level resupply except that the effective service rate \( \mu \) is given in terms of a weighted combination of \( \mu_1 \) and \( \mu_2 \).

A second approach to model a dual level resupply system is to use a two index resupply state. The description of resupply rates is given by

\[
\mu^{(1)}_a = a\mu_1, \quad 0 \leq a \leq a_{\text{max}}, \quad (3.21)
\]

\[
\mu^{(1)}_a = a_{\text{max}}\mu_1, \quad a_{\text{max}} \leq a \leq N.
\]
\[ \mu^{(2)}_\beta = \beta \mu_2 , \quad 0 \leq \beta \leq \beta_{\text{max}}, \]

\[ \mu^{(2)}_\beta = \beta_{\text{max}} \mu_2 , \quad \beta_{\text{max}} \leq \beta \leq N ; \quad \text{(3.21)} \]

\[ \mu = f_1 \mu_1 + f_2 \mu_2 , \quad f_1 + f_2 = 1 ; \]

where

\[ \mu^{(1)} = \text{resupply rate for level 1 resupply when } \alpha \text{ units are in resupply at level 1,} \]

\[ \mu^{(2)} = \text{resupply rate for level 2 resupply when } \beta \text{ units are in resupply at level 2,} \]

\[ \alpha = \text{number of units in resupply at level 1,} \]

\[ \beta = \text{number of units in resupply at level 2,} \]

\[ \alpha + \beta = n = \text{total number of units in resupply, } 0 \leq \alpha + \beta = n \leq N, \]

\[ f_1 = \text{probability for level 1 resupply,} \]

\[ f_2 = \text{probability for level 2 resupply,} \]

\[ \alpha_{\text{max}} = \text{assigned number of identical parallel servers at level 1,} \]

\[ \beta_{\text{max}} = \text{assigned number of identical parallel servers at level 2,} \]

\[ \mu_1 = \text{resupply rate at level 1 when one unit is in resupply at level 1,} \]

\[ \mu_2 = \text{resupply rate at level 2 when one unit is in resupply at level 2,} \]

\[ N = \text{total number of units in the system (including installed items as well as items in resupply and in stock).} \]

The two index resupply state is labeled by \((\alpha, \beta)\), the number of resupply items at level 1 and the number of resupply items at level 2, respectively.

In the single index resupply state models given above, the arrival rate for resupply (failure rate) can be modeled by

\[ \lambda_m = m_0 \lambda , \quad m_0 \leq m \leq N ; \quad \text{(3.22)} \]
\[ \lambda_m = m \lambda, \quad 0 \leq m \leq m_0; \quad (3.23) \]

or equivalently by

\[ \lambda_m = m_0 \lambda, \quad 0 \leq n \leq N - m_0; \quad (3.24) \]
\[ \lambda_m = (N - n) \lambda, \quad N - m_0 + 1 \leq n \leq N; \quad (3.25) \]

where

\[ n = \text{number of units in resupply}, \]
\[ \lambda_m = \text{resupply arrival rate when there are } m \text{ units installed}, \]
\[ \lambda = \text{resupply arrival rate when there is one unit installed}, \]
\[ m_0 = \text{minimum number of installed items resulting in meeting total operational requirements with installed units operating at full capability (when } m > m_0, \text{ then total operational requirements are met with installed items operating at less than full capability; when } m < m_0 \text{ total operational requirements are not met with installed items operating at full capability)}, \]
\[ N = \text{total number of units in system}. \]

In the two index resupply state model, the arrival process can be described by

\[ \lambda_m^{(1)} = f_1 m_0 \lambda, \quad m_0 \leq m \leq N, \]
\[ \lambda_m^{(2)} = f_2 m_0 \lambda, \quad m_0 \leq m \leq N, \]
\[ \lambda_m^{(1)} = f_1 m \lambda, \quad 0 \leq m \leq m_0, \quad (3.26) \]
\[ \lambda_m^{(2)} = f_2 m \lambda, \quad 0 \leq m \leq m_0, \]
\[ f_1 + f_2 = 1 \]
where

\[ \lambda = \text{failure rate when one unit is installed}, \]
\[ \lambda^{(1)}_m = \text{resupply arrival rate for level 1 when } m \text{ units are installed}, \]
\[ \lambda^{(2)}_m = \text{resupply arrival rate for level 2 when } m \text{ units are installed}, \]
\[ f_1 = \text{probability for level 1 resupply}, \]
\[ f_2 = \text{probability for level 2 resupply}, \]
\[ m = \text{number of installed units}, \]
\[ m_0 = \text{minimum number of installed units resulting in meeting total operational requirements}. \]

Define \( \eta^{(1)}_n \) to be the resupply arrival rate at level 1 when the total number in resupply is \( n \) and \( \eta^{(2)}_n \) to be the resupply arrival rate at level 2 when the total number in resupply is \( n \), \( 0 \leq n = \alpha + \beta \leq N \). Then

\[ \eta^{(1)}_n = f_1 m_0 \lambda, \quad 0 \leq n \leq N - m_0; \quad (3.27) \]
\[ \eta^{(2)}_n = f_2 m_0 \lambda, \quad 0 \leq n \leq N - m_0; \quad (3.28) \]
\[ \eta^{(1)}_n = f_1 (N - n) \lambda, \quad N - m_0 + 1 \leq n \leq N; \quad (3.29) \]
\[ \eta^{(2)}_n = f_2 (N - n) \lambda, \quad N - m_0 + 1 \leq n \leq N. \quad (3.30) \]

The state transition probabilities for a singly indexed resupply state are given for \( n_{\text{max}} = N \) and \( \mu_n = n \mu \), \( 0 \leq n \leq N \) (to first order in \( \Delta t \)) by

\[ P_0(t + \Delta t) = P_0(t) [1 - m_0 \lambda \Delta t] + P_1(t) [\mu \Delta t], \quad n = 0; \quad (3.31) \]
\[ P_n(t + \Delta t) = P_n(t)[1 - m_0 \lambda \Delta t - \mu \Delta t] + P_{n+1}(t)[(n + 1)\mu \Delta t] + P_{n-1}(t)[m_0 \lambda \Delta t], \]
\[ 0 < n \leq N - m_0; \quad (3.32) \]

\[ P_n(t + \Delta t) = P_n(t)[1 - (N - n)\lambda \Delta t - \mu \Delta t] + P_{n+1}(t)[(n + 1)\mu \Delta t] + P_{n-1}(t)[(N - n + 1)\lambda \Delta t], \]
\[ N - m_0 + 1 \leq n < N; \quad (3.33) \]

\[ P_N(t + \Delta t) = P_N(t)[1 - \nu \lambda \Delta t] + P_{N-1}(t)[\lambda \Delta t], \quad n = N. \quad (3.34) \]

The differential equations which result when \( \Delta t \to 0 \) and the corresponding steady-state solution are given in the section entitled "Conservative Parallel Server Inventory Model."

When \( \mu_n = \mu \) for \( n \geq 1 \) and \( \mu_n = 0 \) for \( n = 0 \), the state transition probabilities for a singly indexed resupply state are given (to first order in \( \Delta t \)) by

\[ P_0(t + \Delta t) = P_0(t)[1 - m_0 \lambda \Delta t] + P_1(t)[\mu \Delta t], \quad n = 0; \quad (3.35) \]

\[ P_n(t + \Delta t) = P_n(t)[1 - m_0 \lambda \Delta t - \mu \Delta t] + P_{n+1}(t)[\mu \Delta t] + P_{n-1}(t)[m_0 \lambda \Delta t], \]
\[ 0 < n \leq N - m_0; \quad (3.36) \]

\[ P_n(t + \Delta t) = P_n(t)[1 - (N - n)\lambda \Delta t - \mu \Delta t] + P_{n+1}(t)[\mu \Delta t] + P_{n-1}(t)[(N - n + 1)\lambda \Delta t], \]
\[ N - m_0 + 1 \leq n < N; \quad (3.37) \]

3-9
\[ P_N(t + \Delta t) = P_N(t)[1 - \mu \Delta t] + P_{N-1}(t)[\lambda \Delta t] , \quad n = N . \quad (3.38) \]

The differential equations which result when \( \Delta t \to 0 \) and the corresponding steady-state solution are given in the section entitled "Conservative Single Server Inventory Model."

The two index resupply state model is described (to first order in \( \Delta t \)) by

\[ P_{\alpha,\beta}(t + \Delta t) = P_{\alpha,\beta}(t)[1 - (1) \Delta t - \eta_{\alpha+\beta} \Delta t - \mu_{\alpha} \Delta t - \mu_{\beta} \Delta t] \]

\[ + P_{\alpha+1,\beta}(t)[\mu_{\alpha+1}] \Delta t] + P_{\alpha-1,\beta}(t)[\eta_{\alpha+\beta-1}] \Delta t] \]

\[ + P_{\alpha,\beta+1}(t)[\mu_{\beta+1}] \Delta t] + P_{\alpha,\beta-1}(t)[\eta_{\alpha+\beta-1}] \Delta t] , \quad (3.39) \]

where

\[ P_{\alpha,\beta}(t) \equiv 0 \text{ when } \alpha < 0 \text{ or when } \beta < 0 . \]

The differential equations obtained when \( \Delta t \to 0 \) are

\[ \frac{\partial P_{\alpha,\beta}(t)}{\partial t} = - \left[ (\eta_{\alpha+\beta} + (2)) + (\mu_{\alpha} + (2)) \right] P_{\alpha,\beta}(t) + (1) P_{\alpha+1,\beta}(t) \]

\[ + (2) P_{\alpha,\beta+1}(t) + (1) P_{\alpha-1,\beta}(t) + (2) P_{\alpha+\beta-1,\beta}(t) , \quad (3.40) \]

\[ 0 \leq \alpha + \beta = n \leq N . \]
The two index resupply state system described above is one that most closely describes the two-echelon inventory system currently adopted by Air Force management [88, 89, 104] and is of major usefulness in the trade-off analysis described earlier in this document. Below is diagrammed the conservative two level parallel server inventory model, and a mathematical treatment showing the equivalence of this model to the singly indexed parallel server model follows.

CONSERVATIVE TWO LEVEL PARALLEL SERVER INVENTORY MODEL

Define

$$p_n(t) = \sum_{\alpha=0}^{n} p_{\alpha, n-\alpha}(t),$$  \hspace{1cm} (3.41)

and observe that

$$p_{\alpha, n-\alpha}(t) = \frac{1^{\alpha} n^{n-\alpha}}{\alpha! n!} p_n(t), \hspace{1cm} 0 \leq \alpha \leq n, \hspace{0.5cm} 0 \leq n \leq N,$$  \hspace{1cm} (3.42)
where \( P_n^{(t)} \) = the probability that there are \( n \) units in resupply at time \( t \),

\[
\begin{align*}
\mu_\alpha^{(1)} &= \alpha \mu_1 , \quad 0 \leq \alpha \leq n , \quad 0 \leq n \leq N ,  \\
\mu_\alpha^{(2)} &= (n-\alpha) \mu_2 , \quad 0 \leq \alpha \leq n , \quad 0 \leq n \leq N ,
\end{align*}
\]

(3.45)

where \( \lambda_n \) = resupply arrival rate when a total of \( n \) items are in resupply.

Also note that

\[
\begin{align*}
\eta_0^{(1)} + \eta_0^{(2)} &= m_0 \lambda \equiv \lambda_n , \quad 0 \leq n \leq N-m_0 ,  \\
\eta_0^{(1)} + \eta_0^{(2)} &= (N-n) \lambda \equiv \lambda_n , \quad N-m_0 + 1 \leq n \leq N .
\end{align*}
\]

(3.43)

Therefore,

\[
\begin{align*}
\eta_0^{(1)} + \eta_0^{(2)} &\equiv \lambda_n , \quad 0 \leq n \leq N ,
\end{align*}
\]

(3.44)

where \( \lambda_n \) = resupply arrival rate when a total of \( n \) items are in resupply.

Also note that

\[
\begin{align*}
\mu_\alpha^{(1)} &= \alpha \mu_1 , \quad 0 \leq \alpha \leq n , \quad 0 \leq n \leq N ,  \\
\mu_\alpha^{(2)} &= (n-\alpha) \mu_2 , \quad 0 \leq \alpha \leq n , \quad 0 \leq n \leq N ,
\end{align*}
\]

(3.45)

when \( \alpha_{\max} = N = \beta_{\max} \) (infinite number of parallel servers in resupply at level 1 and at level 2). Observe that

\[
\begin{align*}
\mu_{\alpha+1, \alpha+1, n-\alpha}^{(1)} &= (\alpha+1) \mu_1 f_1^{\alpha+1} f_2^{n-\alpha} p^{n+1} (t) ,  \\
&= (n+1) \mu_1 f_1^{\alpha} f_2^{n-\alpha} p^{n+1} (t) ;
\end{align*}
\]

(3.46)
\[
\sum_{\alpha=0}^{n} \mu^{(1)}_{\alpha+1, n-\alpha}(t) = (n+1)\mu f_{1} P_{n+1}(t) \quad . \tag{3.47}
\]

Similarly,

\[
\sum_{\alpha=0}^{n} \mu^{(2)}_{n-\alpha+1, \alpha, n-\alpha+1}(t) = (n-\alpha+1)\mu_{2} f_{\alpha}^{n-\alpha+1} P_{n+1}(t) \quad . \tag{3.48}
\]

and

\[
\sum_{\alpha=0}^{n} \mu^{(2)}_{\alpha, n-\alpha+1}(t) = (n+1)\mu_{2} P_{n+1}(t) \quad ; \tag{3.49}
\]

\[
\mu_{\alpha}^{(1)} p_{\alpha, \beta}(t) = a\mu_{1} p_{\alpha, n-\alpha}(t) = a\mu_{1} f_{1}^{n-\alpha}(\alpha) P_{n}(t) \quad , \tag{3.50}
\]

\[
= \mu_{1} f_{1}^{n-\alpha} P_{n}(t) \quad ; \tag{3.51}
\]

\[
\mu^{(2)}_{\alpha, n-\alpha}(t) = (n-\alpha)\mu_{2} f_{2}^{n-\alpha}(\alpha) P_{n}(t) \quad ; \tag{3.52}
\]

\[
= n\mu_{2} f_{2}^{n-\alpha} P_{n}(t) \quad ; \tag{3.53}
\]

\[ \text{3-13} \]
Continuing,

\[
\sum_{\alpha=0}^{n} \eta_{n-1, \alpha-1, n-\alpha}^{(1)}(t) = \eta_{n-1, n-1}^{(1)}(t),
\] (3.54)

and

\[
\sum_{\alpha=0}^{n} \eta_{n-1, \alpha, n-\alpha-1}^{(2)}(t) = \eta_{n-1, n-1}^{(2)}(t),
\] (3.55)

so that

\[
\sum_{\alpha=0}^{n} \left[ \eta_{n-1, \alpha-1, n-\alpha}^{(1)}(t) + \eta_{n-1, \alpha, n-\alpha-1}^{(2)}(t) \right] = \lambda_{n-1, n-1}(t).
\] (3.56)

Therefore, summing equation (3.40) over \( \alpha \) with \( \beta = n-\alpha \) obtains

\[
\frac{\partial P_n(t)}{\partial t} = -\lambda_n P_n(t) - n\mu P_n(t) + (n+1)\mu P_{n+1}(t) + \lambda_{n-1, n-1}(t),
\] (3.57)

\[0 \leq n \leq N,\]

where \( P_n(t) = 0 \) for \( n < 0 \) and for \( n > N \).

Thus, the two index resupply state model described by equation (3.40) reduces to the one index resupply state model given by equation (3.57), and is represented by the model described in the section entitled "Conservative Parallel Server Inventory Model" and by equations (3.31) through (3.34) in this section.
SECTION 4
CONSERVATIVE PARALLEL SERVER INVENTORY MODEL

Consider the system for resupply of recoverable items that is diagrammed below.

\[ \text{Resupply (n)} \rightarrow \text{Stock (s)} \]

\[ \text{Installed (m)} \]

The parameters are given as follows:

\[ N = \text{total number of items in the system,} \]
\[ n = \text{number of items in resupply,} \]
\[ m = \text{number of items installed,} \]
\[ s = \text{number of items in stock,} \]
\[ \lambda = \text{failure rate for a single item while installed,} \]
\[ \lambda_m = \text{failure rate of installed items when number of items installed is } m, \]
\[ m_0 = \text{minimum number of serviceable installed items required to accomplish mission objectives (} m \geq m_0 \text{ implies that service performance level is being met; } m < m_0 \text{ implies that the installed serviceable items are working at full capacity and are not fully meeting mission requirements),} \]
\[ m_1 = \text{minimum number of installed serviceable items resulting in} \]

\[4-1\]
no backorder \((m < m_1)\) implies that a backorder exists, \(N \geq m \geq m_0 \geq 1\),

\[ \mu = \frac{1}{\tau} \text{ resupply rate for a single item in resupply}, \]

\[ \tau = \text{ mean resupply time for one item in resupply}, \]

\[ p = \lambda \tau \mu = \text{ratio of failure rate to resupply rate for one item in resupply}, \]

\[ \mu_n = \text{resupply rate when } n \text{ items are in resupply}. \]

For a conservative system observe that

\[ N = m + s + n , \quad (4.1) \]

and require that

\[ \lambda_m = m_0 \lambda , \quad 0 \leq n \leq N - m_0 ; \quad (4.2) \]

\[ \lambda_m = (N - n) \lambda , \quad N - m_0 + 1 \leq n \leq N ; \quad (4.3) \]

\[ \mu_n = n \mu , \quad 0 \leq n \leq N . \quad (4.4) \]

Thus, it is assumed that there are \(N\) parallel servers in resupply. The state transition probabilities are given by

\[ \frac{\partial P_0^{(N)}(t)}{\partial t} = -m_0 \lambda P_0(t) + \mu P_1(t) , \quad n = 0 ; \quad (4.5) \]

\[ \frac{\partial P_n^{(N)}(t)}{\partial t} = -m_0 \lambda P_n(t) + (n + 1) \mu P_{n+1}(t) - n \mu P_n(t) - m_0 \lambda P_{n-1}(t) , \quad 0 < n \leq N - m_0 ; \quad (4.6) \]
\[
\frac{\partial P_n^{(N)}(t)}{\partial t} = -(N-n)\lambda P_n^{(N)}(t) + (n+1)\mu P_{n+1}^{(N)}(t) - n\mu P_n^{(N)}(t) + (N-n+1)\lambda P_{n-1}^{(N)}(t),
\]

\[\text{for } N - m_0 + 1 \leq n \leq N;\]

\[
\frac{\partial P_N^{(N)}(t)}{\partial t} = -N\mu P_N^{(N)}(t) + \lambda P_{N-1}^{(N)}(t), \quad n = N.
\]

In the above, \( P_n^{(N)}(t) \) is the probability that \( n \) items are in resupply at time \( t \). Using standard techniques, one obtains the following steady-state solution \( (t = \infty) \).

\[
P_n^{(N)} = \frac{(m_0 \rho)^n}{n!} P_0^{(N)}, \quad 0 \leq n \leq N - m_0;
\]

\[
P_n^{(N)} = \frac{(m_0 \rho)^{N-m_0}}{n!(N-n)!} p_{n-(N-m_0)} P_0^{(N)}, \quad N - m_0 + 1 \leq n \leq N;
\]

\[
P_0^{(N)} = \left\{ \sum_{n=m_0}^{N} \frac{(m_0 \rho)^n}{n!} + (m_0 \rho)^{N-m_0} \sum_{n=N-m_0+1}^{N} \frac{\rho}{n!(N-n)!} \right\}^{-1}.
\]

Equation (4.9) and equation (4.10) can be written

\[
P_{N-x}^{(N)} = \frac{(m_0 \rho)^{N-x}}{(N-x)!} P_0^{(N)}, \quad m_0 \leq x \leq N,
\]

\[
P_{N-x}^{(N)} = (m_0 \rho)^{N} \left( m_0! m_0^{!} \right) \frac{\rho^{-x}}{(N-x)! x!} P_0^{(N)}, \quad 0 \leq x \leq m_0 - 1.
\]
and equation (4.11) can be written

\[
\begin{align*}
P_0^{(N)} &= \left\{ \sum_{n=0}^{N-m_0} \frac{(m_0 \rho)^n}{n!} + (m_0 \rho)^{N-m_0} \frac{(-\lambda)^x}{x!(N-x)!} \right\}^{-1}.
\end{align*}
\]

(4.14)

The resupply states corresponding to \( 0 \leq n \leq N - m_0 \) (states having full capability to fulfill mission objectives) would be described by a Poisson distribution if \( P_0^{(N)} \) were given by

\[
P_0^{(N)} = e^{-m_0 \rho} = e^{-m_0 \lambda \tau}.
\]

(4.15)

However, the Poisson process applies only if \( N - m_0 = 0 \), a condition which implies that the probability to fail to meet mission objectives is zero. Also observe that the resupply states for \( N \geq n \geq N - m_0 + 1 \) (those states having less than full capability to meet mission objectives) are not described by a Poisson distribution, resulting from the fact that \( N - m_0 \) is finite.

As observed above, the number of backorders is zero for \( N \geq x \geq m_1 \) \((0 \leq n \leq N - m_1)\), and the number of backorders is \( m_1 - x = m_1 - (N - n) \) for \( 0 \leq x \leq m_1 - 1 \) \((N - m_1 + 1 \leq n \leq N)\). Thus, the expected number of backorders \([N_B^{(N)}]\) is given by

\[
[N_B^{(N)}] = \sum_{x=0}^{m_1-1} (m_1 - x) P_0^{(N)}_{N-x}.
\]

(4.16)

\[
[N_B^{(N)}] = (m_1 \rho)_P^{(N)} \left\{ \frac{(-\lambda)^{m_1-1}}{(m_1-x)} \frac{(-\lambda)^x}{x!(N-x)!} \right\}.
\]

(4.17)
Inspection of equation (4.12) and equation (4.13) obtains

\[
\frac{p^{(N)}_{N-x}}{p^{(N-1)}_{N-1-x}} = \frac{m_0 \rho}{N-x} \frac{p^{(N)}_0}{p^{(N-1)}_0}, \quad 0 \leq x \leq N - 1; \quad (4.18)
\]

\[
\frac{p^{(N)}_n}{p^{(N-1)}_{n-1}} = \frac{m_0 \rho}{n} \frac{p^{(N)}_0}{p^{(N-1)}_0}, \quad 1 \leq n \leq N. \quad (4.19)
\]

The expected number of units in resupply is given by

\[
[n^{(N)}] = \sum_{n=1}^{N} n p^{(N)}_n, \quad (4.20)
\]

so that

\[
[n^{(N)}] = m_0 \rho \frac{p^{(N)}_0}{p^{(N-1)}_0}, \quad (4.21)
\]

or

\[
\frac{p^{(N)}_{N-x}}{p^{(N-1)}_{N-1-x}} = \frac{[n^{(N)}]}{N-x}. \quad (4.22)
\]

Now, the probability for stockout is given by

\[
p^{(N)}_s = \sum_{x=0}^{m_1-1} p^{(N)}_{N-x}, \quad (4.23)
\]
and therefore

\[ [N_B^{(N)}] = [n^{(N)}] \frac{p_s^{(N-1)}}{\rho} - (N-m) \frac{p_s^{(N)}}{\rho} \quad . \]  (4.24)

Thus,

\[ \frac{\partial [N_B^{(N)}]}{\partial \rho} = \left[ n^{(N)} \right] \rho \left\{ [N_B^{(N-1)}] - [N_B^{(N)}] \right\} . \]  (4.25)

Define

\[ \Delta [N_B^{(N)}] \equiv [N_B^{(N+1)}] - [N_B^{(N)}] \]  (4.26)

and

\[ \Delta p_s^{(N)} \equiv p_s^{(N+1)} - p_s^{(N)} \]  (4.27)

to obtain

\[ \frac{\partial [N_B^{(N)}]}{\partial \rho} = - \left[ n^{(N)} \right] \rho \Delta [N_B^{(N-1)}] , \]  (4.28)

and

\[ \frac{\partial p_s^{(N)}}{\partial \rho} = - \left[ n^{(N)} \right] \rho \Delta p_s^{(N-1)} . \]  (4.29)

Also observe that equation (4.12) and equation (4.13) yield

\[ \frac{1}{p_s^{(N)}} \left( \frac{\partial p_s^{(N)}}{\partial \rho} \right)_{N-x} = \frac{N-x}{\rho} + \frac{1}{p_s^{(N)}} \left( \frac{\partial p_s^{(N)}}{\partial \rho} \right)_{N-x} , \quad 0 \leq x \leq N ; \]  (4.30)

\[ \frac{1}{p_s^{(N)}} \left( \frac{\partial p_s^{(N)}}{\partial \rho} \right)_{N-x} = - \left[ n^{(N)} \right] \rho . \]  (4.31)
It is observed that when \( m_0 = 1 \), then \( \lambda_m = \lambda \) for all \( m > 0 \), corresponding to the situation where the failure rate is constant and independent of the number of installed serviceable units (except for \( m = 0 \)). Thus,

\[
P^n_{m_0} = \frac{\rho^n}{n!} p_0(N), \quad m_0 = 1, \quad 0 \leq n \leq N; \quad (4.32)
\]

\[
p_0(N) = \left\{ \sum_{n=0}^{N} \frac{\rho^n}{n!} \right\}^{-1}, \quad m_0 = 1 \quad ; \quad (4.33)
\]

and

\[
[N_B(N)] = \sum_{x=0}^{N} (m_1 - x) \frac{\rho^{N-x}}{(N-x)!} p_0(N), \quad m_0 = 1. \quad (4.34)
\]

Note that

\[
\lim_{N \to \infty} p_0(N) = e^{-\rho} = e^{-\lambda T}, \quad (4.35)
\]

\[
\lim_{N \to \infty} p(N) = \frac{\rho^n}{n!} e^{-\lambda T}, \quad (4.36)
\]

and

\[
\lim_{N \to \infty} [N_B(N)] = \lim_{N \to \infty} \sum_{x=0}^{m_1-1} (m_1 - x) \frac{\rho^{N-x}}{(N-x)!} p_0(N) = 0 \quad \text{for} \quad m_0 = 1 \quad \text{and} \quad \rho < 1. \quad (4.37)
\]

Thus, a Poisson demand process occurs for large \( N \), and for large \( N \) and \( \rho < 1 \) the expected number of backorders is zero.

When \( m_0 = m_1 = 1 \), then

\[
[N_B(N)] = \frac{\rho^N}{N!} p_0(N) = p(N), \quad m_0 = m_1 = 1; \quad (4.38)
\]
\[ [\alpha^{(N)}] = \rho \left\{ 1 - \frac{\rho^N p_0^{(N)}}{N!} \right\} \quad ; \quad (4.39) \]

\[ \left( \frac{\partial [N_B^{(N)}]}{\partial \rho} \right)_N = \frac{\rho^{N-1} p_0^{(N)}}{N!} \left\{ N - \rho + \frac{\rho^{N+1} p_0^{(N)}}{N!} \right\} \quad ; \quad (4.40) \]

\[ [N_B^{(N+1)}] - [N_B^{(N)}] = -\frac{\rho^N p_0^{(N)}}{N!} \left\{ 1 - \left( \frac{\rho}{N + 1} \right) \frac{p_0^{(N+1)}}{p_0^{(N)}} \right\} \quad . \quad (4.41) \]

If \( N = m_0 = m_1 = 1 \), then

\[ p_0^{(1)} = \frac{1}{1 + \rho} \quad , \quad (4.42) \]

\[ [N_B^{(1)}] = [\alpha^{(1)}] = \frac{\rho}{1 + \rho} \quad ; \quad (4.43) \]

\[ \left( \frac{\partial [N_B^{(1)}]}{\partial \rho} \right)_N = \frac{1}{(1 + \rho)^2} \quad ; \quad (4.44) \]

\[ [N_B^{(2)}] - [N_B^{(1)}] = -\frac{\rho (2 + \rho)}{(1 + \rho)(2 + 2\rho + \rho^2)} \quad . \quad (4.45) \]
When $N = a_0 = a_1 = \rho = 1$, then

\[
\left( \frac{\partial [N_B^{(1)}]}{\partial \rho} \right)_N = \frac{1}{4} ; \tag{4.46}
\]

\[
[N_B^{(2)}] - [N_B^{(1)}] = -\frac{3}{10} ; \tag{4.47}
\]

\[
[N_B^{(1)}] = \frac{1}{2} . \tag{4.48}
\]

A case of particular interest occurs when $N = m_0 = m_1$. This is the case when there are no extra spares and when one or more units in re-supply results in system performance degradation. When $N = m_0 = m_1$, then

\[
p_0^{(N)} = \frac{1}{(1 + \rho)^N} , \quad m_0 = N ; \tag{4.49}
\]

\[
p_n^{(N)} = \frac{\rho^n N!}{(1 + \rho)^N n! (N - n)!} , \quad m_0 = N ; \tag{4.50}
\]

\[
[N_B^{(N)}] = \left[ n^{(N)} \right] = \frac{N\rho}{1 + \rho} , \quad m_1 = m_0 = N ; \tag{4.51}
\]

When there is one extra spare, then

\[
p_0^{(N+1)} = \frac{N + 1}{1 + N(1 + \rho)^{N+1}} , \quad m_0 = N ; \tag{4.52}
\]
\[ p_n^{(N+1)} = \frac{N(N+1)!}{1 + N(1 + \rho)^{N+1}} \cdot \frac{\rho^n}{n!(N + 1 - n)!}, \quad 1 \leq n \leq N + 1, \quad m_0 = N; \quad (4.53) \]

\[ \left[ N_B^{(N+1)} \right] = \frac{N\left\{1 - (N - \rho)(1 + \rho)^N\right\}}{1 + N(1 + \rho)^N}, \quad m_1 = m_0 = N; \quad (4.54) \]

\[ \left( \frac{\partial [N_B^{(N)}]}{\partial \rho} \right)_N = \frac{N}{(1 + \rho)^2}, \quad m_1 = m_0 = N; \quad (4.55) \]

\[ \left[ N_B^{(N)} \right] - \left[ N_B^{(N+1)} \right] = N \left\{ \frac{(1 + \rho)^{N+1} - 1}{(1 + \rho)[1 + N(1 + \rho)^{N+1}]} \right\}, \quad N = m_0 = m_1. \quad (4.56) \]

Thus,

\[ \delta [N_B^{(N)}] = \frac{N}{(1 + \rho)^2} \quad \text{for } N = \text{constant}, \quad N = m_0 = m_1; \quad (4.57) \]

\[ \delta [N_B^{(N)}] = -N \left\{ \frac{(1 + \rho)^{N+1} - 1}{(1 + \rho)[1 + N(1 + \rho)^{N+1}]} \right\} \delta N, \quad \delta N = 1, \quad \text{for } \rho = \text{constant}, \quad N = m_0 = m_1. \quad (4.58) \]
In the Poisson Model the demand process described in the section entitled "Conservative Parallel Server Inventory Model" is replaced by a Poisson process as was done by Sherbrooke [104] in the development of "Metric: A Multi-Echelon Technique for Recoverable Item Control" and by Muckstadt [88, 89] in the development of "Mod-Metric: A Multi-Item, Multi-Echelon, Multi-Indenture Inventory Model." The following formulae apply when this approach is taken.

\[ P_n = e^{-\lambda \rho} \frac{(\lambda \rho)^n}{n!}, \quad 0 \leq n < \infty; \quad (5.1) \]

\[ \left[ N_B^{(N)} \right] = \sum_{n=N-m_1+1}^{\infty} (n-N+m_1) P_n; \quad (5.2) \]

\[ \left[ N_B^{(N+1)} \right] - \left[ N_B^{(N)} \right] \equiv \Delta \left[ N_B^{(N)} \right] = -p_s^{(N)}; \quad (5.3) \]

\[ \frac{\partial \left[ N_B^{(N)} \right]}{\partial \rho} = m_0 p_s^{(N-1)}; \quad (5.4) \]

\[ \frac{\partial p_s^{(N)}}{\partial \rho} = m_0 p_{N-m_1+1}; \quad (5.5) \]

\[ \Delta p_s^{(N)} = p_s^{(N+1)} - p_s^{(N)} = -p_{N-m_1+1}; \quad (5.6) \]

\[ \left[ N_B^{(N)} \right] = [n^{(N)}] p_s^{(N-1)} - (N-m_1) p_s^{(N)}; \quad (5.7) \]

SECTION 5
POISSON MODEL
\[ \begin{align*}
\left[ n^{(N)} \right] &= \left[ n^{(N+1)} \right] = m_0 \rho ; \\
\frac{P_{n+1}^{(N+1)}}{P_n^{(N)}} &= \frac{n^{(N+1)}}{n+1} ;
\end{align*} \tag{5.8}\]

\[ \frac{P_{0}^{(N+1)}}{P_{0}^{(N)}} = 1 ; \tag{5.9} \]

\[ P_{0}^{(N)} = e^{-m_0 \rho} ; \tag{5.10} \]

\[ \frac{1}{P_n^{(N)}} \frac{\partial P^{(N)}}{\partial \rho} = \frac{n}{\rho} - \frac{\left[ n^{(N)} \right] }{\rho} ; \tag{5.11} \]

\[ \frac{\partial [N^{(N)}_B] }{\partial \rho} = - \frac{\left[ n^{(N)} \right] }{\rho} \Delta [N^{(N-1)}_B] ; \tag{5.12} \]

\[ \frac{\partial P^{(N)}}{\partial \rho} = - \frac{\left[ n^{(N)} \right] }{\rho} \Delta p^{(N-1)}_s . \tag{5.13} \]

Note that each of the equations whose identifying numbers are starred (*) has a form identical to that obtained for the finite population model (parallel server inventory model). The fundamental differences between the two models arise from the fact that one model assumes a finite population, while the other assumes an infinite population. In particular, note that the \( P_n \) for the Poisson Model corresponding to those states where total mission requirements cannot be fulfilled \( (n > N - m_0 + 1) \) have derivatives and magnitudes which are very different from those given by
the Parallel Server Model. Also, the Poisson Model defines an infinite number of states associated with the backorder condition, while the Parallel Server Model defines a finite number of states ($m_1$ in number).
SECTION 6
CONSERVATIVE SINGLE SERVER INVENTORY MODEL

The queueing model for recoverable items diagrammed below has been investigated, and the steady-state solution has been obtained.

The parameters for the system diagrammed above are given as follows:

\[ \begin{align*}
    N &= \text{total number of items in the system}, \\
    n &= \text{number of items in resupply}, \\
    m &= \text{number of items installed}, \\
    s &= \text{number of items in stock}, \\
    \lambda &= \text{failure rate for a single item while installed}, \\
    \lambda_m &= \text{failure rate of installed items when number of items installed is } m, \\
    m_0 &= \text{minimum number of serviceable installed items required to accomplish mission objectives }(m \geq m_0 \text{ implies that service performance level is being met; } m < m_0 \text{ implies that the installed serviceable items are working at full capacity and are not fully meeting mission requirements}), \\
    m_1 &= \text{minimum number of installed serviceable items resulting in}
\end{align*} \]
no backorder \((m < m_1)\) implies that a backorder exists, 
\[N \geq m_1 \geq m_0 \geq 1,\]
\[
\mu = \frac{1}{\tau} = \text{resupply rate for a single item in resupply},
\]
\[
\tau = \text{mean resupply time for one item in resupply},
\]
\[
\rho = \frac{\lambda}{\mu} = \text{ratio of failure rate to resupply rate for one item in resupply},
\]
\[
\mu_n = \text{resupply rate when } n \text{ items are in resupply}.
\]

In the following development, a single server model is assumed so that

\[
\mu_n = \mu \text{ for } n \geq 1 \quad (6.1)
\]

and

\[
\mu_n = 0 \text{ for } n = 0 \quad (6.2)
\]

Note that when \(m > m_0\), mission requirements are being met with installed serviceable items operating at less than full capacity with failure rate 
\[
\lambda_m = m_0 \lambda; \text{ when } m < m_0 \text{ mission requirements are not being met and items are operating at full capacity with failure rate } \lambda_m = m \lambda.
\]

Observe that for a conservative system

\[
N = m + s + n \quad ,\]

(6.3)

and require that

\[
\lambda_m = m \lambda \quad , \quad 0 \leq m \leq m_0 \quad ,\]

(6.4)

\[
\lambda_m = m_0 \lambda \quad , \quad m_0 \leq m \leq N \quad .
\]

(6.5)
The above requirement for demand rate is equivalent to

\[ \lambda_m = (N-n)\lambda, \quad N-m_0 + 1 \leq n \leq N, \quad (6.6) \]

\[ \lambda_m = m_0\lambda, \quad 0 \leq n \leq N - m_0. \quad (6.7) \]

The state transition probabilities are given by

\[ \frac{\partial p^{(N)}_0(t)}{\partial t} = -m_0\lambda p^{(N)}_0(t) + \mu_1 p^{(N)}(t) \quad (6.8) \]

\[ \frac{\partial p^{(N)}_n(t)}{\partial t} = -m_0\lambda p^{(N)}_n(t) + \mu_{n+1} p^{(N)}(t) - \mu_n p^{(N)}_n(t) + m_0\lambda p^{(N)}_{n-1}(t), \quad 0 < n \leq N - m_0; \quad (6.9) \]

\[ \frac{\partial p^{(N)}_{N-1}(t)}{\partial t} = -(N-m_0)\lambda p^{(N)}_{N-1}(t) + \mu_{N-1}(t) - \mu_{N-1} p^{(N)}_{N-1}(t) + (N-m_0)\lambda p^{(N)}_{N-1}(t), \quad N > n \geq N - m_0 + 1; \quad (6.10) \]

\[ \frac{\partial p^{(N)}_N(t)}{\partial t} = -\mu_N p^{(N)}_N(t) + \lambda p^{(N)}_{N-1}(t), \quad n = N. \quad (6.11) \]

In the above, \( p^{(N)}_n(t) \) is the probability that \( n \) items are in resupply at time \( t \). Using standard techniques, one obtains the following steady-state solution \( t = \infty \).
\[
\begin{align*}
\Pr_n^{(N)} &= (m_0 \rho)^n \Pr_0^{(N)}, & 0 \leq n \leq N - m_0, \\
\Pr_n^{(N)} &= (m_0 \rho)^{N-m_0} \frac{m_0!}{(N-n)!} \rho^{n-(N-m_0)} \Pr_0^{(N)}, & N - m_0 + 1 \leq n \leq N.
\end{align*}
\]

Equation (6.13) can be rewritten
\[
\Pr_N^{(N)} = (m_0 \rho)^N \frac{m_0!}{x!} \Pr_0^{(N)}, & 0 \leq x \leq m_0 - 1.
\]

where \( x \in N - n \). The probability that there are zero units in resupply is
\[
\Pr_{0}^{(N)} = \frac{1 - m_0 \rho}{1 - (m_0 \rho) + (1 - m_0 \rho)(m_0 \rho)^N m_0! G_m(\rho)}.
\]

where
\[
G_m(\rho) = \sum_{x=0}^{m_0-1} \frac{\rho^{-x}}{x!}.
\]

Note that the backorder level is zero for \( 0 \leq n \leq N - m_1 \) and is equal to \( m_1 - x = m_1 - (N - n) \) for \( N - m_1 + 1 \leq n \leq N \). Thus, the expected backorder level is
\[
[N_B^{(N)}] = \sum_{x=1}^{m_1-1} (m_1 - x) \Pr_{N-x}^{(N)}.
\]
so that

\[ [N_B^{(N)}] = \sum_{x=0}^{m_0 - 1} (m_1 - x)(m_0 \rho)^N (m_0!m_\rho) P^N(x) \frac{x^x}{x!} + \sum_{x=m_0}^{m_1 - 1} (m_1 - x)(m_0 \rho)^N - x P^N(x). \]  

(6.18)

Solving for \( N = N([N_B^{(N)}], \rho) \) obtains

\[ \{ \ln(m_\rho) \} \{ N(\rho, [N_B^{(N)}]) \} = -\ln[N_B^{(N)}] + \ln \left\{ (1 - m_0 \rho) \left[ \sum_{x=0}^{m_0 - 1} (m_1 - x)m_\rho \frac{x^x}{x!} + \sum_{x=m_0}^{m_1 - 1} (m_1 - x)(m_0 \rho)^{-x} \right] \right. \]

\[ + \left. [N_B^{(N)}] \left[ (m_0 \rho)^{-m_0 + 1} - (1 - m_0 \rho)m_\rho ! m_0 ! G_{m_0}(\rho) \right] \right\}. \]

(6.19)

If \( m_0 = m_1 \), then

\[ [N_B^{(N)}] = \frac{(m_0 \rho)^N m_0 ! (1 - m_0 \rho)}{1 - (m_0 \rho) + (1 - m_0 \rho)(m_0 \rho)^N m_0 ! G_{m_0}(\rho)} \], \( m_0 = m_1 \),

(6.20)

and

\[ \{ \ln(m_\rho) \} \{ N(\rho, [N_B^{(N)}]) \} = \ln[N_B^{(N)}] - \ln \left\{ -m_0 ! (1 - m_0 \rho) G_{m_0}(\rho) \{ m - [N_B^{(N)}] \} \right. \]

\[ + \left. [N_B^{(N)}](m_0 \rho)^{-m_0 + 1} \right\}.

(6.21)

\[ m_0 = m_1 \].
For a given fixed value of \([N_B^{(N)}]\), equation (6.19) gives the relationship of \(N\) and \(p\). Thus, all values of \((N, p)\) that yield a particular value of \([N_B^{(N)}]\) may be obtained for comparison of inventory investment level to resupply time level for purposes of trade-off analysis described earlier in this document.

Note that when \(m_0 = 1\), then \(\lambda_m = \lambda\) for all \(m > 0\), corresponding to the situation where the failure rate is constant and independent of the number of installed serviceable units (except for \(m = 0\)). Returning to equation (6.15) and equation (6.18), one obtains

\[
P_n(N) = \frac{\rho^n (1 - \rho)}{1 - \rho^{N+1}} , \quad m_0 = 1 , \quad 0 \leq n \leq N ;
\]

\[
P_0(N) = \frac{1 - \rho}{1 - \rho^{N+1}} , \quad m_0 = 1 ;
\]

and

\[
[N_B^{(N)}] = \frac{\rho^{N-m_1+1}}{(1 - \rho)(1 - \rho^{N+1})} \left\{ 1 - \rho^{m_1} [m_1 (1 - \rho) + 1] \right\} , \quad m_0 = 1 .
\]

Further,

\[
N_S([N_B^{(N)}]) = \frac{(m_1-1)\lambda^m + \lambda[n_s^{(N)}] + \lambda(1-\rho) - \lambda[n_1 - \rho^{m_1} [1 + (1-\rho)(m_1-[n_s^{(N)}])]}{\lambda^m} , \quad m_0 = 1 .
\]

6-6
Thus, when \( m_0 = m_1 = 1 \)

\[
\left[ N_B^{(N)} \right] = \frac{(1 - \rho)^N}{1 - \rho^{N+1}}, \quad m_0 = m_1 = 1; \quad (6.26)
\]

\[
N(\rho, [N_B^{(N)}]) = \frac{\ln[N_B^{(N)}] - \ln \left\{ 1 - \rho (1 - [N_B^{(N)}]) \right\}}{\ln \rho}, \quad m_0 = m_1 = 1; \quad (6.27)
\]

\[
\left( \frac{\partial N}{\partial \rho} \right)_{[N_B^{(N)}]} = \frac{N / \rho - \left[ (1 - \rho^N) / (1 - \rho) \right]}{(-\ln \rho)}, \quad m_0 = m_1 = 1; \quad (6.28)
\]

\[
\left( \frac{\partial [N_B^{(N)}]}{\partial \rho} \right)_\rho = \frac{\ln \rho}{1 - \rho^{N+1}} \cdot \frac{(1 - \rho)^N}{1 - \rho^{N+1}}, \quad m_0 = m_1 = 1; \quad (6.29)
\]

\[
\left( \frac{\partial [N_B^{(N)}]}{\partial \rho} \right)_N = \frac{N - (1 - \rho)}{1 - \rho} \cdot \frac{(1 - \rho)^N}{1 - \rho^{N+1}}, \quad m_0 = m_1 = 1. \quad (6.30)
\]

Now consider \( \rho = m_1 = m_3 = 1 \). Then

\[
\left[ N_B^{(N)} \right] = \frac{1}{N+1} \cdot \left( \frac{\partial [N_B^{(N)}]}{\partial N} \right)_\rho = - \frac{1}{(N+1)^2}; \quad \left( \frac{\partial [N_B^{(N)}]}{\partial \rho} \right)_N = \frac{N}{2(N+1)};
\]

\[
\left( \frac{\partial N}{\partial \rho} \right)_{[N_B^{(N)}]} = \frac{N(N+1)}{2};
\]
and

\[ \delta N = \frac{N(N+1)}{2} \delta \rho \text{ for } [N_B^{(N)}] = \text{constant}. \] (6.31)

When \( N = \rho = m_1 = m_0 = 1 \), then

\[ [N_B^{(1)}] = \frac{1}{2}; \quad \left( \frac{\partial [N_B^{(1)}]}{\partial N} \right)_\rho = -\frac{1}{4}; \quad \left( \frac{\partial [N_B^{(1)}]}{\partial \rho} \right)_N = \frac{1}{4}; \]

\[ \left( \frac{\partial N}{\partial \rho} \right) [N_B^{(1)}] = 1. \]

Thus, if \([N_B^{(1)}]\) = constant, then

\[ \delta N = \lambda \delta \tau_\mu; \] (6.32)

if \( \rho = \) constant, then

\[ \delta [N_B^{(1)}] = -\frac{1}{4} \delta N; \] (6.33)

if \( N = \) constant, then

\[ \delta [N_B^{(1)}] = \frac{1}{4} \lambda \delta \tau_\mu. \] (6.34)

It is interesting to calculate \([N_B^{(N+1)}] - [N_B^{(N)}]\) and \(\left( \frac{\partial [N_B^{(N)}]}{\partial \rho} \right)_N\) in a fashion similar to that used in the case of the conservative parallel server model. Inspection of equation (7.12) and equation (7.13) obtains

6-8
\[ p_{N-x+1}^{(N+1)} = \left( m_0 \rho \right) \frac{p_0^{(N+1)}}{p_0^{(N)}} p_{N-x}^{(N)} , \quad 0 \leq x \leq N ; \quad (6.35) \]

and returning to equation (7.17) leads to

\[ \frac{[N_B^{(N)}] - [N_B^{(N+1)}]}{[N_B^{(N)}]} = 1 - m_0 \rho \frac{p_0^{(N+1)}}{p_0^{(N)}} > 0 . \quad (6.36) \]

Equation (6.12) and equation (6.13) yield

\[ \frac{1}{p_0^{(N)}} \left( \frac{\partial p_0^{(N)}}{\partial \rho} \right)_{N-x} = \frac{N-x}{\rho} + \frac{1}{p_0^{(N)}} \left( \frac{\partial p_0^{(N)}}{\partial \rho} \right)_{N} , \quad 0 \leq x \leq N ; \quad (6.37) \]

\[ \frac{1}{p_0^{(N)}} \left( \frac{\partial p_0^{(N)}}{\partial \rho} \right)_{N} \bigg) = - \frac{[n^{(N)}]}{\rho} . \quad (6.38) \]

Thus,

\[ \left( \frac{\partial [N_B^{(N)}]}{\partial \rho} \right)_{N} = \frac{[N_B^{(N)}]}{\rho} \left\{ \left( N - m_0 \right) - [n^{(N)}] + \frac{[N_B^{(N)}]^2}{[N_B^{(N)}]} \right\} > 0 , \quad (6.39) \]

an expression which is identical in form to that obtained for the parallel server case.

Other formulae which apply are given below.

\[ [N_B^{(N+1)}] - [N_B^{(N)}] = \Delta[N_B^{(N)}] = -p_0^{(N+1)} [N_B^{(N)}] ; \quad (6.40) \]
\begin{align*}
\Delta p_s^{(N+1)} & = \sum_{s=0}^{N} - p_s^{(N)} = -P_0 p_s^{(N)}; \quad (6.41) \\
\frac{P_{n+1}^{(N)}}{P_n^{(N)}} & = m_n \rho, \quad 0 \leq n \leq N - m_0; \\
\frac{P_{N-x+1}^{(N)}}{P_{N-x}^{(N)}} & = m_0 \rho, \quad N - 1 \geq x \geq m_0; \quad (6.42) \\
\frac{P_{n+1}^{(N)}}{P_n^{(N)}} & = \rho(N-n), \quad N - m_0 + 1 \leq n \leq N - 1; \quad (6.43) \\
\frac{P_{N+1-x}^{(N)}}{P_{N-x}^{(N)}} & = \rho x, \quad 1 \leq x \leq m_0 - 1; \\
\frac{m_n \rho}{P_0^{(N)}} & = 1 - P_0^{(N+1)}; \quad (6.44)
\end{align*}
SECTION 7
EXAMPLE COST FUNCTION

The total system cost is given by

\[ C = C(N, \rho) \],

(7.1)

where

\[ N = \text{number of inventory investment items}, \]
\[ \rho = \frac{\lambda}{\mu} = \frac{\tau}{\tau_{\lambda}} = \text{ratio of mean resupply time to mean time between demands.} \]

Total system cost is expected to increase linearly with increasing \( N \) and to increase with decreasing \( \rho \) (decreasing \( \tau_{\mu} \)). For purposes of exposition, assume a simple cost function of the form

\[ C(N, \rho) = c_1 N + \frac{c_2 N}{\rho}, \]

(7.2)

where

\[ c_1 = \text{cost per investment item}, \]
\[ c_2 = \text{cost per investment item per unit } \rho. \]

Thus,

\[ \text{inventory investment cost } = c_1 N, \]
\[ \text{cost for resupply system } = \frac{c_2 N}{\rho}. \]

Define

\[ z(N, \rho) \equiv \frac{C(N, \rho)}{c_1} = N \left(1 + \frac{c_2}{c_1 \rho}\right), \]

(7.3)

and

\[ \rho_0 \equiv \frac{c_2}{c_1}. \]

(7.4)
Thus,

\[ z(N, \rho) = N \left( 1 + \frac{\rho_0}{\rho} \right) \]  

(7.5)

\( z(N, \rho) \) is a unitless cost function expressed in terms of number of equivalent investment items.

\[ z = z_1 + z_2 \]  

(7.6)

\[ z_1 = N = \text{number of investment items} \]

\[ z_2 = \frac{N\rho_0}{\rho} = \text{cost of resupply system in terms of equivalent investment items} \]

If \( \frac{\rho_0}{\rho} = 1 \), then for every dollar spent for inventory investment an equal amount is spent for the resupply system.
SECTION 8
PROBLEM FORMULATION

The mathematical statement of the problem takes on two possible forms.

**PROBLEM 1**

Minimize

\[ z = z(N, \rho) \] \hspace{1cm} (8.1)

Subject to

\[ [N_B(N, \rho)] = N_B^{(0)} = \text{constant} \ , \]

\[ N = \text{integer} > 0 \] \hspace{1cm} (8.2)

\[ \rho > 0 \ . \]

**PROBLEM 2**

Minimize

\[ [N_B] = [N_B(N, \rho)] \] \hspace{1cm} (8.3)

Subject to

\[ z(N, \rho) = z_0 = \text{constant} \ , \]

\[ N = \text{integer} > 0 \ , \] \hspace{1cm} (8.4)

\[ \rho > 0 \ . \]
SECTION 9
PROBLEM SOLUTION

In this section are presented the procedures to be followed for problem solution. PROBLEM 1 and PROBLEM 2 are restated below, and associated optimality conditions are given.

**PROBLEM 1**

Minimize

\[ z = z(N, \rho) \tag{9.1} \]

Subject to

\[ \rho = \rho(N_{B}^{(0)}, N) = \rho(N), \text{ with } N_{B}^{(0)} \text{ fixed.} \tag{9.2} \]

\[ z = N \left( 1 + \frac{\rho_{0}}{\rho(N_{B}^{(0)}, N)} \right), \tag{9.3} \]

\[ z = N \left( 1 + \frac{\rho_{0}}{\rho(N)} \right) \tag{9.4} \]

Since \( N \) is an integer,

\[ z[N+1, \rho(N+1)] - z[N, \rho(N)] \geq 0 \tag{9.5} \]

and

\[ z[N-1, \rho(N-1)] - z[N, \rho(N)] \geq 0 \tag{9.6} \]

when \( z \) is a minimum.
PROBLEM 2

Minimize

\[ [N_B] = [N_B(N, \rho)] \quad , \quad (9.7) \]

Subject to

\[ \rho = \rho(N, z_0) = \rho(N) \quad , \quad \text{with } z_0 \text{ fixed.} \quad (9.8) \]

\[ \rho(N) = \frac{N\rho_0}{z_0 - N} \quad , \quad N < z_0 ; \quad (9.9) \]

\[ [N_B(N, \rho)] = [N_B(N, \rho(N))] \quad . \quad (9.10) \]

If \( N \) is an integer, then

\[ [N_B(N+1, \rho(N+1))] \geq [N_B(N, \rho(N))] \quad (9.11) \]

and

\[ [N_B(N-1, \rho(N-1))] \geq [N_B(N, \rho(N))] \quad , \quad (9.12) \]

when \([N_B] \) is a minimum.

Both of the above integer programming problems are solved by a simple search utilizing a gradient technique (described in Section 10) that obtains solutions in two to six interactions. A similar technique is utilized to obtain solutions to the equation

\[ \rho = \rho(N_B^{(0)}, N) \quad , \quad \]

where \( \rho \) is to be determined when \( N \) and the backorder level \( N_B^{(0)} \) are
specified. In general, PROBLEM 1 requires more computing time than that required for PROBLEM 2. The solution to \( p = p(N_B^{(o)}, N) \) requires a great deal more time to obtain than that required for the solution of

\[
p(N) = \frac{N_0}{z_0 - N}
\]

The mathematical programming problems described above have been thoroughly investigated for the backorder function described by (1) the Parallel Server Model and for (2) the Poisson Model. Comparisons and conclusions are given in Section 12.
SECTION 10
A GRADIENT TECHNIQUE TO SOLVE A SET OF COUPLED EQUATIONS

Let

\[
\mathbf{r} = \begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{pmatrix}
\quad \text{and} \quad
\mathbf{e} = \begin{pmatrix}
e_1 \\
e_2 \\
\vdots \\
e_n
\end{pmatrix}
\]

be n component column vectors and the gradient operator \( \nabla \) be defined to be the row vector

\[
\nabla \equiv \begin{pmatrix}
\frac{\partial}{\partial x_1} \\
\frac{\partial}{\partial x_2} \\
\vdots \\
\frac{\partial}{\partial x_n}
\end{pmatrix}
\]

It is desired to find a solution to the set of n simultaneous equations

\[
g_i(\mathbf{r}) = 0, \quad i = 1, 2, 3, \ldots, n
\]

in n unknowns represented by the column vector \( \mathbf{r} \). The first order Taylor expansion about a given point \( \mathbf{r}_o \) is

\[
g_i(\mathbf{r}_o + \mathbf{e}) = \nabla g_i \mathbf{e} + g_i(\mathbf{r}_o)
\]

where \( \nabla g_i = \nabla g_i(\mathbf{r}_o) \); i.e., the first derivatives are evaluated at the point \( \mathbf{r}_o \) and \( \mathbf{r}_o \) is assumed to be an approximate solution to equations (10.3).

If \( \mathbf{r} = \mathbf{r}_o + \mathbf{e} \) is to be a solution to equation (10.3), then

\[
g_i(\mathbf{r}_o + \mathbf{e}) = 0, \quad i = 1, 2, \ldots, n
\]
and
\[ \nabla g_i = -g_i(r_o), \quad i = 1, 2, \ldots, n. \]  
(10.6)

Restating equation (10.6) leads to
\[ \sum_{j=1}^{n} (\nabla g_i)_j e_j = -g_i(r_o). \]  
(10.7)

Now define the square n x n matrix \( G \) by
\[ G = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_n} \end{pmatrix} \]
(10.8)

and the n component column vector \( g(r_o) \) by
\[ g(r_o) = \begin{pmatrix} g_1(r_o) \\ g_2(r_o) \\ \vdots \\ g_n(r_o) \end{pmatrix}. \]  
(10.9)
In the above it is understood that all derivatives \( \frac{\partial g_i}{\partial x_j} \) and all functions \( g_i(r_o) \) are evaluated at the point \( r = r_o \). It is observed that

\[
[G(r_o)]_{ij} = [\nabla g_i(r_o)]_{j} \quad \text{and} \quad [g(r_o)]_i = g_i(r_o)
\]  

(10.10)

Therefore,

\[
\sum_{j=1}^{n} G_{ij}(r_o) \epsilon_j = -g_i(r_o)
\]

\[
G(r_o) \epsilon = -g(r_o)
\]  

(10.11)

The above implies that

\[
\epsilon = -G^{-1}(r_o)g(r_o)
\]  

(10.12)

or

\[
\epsilon = -G^{-1}g
\]  

(10.13)

if the matrix \( G^{-1} \) exists (i.e., if \( \det |G| \neq 0 \)). The individual components of \( \epsilon \) are given by

\[
\epsilon_j = - \sum_{k=1}^{n} G_{jk}^{-1} g_k, \quad j = 1, 2, \ldots, n.
\]  

(10.14)

As long as \( r_o \) is "reasonably close" to the particular solution desired and no pathologic conditions exist (forcing \( \det |G| = 0 \)), then equation (10.13) can be used to successive arrive at values of \( r \) which satisfy equation (10.3) as precisely as desired.

The procedure simply stated is as follows:

(i) select an initial trial solution value \( r = r_1 \), where \( i = \) the iteration number (\( i = 0 \) initially);
(ii) apply equation (10.13) to determine the correction vector \( \xi_i \);
(iii) check to see if \( \xi_i \) satisfies \( |\xi_i| < \xi_{\text{max}} \) where the \( \xi_{\text{max}} \) = \( \xi_{\text{max}} \) > 0 represent upper limit values on the errors to be tolerated in the solution values for each of the variables \( x_j \);
(iv) (a) if \( |\xi_i| > \xi_{\text{max}} \), then set \( r_{i+1} = r_i + \xi_i \) (i.e., calculate a new trial solution to the system of equations \( g(r) = 0 \)) and return to step (ii) above with \( i = i + 1 \); (b) if \( |\xi_i| < \xi_{\text{max}} \), then stop and calculate \( r_{i+1} = r_i + \xi_i \), where \( r_{i+1} \) represents the desired solution to \( g(r) = 0 \).

As an example, consider the problem

\[
\begin{align*}
g_1(x, y) &= x^2 - y^2 - 4 = 0, \\
g_2(x, y) &= x + y - 1 = 0.
\end{align*}
\]

The solution to the above problem is readily obtained by conventional techniques:

\[
\begin{align*}
x + y &= 1; \\
x - y &= 4; \\
2x &= 5 \\
x &= 5/2 \text{ and } y = -3/2.
\end{align*}
\]

To illustrate the technique developed above proceed as follows:

\[
\nabla g_1 = (2x, -2y), \\
\nabla g_2 = (1, 1); \\
G = \begin{pmatrix} 2x & -2y \\ 1 & 1 \end{pmatrix}; \\
\det G = 2(x + y); \\
G^{-1} = \frac{1}{2(x + y)} \begin{pmatrix} 1 & 2y \\ -1 & 2x \end{pmatrix}; \\
g = \begin{pmatrix} x^2 - y^2 - 4 \\ x + y - 1 \end{pmatrix}.
\]
Choose \( \mathbf{r}_0 = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \) as the initial trial solution.

\[
G^{-1}(\mathbf{r}_0) = \frac{1}{4} \begin{pmatrix} 1 & 2 \\ -1 & 2 \end{pmatrix}; \quad g(\mathbf{r}_0) = \begin{pmatrix} -4 \\ +1 \end{pmatrix};
\]

\[
\mathbf{e}_0 = -\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} -4 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ -\frac{3}{2} \end{pmatrix} \neq 0 .
\]

Iteration No. 1 proceeds as follows:

\[
\mathbf{r}_1 = \mathbf{r}_0 + \mathbf{e}_0 = \begin{pmatrix} 3/2 \\ 1/2 \end{pmatrix}
\]

\[
G^{-1}(\mathbf{r}_1) = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} \end{pmatrix}; \quad g(\mathbf{r}_1) = \begin{pmatrix} -2 \\ 0 \end{pmatrix};
\]

\[
\mathbf{e}_1 = -\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} \end{pmatrix} \begin{pmatrix} -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \neq 0 .
\]

Iteration No. 2 proceeds as follows:

\[
\mathbf{r}_2 = \mathbf{r}_1 + \mathbf{e}_1 = \begin{pmatrix} 3/2 \\ -1/2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 5/2 \\ -3/2 \end{pmatrix}
\]

\[
G^{-1}(\mathbf{r}_2) = \frac{1}{2} \begin{pmatrix} 1 & -3 \\ -1 & 5 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{3}{2} \\ -\frac{1}{2} & \frac{5}{2} \end{pmatrix}; \quad g(\mathbf{r}_2) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]

10-5
\[ \therefore \xi_3 = 0 \text{ and } \mathbf{x}_3 = \begin{pmatrix} 5 \\ -3 \\ 2 \end{pmatrix} \text{ is a solution.} \]

Note that it is not necessary that the first approximation \( x_0 \) be terribly good; what is required is that the \( x_0 \) chosen be "closer" to the solution of interest than to any other possible solution (thus preventing the possibility of moving along in the wrong direction) and that in successive calculations no pathological conditions causing \( \det |G| \) to be zero occur [such as encountering a saddle point or a relative minimum or maximum for any of the \( g_i(x) \)]. If such a condition does arise, however, it is still possible to apply the technique by "skipping over" the critical point encountered [Note that \( \det |G| = 0 \) for any \( x \) which is a critical point for any of the \( g_i(x) \) in \( g(x) \)]. A "good" value for \( x_0 \) eliminates such a possibility from occurring, and in general no such problem occurs in practical applications.

The one dimensional case of the solution technique is a special case of the development presented above. To obtain a solution to the equation

\[ g(x) = 0 \quad (10.15) \]

note that

\[ g(x_0 + \varepsilon) = \left[ \frac{\partial g(x)}{\partial x} \right]_{x=x_0} \varepsilon + g(x_0) \quad (10.16) \]

and

\[ \varepsilon = -g(x_0) \left[ \frac{\partial g}{\partial x} \right]_{x=x_0}^{-1} \quad (10.17) \]

if \( x = x_0 + \varepsilon \) is to be a solution to equation (10.15). The procedure to determine the solution is precisely the same as that given in steps (i) through (iv) above.
An example illustration of the one dimensional case is given below.

In order to solve

\[ x + \ln x = 0 \]

for \( x \), one proceeds as follows:

\[
g(x) = x + \ln x \quad ;
\]

\[
\frac{\partial g}{\partial x} = 1 + \frac{1}{x} \quad ;
\]

\[
\epsilon = \frac{-x(x + \ln x)}{1 + x} .
\]

Try \( x = 1 \) as a trial solution. Then

\[ \epsilon_o = -0.500. \]

Iteration No. 1 proceeds as follows:

\[ x_1 = x_0 + \epsilon_0 = 0.5 \quad ; \]

\[ \epsilon_1 = \frac{-0.5(0.5 - 0.693147)}{1.5} = +0.0643824 \quad . \]

Iteration No. 2 proceeds as follows:

\[ x_2 = x_1 + \epsilon_1 = 0.5 + 0.0643824 = 0.5643824 \quad ; \]

\[ \epsilon_2 = \frac{-(0.5643824)(0.5643824 - 0.5720233)}{1.5643824} ; \]

\[ \epsilon_2 = +0.00275659 \quad . \]

Iteration No. 3 proceeds as follows:

\[ x_3 = x_2 + \epsilon_2 = 0.5671389835 \quad ; \]

\[ \epsilon_3 = \frac{-(0.5671389835)(0.5671389835 - 0.5671508844)}{1.5671390} ; \]

\[ \epsilon_3 = +0.0000043069 \quad . \]
Iteration No. 4 proceeds as follows:

\[ x_4 = x_3 + \varepsilon_3 = 0.5671432904 \ ; \]

\[ \varepsilon_4 = \frac{-(0.5671432904)(0.5671432904 - 0.5671432904)}{1.5671432904} ; \]

\[ \varepsilon_4 = 0 . \]

\[ \therefore \ x = 0.5671432904 \text{ to nine significant figures} . \]

If a solution value to the equation \( g(x) = 0 \) is desired when \( x \) is constrained to be an integer, then the above procedure is modified as follows.

\[ g(x_0 + \varepsilon) = g(x_0) + \Delta g(x_0) \varepsilon \approx 0 \quad , \]

\[ \Delta g(x_0) = g(x_0 + 1) - g(x_0) \quad , \]

\[ \varepsilon = \begin{bmatrix} - \frac{g(x_0)}{\Delta g(x_0)} \end{bmatrix}^T \quad , \]

where the subscript \( T \) means set \( \varepsilon = \text{integer} = \text{truncated value of the numerical quantity enclosed in the brackets}. \) The iterative procedure is then that given by steps (i) through (iv) given earlier in this section with equation (10.20) being substituted for equation (10.17) and the initial value for \( x_0 \) being set equal to an integer. If no solution to \( g(x) = 0 \) exists (this is possible since \( x = \text{integer} \)), then the above procedure may be utilized in finding the value of \( x \geq 0 \) that minimizes \( |g(x)| \) and the value of \( x < 0 \) (if allowed) that minimizes \( |g(x)| \).

In the solution procedure for PROBLEM 1, the gradient technique is applied to determine \( \rho \) in equation (9.2) and is used to aid in the search for the optimal value of \( N \) which satisfies the conditions expressed in relations (9.5) and (9.6). The search for \( (N, \rho) \) is conducted along a curve of constant \([N_B]\). In the solution procedure for PROBLEM 2, the gradient technique is
used to aid in the search for the optimal value of $N$ which satisfies the conditions expressed in relations (9.11) and (9.12). Here, the search for $(N, \rho)$ is conducted along a curve of constant $z$.

The solution procedure developed here for PROBLEM 1 and PROBLEM 2 is easily adapted to any cost function of the form $C = C(N, \rho)$ and for any backorder function of the form $[N_B] = [N_{B}(N, \rho)]$. In this study, PROBLEM 1 and PROBLEM 2 have been investigated with the backorder functions described by the Poisson Model (Section 5) and the Parallel Server Model (Section 4) and with $z = z(N, \rho)$ as described in Section 7.
SECTION 11
SENSITIVITY ANALYSIS

It is of interest to determine the range of \( p \) for which the optimum value of \( N \) remains unchanged. For PROBLEM 1, application of the optimality condition and the corresponding constraint yields values for \((p_0)_{\text{max}}\) and \((p_0)_{\text{min}}\) such that when

\[
(p_0)_{\text{min}} \leq p_0 \leq (p_0)_{\text{max}}
\]

the current optimal value obtained for the \((N, p)\) ordered pair remains unchanged; i.e., the optimal solution to PROBLEM 1 is independent of the choice of \( p_0 \) within the interval \((p_0)_{\text{min}} \leq p_0 \leq (p_0)_{\text{max}}\). Similarly, application of the optimality condition for PROBLEM 2 and the corresponding constraint yields values for \((p_0)_{\text{max}}\) and \((p_0)_{\text{min}}\) such that when

\[
(p_0)_{\text{min}} \leq p_0 \leq (p_0)_{\text{max}}
\]

the current optimal value of \( N \) remains unchanged; i.e., the optimal solution value for \( N \) for PROBLEM 2 is independent of the choice of \( p_0 \) within the interval \((p_0)_{\text{max}} \geq p_0 \geq (p_0)_{\text{min}}\). However, in the case for PROBLEM 2, if \( p_0 \) is allowed to vary and the optimum value of \( N \) remains fixed, then the optimum value of \( p \) must change to keep \( z = \text{constant} \).

Let \((N, p)\) be an optimal solution to PROBLEM 1. Define \( \Delta p_1 \geq 0 \) and \( \Delta p_2 \geq 0 \) by

\[
[N_B(N_1 + 1, p_1 + \Delta p_1)] = N_B^{(1)} \quad ,
\]

\[
[N_B(N_1 - 1, p_1 - \Delta p_2)] = N_B^{(0)} \quad .
\]
Condition (9.5) in Section 9 becomes

\[(N_1 + 1) \left( 1 + \frac{\rho_0}{\rho_1 + \Delta \rho_1} \right) \geq N_1 (1 + \frac{\rho_0}{\rho_1}) \quad (11.5)\]

and yields

\[\frac{\rho_0}{\rho_1} \leq \frac{1 + \Delta \rho_1 / \rho_1}{N_1 \frac{\rho_1 - 1}{\rho_1}} \quad (11.6)\]

Condition (9.6) in Section 9 becomes

\[(N_1 - 1) \left( 1 + \frac{\rho_0}{\rho_1 - \Delta \rho_2} \right) \geq N_1 (1 + \frac{\rho_0}{\rho_1}) \quad (11.7)\]

and yields

\[\frac{\rho_0}{\rho_1} \geq \frac{1 - \Delta \rho_2 / \rho_1}{N_1 \frac{\rho_1 - 1}{\rho_1}} \quad (11.8)\]

Thus, any \(\rho_0\) satisfying the condition

\[\frac{1 - \frac{\Delta \rho_2}{\rho_1}}{N_1 \frac{\rho_1 - 1}{\rho_1}} \leq \frac{\rho_0}{\rho_1} \leq \frac{1 + \frac{\Delta \rho_1}{\rho_1}}{N_1 \frac{\rho_1 - 1}{\rho_1}} \quad (11.9)\]

will yield an optimal solution \((N_1, \rho_1)\) for PROBLEM 1.

Let \((N_1, \rho_1)\) now be an optimum solution to PROBLEM 2. Define \(\Delta \rho_1 \geq 0\) and \(\Delta \rho_2 \geq 0\) by

\[[N_B(N_1 + 1, \rho_1 + \Delta \rho_1)] = [N_B(N_1, \rho_1)] \quad , \quad (11.10)\]
\[ [N_B(N_1 - 1, \rho_1 - \Delta \rho_2)] = N_B(N_1, \rho_1) \quad . \] (11.11)

Equation (9.9) yields
\[ \rho_1 + \Delta \rho_1 = \frac{(N_1 + 1) \rho_0'}{z_0 - (N_1 + 1)} \quad , \] (11.12)

\[ \frac{\rho_0'}{\rho_1} = (1 + \Delta \rho_1 / \rho_1) \left( \frac{z_0}{N_1 + 1} - 1 \right) \quad , \] (11.13)

where \( \rho_0' \) is the upper limit for \( \rho_0 \) such that the optimal value for \( N \) remains \( N_1 \). Equation (9.9) also yields
\[ \rho_1 - \Delta \rho_2 = \frac{(N_1 - 1) \rho_0''}{z_0 - (N_1 - 1)} \quad , \] (11.14)

\[ \frac{\rho_0''}{\rho_1} = \left( 1 - \frac{\Delta \rho_2}{\rho_1} \right) \left( \frac{z_0}{N_1 - 1} - 1 \right) \quad , \] (11.15)

where \( \rho_0'' \) is the lower limit for \( \rho_0 \) such that the optimal value for \( N \) remains \( N_1 \). Thus, any \( \rho_0 \) satisfying the condition
\[ \left( 1 - \frac{\Delta \rho_2}{\rho_1} \right) \left( \frac{z_0}{N_1 - 1} - 1 \right) \leq \frac{\rho_0}{\rho_1} \leq (1 + \Delta \rho_1 / \rho_1) \left( \frac{z_0}{N_1 + 1} - 1 \right) \] (11.16)

will yield a value of \( N_1 \) in the optimal solution for PROBLEM 2. Note that the optimal value of \( \rho_1 \) varies according to
\[ \rho = \frac{N \rho_0}{z_0 - N} \quad . \] (11.17)

as the value of \( \rho_0 \) is varied.
Conditions (11.9) and (11.16) can be utilized to study the sensitivity of the optimal fractional investment level to the relative unit cost associated with the resupply system. Results of comparing the Poisson Model to the Parallel Server Model are given in Section 12.
SECTION 12
RESULTS AND CONCLUSIONS

Computer programs have been written and executed for

1) The Conservative Parallel Server Inventory Model described in Section 4,
2) The Poisson Model described in Section 5,
3) The Conservative Single Server Inventory Model described in Section 6.

Programs have also been written and executed for solving PROBLEM 1 and PROBLEM 2 described in Sections 8 and 9 utilizing the gradient technique described in Section 10 and procedures outlined in Section 9. Included in these programs are the sensitivity analyses given in Section 11. Program listings are found in Appendix A.

Reference material developed during the course of this project is given in the "Reference List and Bibliography" and in "Documents Internal to AFLC".

Various inputs were assumed to execute the computer programs to perform a detailed analysis and comparison of the behavior of the parallel server model (finite population) and the Poisson model (infinite population). Below are listed the major results.

- The Poisson demand process overestimates the backorder level for a given \((N,p)\) specification.
- In general, the Poisson Model obtains a lower value for the optimum value of inventory investment than does the finite population model; as the system requirement \(m_0\) increases, the difference between the optimum levels of inventory investment given by the two models becomes increasingly larger. Thus,
use of the Poisson Model underestimates optimum inventory investment level and results in a higher backorder level and a higher stockout probability than does the finite population level.

- If the relative unit cost ($\rho_0$) for resupply increases, then the difference between the optimum levels of inventory investment given by the two models becomes increasingly larger, approaching levels of 15-50%.

- In general, the finite population model given an optimum level for inventory investment which is less sensitive to a change in unit resupply cost than that suggested by the Poisson Model; i.e., the range for $\rho_0$ where the optimal solution remains invariant is larger for the finite population model than for the infinite population model.

- As the system requirement $m_0$ becomes larger, the optimum level for inventory investment becomes more sensitive to a change in unit resupply cost.

Some numerical outputs are tabulated in Appendix B.

The above results conclude that if the relative cost for resupply (manpower, equipment, administration, transportation, etc.) compared to inventory investment increases, then it becomes increasingly important that demand and resupply processes be accurately described to obtain cost effective allocations of expenditures between inventory investment levels (capital intensive) and resupply time levels (labor intensive). In addition, if the variance (uncertainty) of costs increases, then it is also important that correct descriptions of the demand and resupply processes be utilized in the decision making process. It is noted that inventory investment costs for recoverable items are well-known in comparison to projected costs for resupply during the life cycle of a typical weapons system. Thus, the fact that the finite population model tends to allocate a greater fraction of total expenditures to inventory investment than does the infinite population model makes the
finite population model even more attractive for adoption in the decision making process.

Non-optimal allocations that result when incorrect demand and resupply processes are assumed suggest the following areas for future investigation.

- Study resupply processes which allow for non-zero queue lengths (finite number of parallel servers).
- Generalize the trade-off analysis procedure developed in this report (resupply time level versus inventory investment level to minimize total system cost) to include interactions due to movement of many investment items in the transportation pipelines among user bases and supply depots.
- Further investigate demand and resupply modeling for recoverable items for purposes of determining transportation flows and optimal allocating of resources to inventory investment level and resupply time level.
- Investigate and compare the utility of the finite population model developed in this report with that of the Poisson model currently adopted by Air Force management [88, 89, 104] when incorporated into the METRIC and MOD-METRIC recoverable item control and allocation models.
- Investigate and compare the finite population model with the Poisson model in determining impacts on output of the Comprehensive Engine Management System (CEMS) being developed by the U.S. Air Force [D-11].

The above areas of investigation and those described in this report should contribute to a better understanding of certain problems of immediate concern to the U.S. Air Force. The questions dealt with here are very important for considerations that must be made when an integrated design effort is made to develop any proposed weapon system.
The work reported in this document was presented at two technical conferences which took place during the months of January and February of 1978. Particulars are given below.


The above conferences were sponsored by the Operations Research Society of America (ORSA), The Institute of Management Sciences (TIMS), and the American Institute of Industrial Engineers (AIIE).
SECTION 13
REFERENCE LIST AND BIBLIOGRAPHY


SECTION 14
DOCUMENTS INTERNAL TO AFLC


D-4. "Civil Aeronautics Board Coordinates Sequenced Ascending by City Name,"--longitude, latitude coordinates for cities and distances between cities.


APPENDIX A
PROGRAM LISTINGS

CONSERVATIVE PARALLEL SERVER INVENTORY MODEL
(FINITE POPULATION) PROGRAM LISTING A-2

POISSON MODEL (INFINITE POPULATION) PROGRAM LISTING A-17

CONSERVATIVE SINGLE SERVER INVENTORY MODEL
(FINITE POPULATION) PROGRAM LISTING A-27

PROBLEM 1 PROGRAM LISTING A-45

PROBLEM 2 PROGRAM LISTING A-52
CONSERVATIVE PARALLEL SERVER
INVENTORY MODEL
(FINITE POPULATION)
PROGRAM LISTING
/LOGON "NC"
/DO RUN.WATFIV

$JOB (L9NTJQQ) KOVACS

DOUBLE PRECISION P(101),PA(101),DER2(101),NB,NB1,PO,PO1,Z,NTOT,R0

1M1,M0,ANS19,DER1,DER3,VAR1,VAR2,NBSQ,VARN,RESPSQ,VARS,SN,MN,VAR,M

2SSQ,MNSQ,T
Z=0.

PRINTS

5 FORMAT(*10,5X,**KOVACS FINITE SERVER INVENTORY MODEL***)

READ(5,**) NTOT,RO,M0,M1

PRINT10,NTOT,RO,M0,M1


IX=NY=0

CALL NINE(NTOT,M0,P,PO,RO,IX,II)

CALL TEN(NTOT,M0,P,PO,RO,IX,II)

C IX AND IY ARE TOTALS FOR OVERFLOWS AND UNDERFLOWS IN THE PROGRAM.

PRINT220,IX,II

150 FORMAT(*-P(N) IS THE PROBABILITY THAT N ITEMS ARE IN RESUPPLY AT
TIME NEARS INFINITY*)

C T IS THE TOTAL OF THE PROBABILITIES -- SHOULD EQUAL 1.00.

T=PO

DO 100 I=1,ITOT

100 T=T+P(I)

140 FORMAT(*THE EXPECTED NUMBER OF BACKORDERS IS **D20.14*)

A-3
160 FORMAT(‘THE TOTAL OF ALL THE P VALUES IS ’,D20.14)
PRINT160,T
CALL SIXTEN(NTOT,M1,P,NB)
PRINT140,NB
I6=NTOT-M1+1
IF(16.LE.0)Z=P0
IF(16.LE.0)I6=1
DO 300 I7=16,ITOT
300 Z=Z+P(I7)
C Z IS THE TOTAL OF ALL THE PROBABILITIES FROM I=NTOT-M1+1 TO I=NTOT.
PRINT301,Z
301 FORMAT(‘THE PROBABILITY FOR STOCKOUT IS ’,D20.14)
IX=IY=0
CALL NINE(NTOT+1,M0,PA,P01,R0,IX,IY)
CALL TEN(NTOT+1,M0,PA,P01,R0,IX,IY)
I79=ITOT+1
PRINT220,IX,IY
220 FORMAT(‘THERE WERE ’,I3,’ UNDERFLOWS AND ’,I3,’ OVERFLOWS’)
T=P01
DO 102 I=1,I79
102 T=T+PA(I)
PRINT160,T
CALL SIXTEN(NTOT+1,M1,PA,NB1)
PRINT140,NB1
PRINT142,NE-NB1
142 FORMAT(‘ DELTA NB=’,D20.14)
PRINT400,ITOT,ITOT+1,ITOT+(NE-NB1)/NB
400 FORMAT(*ONB(*,I3,*)) = NB(*,I3,*)) / NB(*,I3,*)) = *D20,14)
CALL SUB21(NTOT,P,DER1)
PRINT310,DER1

310 FORMAT(*THE EXPECTED NUMBER OF UNITS IN RESUPPLY IS *D20,14)
PRINT320,DER1/(0.0-RO)

320 FORMAT(*THE RESULT OF FORMULA 21 IS *D24,14)
CALL SUB20(NTOT,RO,DER1,DER2)
PRINT330
PRINT330,0,DER1/(0.0-RO)
PRINT330,1,DER2(1),I=1,ITOT)

330 FORMAT(*ANSWERS FOR FORMULA 20*)

330 FORMAT(*PC(*,I3,*)) GIVES *D24,14)
CALL VAR1(PO,M1,NTOT,VAR,P,NB)
CALL VAR1(P01,M1,NTOT+1,VARA,PA,NB1)
PRINT350,ITOT+VAR

350 FORMAT(*THE VARIANCE OF NB FOR NTOT=*I3,* IS *D20,14)
PRINT350,ITOT+1,VARA

DER3=0.0
CALL VAR1(PO,M1,NTOT,NBSQ,P,DER3)
CALL SUB22(NTOT,M1,NE,DER1,DER3,RO,NBSQ)
PRINT360,DER3

360 FORMAT(*THE ANSWER TO FORMULA 22 IS *D20,14)
PRINT352,NBSQ
CALL VAR2(DER1,P,VARN,PO,NTOT)

352 FORMAT(*THE EXPECTED VALUE OF THE SQUARE OF THE BACKORDER LEVEL IS *D20,14)
PRINT410,VARN

A-5
410 FORMAT('THE VARIANCE OF THE EXPECTED NUMBER OF UNITS IN RESUPPLY IS: \$D20.14)

    DER3=0.0

    CALL VAR2(DER3, P, RESPSQ, PO, NTOT)

    PRINT420, RESPSQ

420 FORMAT('THE EXPECTED VALUE OF THE SQUARE OF THE NUMBER OF UNITS IN RESUPPLY IS: \$D20.14)

    CALL SNAVE(NTOT, M1, P, PO, SN)

    PRINT430, SN

430 FORMAT('THE EXPECTED STOCK LEVEL IS: \$D20.14)

    CALL VAR3(NTOT, M1, P, PO, VARS, SN)

    PRINT440, VARS

440 FORMAT('THE VARIANCE OF THE STOCK LEVEL IS: \$D20.14)

    U=0.0

    CALL VAR3(NTOT, M1, P, PO, SSQ, U)

425 FORMAT('THE EXPECTED VALUE OF THE SQUARE OF THE STOCK LEVEL IS: \$D20.14)

    PRINT425, SSQ

    CALL MNAVE(NTOT, M1, P, PO, MN)

    PRINT435, MN

435 FORMAT('THE EXPECTED NUMBER IN SERVICE IS: \$D20.14)

    CALL VAR4(NTOT, M1, MN, P, VARM, PO)

    PRINT444, VARM

444 FORMAT('THE VARIANCE OF THE EXPECTED NUMBER IN SERVICE IS: \$D20.14)

    CALL VAR4(NTOT, M1, U, P, MNSQ, PO)

    PRINT446, MNSQ
FORMAT(*THE EXPECTED VALUE OF THE SQUARE OF THE NUMBER IN SERVICE)

1 IS1*, D20.14)
STOP
END

SUBROUTINE NINE(NTOT, MO, P, PO, R0, IX, IY)
C FINDS THE PROBABILITIES FOR J = 0 TO J = NTOT - MO (FINDS P(J)).
C USES NATURAL LOGS AND EXPONENTS TO ALLOW FOR LARGER SYSTEMS

DOUBLE PRECISION P(101), NTOT, MO, PO, R0, A, B, C, D, E, F, G, H, Z, Y

DOUBLE PRECISION DLOG, DEXP

ITEST = NTOT - MO + 1

DO 10 J1 = 1, ITEST
Y = 2.0
J = J1 + 1

DO 20 N1 = 1, ITEST
NRESUP = N1 - 1
A = (NRESUP - J) + DLOG(MO + R0)
C = D = 0.0

IF(J LE 1) GOTO 35

DO 30 K = 2, J
B = K

30 C = C + DLOG(B)

35 IF(NRESUP LE 1) GOTO 45

DO 40 K = 2, NRESUP
B = K

40 D = D + DLOG(B)

45 E = A + C - D

CALL FLOW(E, J, IX, IY)
Z = Z + DEXP(E)

A-7
20 CONTINUE
ITOT=NTOT
DO 50 NRESUP=TEST,ITOT
C=D=0.0
M=MO
IF(M.LE.1)GOTO 65
DO 60 K=2,M
B=K
60 C=C+LOG(B)
65 IF(J.LE.1)GOTO 75
DO 70 K=2,J
B=K
70 D=D+LOG(B)
75 A=F=0.0
IF(NRESUP.LE.1)GOTO 85
DO 80 K=2,NRESUP
B=K
80 A=A+LOG(B)
85 N=NTOT-NRESUP
IF(N.LE.1)GOTO 95
DO 90 K=2,N
B=K
90 F=F+LOG(B)
95 G=(NTOT-MO-J)*LOG(MO)
H=(NRESUP-J)*LOG(RO)
E=C+D+B+H-A-F
CALL FLOW(E,J,IX,IY)
Y=Y+DEXP(E)
IF(J.EQ.0) P0 = 1.0/(Y+Z)

IF(J.EQ.0) PRINT503, J, P0

IF(J.EQ.0) GOTO 10

P(J) = 1.0/(Y+Z)

PRINT503, J, P(J)

503 FORMAT(*, P(*, 'I3.*'), '(*, D20.14)

10 CONTINUE

RETURN

END

SUBROUTINE TEN(NTOT, MO, P0, PO, RO, IX, IY)

C FINDS THE PROBABILITIES FOR J=NTOT-MO+1 TO J=NTOT (USES NATURAL DLOGST)

0 ALLOW FOR LARGER NTOT).

DOUBLE PRECISION P(*), NTOT, PO, MO, PO, RO, IX, IY

DOUBLE PRECISION DLOG, DEXP

INITIAL = NTOT-MO+1

ITOT = NTOT

DO 10 J = INITIAL, ITOT

Y = Z = 0.0

DO 20 N1 = 1, INITIAL

NRESUP = N1 - 1

A = C = H = G = 0.0

IF(J.LE.1) GOTO 105

DO 100 K = 2, J

B = K

100 G = G + DLOG(B)

105 CONTINUE

A-9
105 N=NTOT-J
IF(N.LE.1)GOTO 115
DO 110 K=2,N
B=K
110 H=H+DLOG(B)
115 IF(NRESUP.LE.1)GOTO 35
DO 30 K=2,NRESUP
B=K
30 A=A+DLOG(B)
35 N=MO
IF(N.LE.1)GOTO 45
DO 40 K=2,N
B=K
40 C=C+DLOG(B)
45 D=(NRESUP-J)*DLOG(RO)
E=(NRESUP-(NTOT-MO))*DLOG(MO)
F=D+E+G+H=A+C
CALL FLOW(F,J,IX,IY)
Y=Y+DEXP(F)
20 CONTINUE
DO 50 NRESUP=INITAL,ITOT
A=C=D=E=F=0.0
IF(J.LE.1)GOTO65
DO 60 K=2,J
B=K
60 A=A+DLOG(B)
65  I=NTOT-J
   IF(I.LE.1)GOTO 75
   DO 70 K=2,I
   B=K
70  C=C+DLOG(B)
75  I=NTOT-NRESUP
   IF(I.LE.1)GOTO 85
   DO 80 K=2,NRESUP
   B=K
80  E=E+DLOG(E)
85  IF(NRESUP.LE.1)GOTO 95
   DO 90 K=2,NRESUP
   B=K
90  F=F+DLOG(B)
95  D=(NRESUP-J)*DLOG(RO)
    B=D+A+C-E-F
    CALL FLOW(B,J,IX,IY)
    Z=Z+DEXP(B)
50  CONTINUE
    P(J)=1.0/(Y+Z)
PRINT506,J,P(J)
506  FORMAT(*,P(*,I3,*),F*D20.14)
10  CONTINUE
    RETURN
END
SUBROUTINE SIXTEN(NTOT, M1, P, NB)
C FINDS THE DEXPECTED NUMBER OF BACKORDERS.

DOUBLE PRECISION P(101), NB, PO, NTOT, M1, B

ITOT=NTOT

NB=0.

INITAL=NTOT-M1+1

DO 10 NRESUP=INITAL, ITOT

B=(M1-(NTOT-NRESUP))*P(NRESUP)

PRINT20=NRESUP*B

20 FORMAT('NB ', 13.', 13.' = ', D20.14)

10 NB=NB+B

RETURN

END

SUBROUTINE SUB2(NTOT, RO, DER1, DER2)
C FINDS THE RESULTS FOR FORMULA 20.

DOUBLE PRECISION NTOT, RO, DER1, DER2(101)

ITOT=NTOT

DO 10 NRESUP=1, ITOT

10 DER2(NRESUP)=NRESUP/RO-DER1/RO

PODER=DER1/(0.0-RO)

RETURN

END

SUBROUTINE SUB21(NTOT, P, DER1)
C FINDS THE DEXPECTED NUMBER OF UNITS IN RESUPPLY.

DOUBLE PRECISION NTOT, P(101), DER1

DER1=0.0

ITOT=NTOT

DO 10 NRESUP=1, ITOT
DER1 = NRESUP * P(NRESUP) * DER1
RETURN
END

SUBROUTINE SUB22(NTOT, M1, NB, DER1, DER2, RO, NBSQ)
C FINDS THE RESULTS FOR FORMULA 22.
DOUBLE PRECISION NTOT, M1, NB, DER1, DER2, RO, NBSQ
DER2 = NB / RO * ((NTOT - M1) - DER1 + NBSQ / NB)
RETURN
END

SUBROUTINE VAR3CNTOT, M1, P, PO, VAR, SN)
DOUBLE PRECISION NTOT, M1, P, PO, VAR, SN
ITEST = NTOT - M1
VAR = PO * (NTOT - M1 - SN) ** 2
DO 10 NRESUP = 1, ITEST
10 VAR = VAR + P(NRESUP) * (NTOT - M1 - NRESUP - SN) ** 2
RETURN
END

SUBROUTINE SNAVECNTOT, M1, P, PO, SN)
DOUBLE PRECISION NTOT, M1, P, PO, SN
ITEST = NTOT - M1
SN = PO * (NTOT - M1)
DO 10 NRESUP = 1, ITEST
SN = SN + P(NRESUP) * (NTOT - M1 - NRESUP)
10 CONTINUE
RETURN
END
SUBROUTINE MNAVE(NTOT, M1, P, PO, MN)

DOUBLE PRECISION NTOT, M1, P(101), PO, MN

INITIAL = NTOT - M1

MN = 0

IF(INITIAL .EQ. 0) MN = PO*(NTOT)

ITOT = NTOT

IF(INITIAL .EQ. 0) INITIAL = 1

DO 10 NRESUP = INITIAL, ITOT

10 MN = MN + (NTOT - NRESUP) * P(NRESUP)

MN = MN + M1 * PO

TEST = INITIAL = 1

DO 20 NRESUP = INITIAL, ITOT

20 MN = MN + M1 * P(NRESUP)

RETURN

END

SUBROUTINE VAR4(NTOT, M1, MN, P, VAR, PO)

DOUBLE PRECISION NTOT, M1, MN, P(101), VAR, PO

INITIAL = NTOT - M1

TEST = INITIAL + 1

VAR = PO

DO 20 NRESUP = INITIAL, ITOT

20 VAR = VAR + P(NRESUP)

VAR = VAR + (M1 - MN)**2

IF(INITIAL .EQ. 0) VAR = VAR + PO*(NTOT - MN)**2

IF(INITIAL .EQ. 0) INITIAL = 1

ITOT = NTOT

DO 10 NRESUP = INITIAL, ITOT

RETURN

END
10 VAR = VAR + P(NRESUP)* (NTOT - NRESUP - MN)**2
RETURN
END

SUBROUTINE FLOW(X, J, IX, IY)
C CHECKS FOR OVERFLOW AND UNDERFLOW -- PRINTS WARNING MESSAGE IF ONE IS FOUND.
DOUBLE PRECISION X
IF(X LT 174.) IX = IX + 1
IF(X GT 174.) IY = IY + 1
IF(X LT 174.) PRINT5, J
5 FORMAT('****** UNDERFLOW ***** J=', I3)
IF(X LT 174.) X = -174.
IF(X GT 174.) PRINT10, J
10 FORMAT('****** OVERFLOW ***** J=', I3)
IF(X GT 174.) X = 174.
C SETS X EQUAL TO MAX OR MIN VALUE SO OVERFLOW OR UNDERFLOW DOESN'T OCCUR IN EXPONENT.
RETURN
END

SUBROUTINE VAR(X, M1, NTOT, VAR, P, NB)
C FINDS THE VARIANCE OF NB OR IF NB IS SET EQUAL TO 0, THE SUM OF THE SQUARES OF THE NB SUBVALUES IS FOUND.
DOUBLE PRECISION P(101), PO, M1, NTOT, VAR, NB
ITOT = NTOT
INITIAL = NTOT - M1 + 1
VAR = 0.0
IF(INITIAL.EQ.0) VAR = VAR + PO*(M1-NTOT-NB)**2
IF(INITIAL.EQ.0) INITIAL = 1
DO 10 NRESUP = INITIAL, ITOT
10 VAR = VAR + P(NRESUP)*(M1-NTOT+NRESUP-NB)**2
RETURN
END

SUBROUTINE VAR2C(DE1, P, VAR, PO, NTOT)
DOUBLE PRECISION PC, DER1, VAR, PO, NTOT
ITOT = NTOT
VAR = PO*DER1**2
DO 10 NRESUP = 1, ITOT
10 VAR = VAR + P(NRESUP)*(NRESUP-DER1)**2
RETURN
END

ENTRY
20 .5 15 17
/LOGOFF
POISSON MODEL
(INFINITE POPULATION)
PROGRAM LISTING
/LOGON "NC"

/DO RUN WATFIV

$JOB (L8NTJQQ) KOVACS

DOUBLE PRECISION P(101), PA(101), DER2(101), NE, NB, PO, PO1, STOCK, STO

1K1, NTOT, RO, M1, MO, ANS19, DER1, DER3, VAR, VARA, NBSQ, VARB, RESPSQ, VARS, R

2, MN, VARM, U, SSSQ, MNSQ, T

READ NTOT, RO, MO, M1

ITOT = NTOT

CALL PVALL(P, NTOT, MO, RO, PO)

CALL NBSTK(NB, STOCK, NTOT, M1, MO, RO, PO)

CALL PVALL(PA, NTOT+1, MO, RO, PO1)

CALL NBSTK(NB1, STOCK1, NTOT+1, M1, MO, RO, PO1)

PRINT 142, NE = NB1

CALL NINTEN(NTOT, M1, NE1, NB, PO1, PO, ANS19, MO, PA, RO)

142 FORMAT(*, DELTA NE =*, D20.14)

PRINT400, ITOT, ITOT+1, ITOT, ANS19

400 FORMAT(*, NB(*) = NE(*) - NB(*) / NB(*) = *D20.14)

PRINT304, (NE-NE1)/NB

304 FORMAT(*, THE LEFT SIDE OF THE EQUATION GIVES *D20.14)

CALL SUB21(NTOT, PA, DER1)

PRINT310, DER1

310 FORMAT(*, THE EXPECTED NUMBER OF UNITS IN RESUPPLY IS *D20.14)

PRINT320, DER1/(-RO)

320 FORMAT(*, THE RESULT OF FORMULA 21 IS *D24.14)

CALL SUB20(NTOT, RO, DER1, DER2)

PRINT333

PRINT330, 0, DER1/(-RO)
333 FORMAT('CANSWERS FOR FORMULA 20\n')
330 FORMAT(* P(*.13*)) GIVES * D24.14)
PRINT330,(J6,DER2(J6),J6=1,ITOT)
CALL VAR1(PO,M1,NTOT,VAR,P,NB)
CALL VAR1(PO,M1,NTOT+1,VAR,PA,NB)
PRINT350,1TOT+1,VARA
350 FORMAT(' THE VARIANCE OF NB FOR NTOT=*13.1S,* D24.14)
PRINT350,1TOT+1,VARA
DER3=0.0
CALL VAR1(PO,M1,NTOT,NBSQ,P,DER3)
CALL SUB22(NTOT,M1,NE,DER1,DER3,RO,NBSQ)
PRINT360,DER3
360 FORMAT(' THE ANS\n')
PRINT352,NBSQ
CALL VAR2(DER3,P,VARN,PO,NTOT)
352 FORMAT(* THE EXPECTED VALUE OF THE SQUARE OF THE BACKORDER LEVEL
1S* D20.14)
PRINT410,VARN
410 FORMAT(* THE VARIANCE OF THE EXPECTED NUMBER OF UNITS IN RESUPPLY
1S* D20.14)
DER3=0.0
CALL VAR2(DER3,P,RESPSQ,PO,NTOT)
PRINT420,RESPSQ
420 FORMAT(* THE EXPECTED VALUE OF THE SQUARE OF THE NUMBER OF UNITS IN RESUPPLY
1S* D20.14)
CALL SNAVE(NTOT,M1,P,PO,SN)
PRINT430,SN
A-19
430 FORMAT('THE EXPECTED STOCK LEVEL IS *D21.14)

CALL VAR3(NTOT,M1,P,PO,VARS,SN)

PRINT440,VARS

440 FORMAT('THE VARIANCE OF THE STOCK LEVEL IS *D20.14)

U=0.0

CALL VAR3(NTOT,M1,P,PO,SSQ,U)

425 FORMAT('THE EXPECTED VALUE OF THE SQUARE OF THE STOCK LEVEL IS *D20.14)

PRINT425,SSQ

CALL MNAVE(NTOT,M1,P,PO,HN)

PRINT435,MN

435 FORMAT('THE EXPECTED NUMBER IN SERVICE IS *D20.14)

CALL VAR4(NTOT,M1,MN,P,VARM,PO)

PRINT444,VARM

444 FORMAT('THE VARIANCE OF THE EXPECTED NUMBER IN SERVICE IS *D20.14)

CALL VAR4(NTOT,M1,U,P,MNSQ,PO)

PRINT446,MNSQ

446 FORMAT('THE EXPECTED VALUE OF THE SQUARE OF THE NUMBER IN SERVICE IS *D20.14)

STOP

END

SUBROUTINE NINTEN(NTOT,M1,NB1,NB,PO1,PO,ANS19,MO,PA,RO)

DOUBLE PRECISION NTOT,NBSQ,M1,NB1,NB,PO1,PO,ANS19,MO,PA(101),RO

ANS19=0.0

CALL VAR1(PO1,M1,NTOT+1,NBSQ,PA,ANS19)

ANS19=1.0-(1.0/(NTOT-M1+1))*(MO*RO*PO1/PO)-NBSQ/NB)
SUBROUTINE SUB2CNTOT, RO, DER1, DER2)
C FINDS THE RESULTS FOR FORMULA 20.
DOUBLE PRECISION NTOT, RO, DER1, DER2(101)
ITOT=NTOT
DO 10 NRESUP=1, ITOT
10 DER2(NRESUP)=NRESUP/RO-DER1/RO
PODER=DER1/(0.0+RO)
RETURN
END

SUBROUTINE SUB21CNTOT, P, DER1)
C FINDS THE EXPECTED NUMBER OF UNITS IN RESUPPLY.
DOUBLE PRECISION NTOT, P(101), DER1
DER1=0.0
ITOT=NTOT
DO 10 NRESUP=1, ITOT
10 DER1=NRESUP*P(NRESUP)+DER1
RETURN
END

SUBROUTINE SUB22CNTQT, H1, NB, DER1, DER2, RO, NBSQ)
C FINDS THE RESULTS FOR FORMULA 22.
DOUBLE PRECISION NTOT, H1, NB, DER1, DER2, RO, NBSQ
DER2=NB/RO*((NTOT-M1)-DER1+NBSQ/NB)
RETURN
END
SUBROUTINE VAR3(NTOT, M1, P, PO, VAR, SN)
DOUBLE PRECISION NTOT, M1, P, PO, VAR, SN
ITEST = NTOT - M1
VAR = PO * (NTOT - M1 - SN)**2
DO 10 NRESUP = 1, ITEST
10 VAR = VAR + P(NRESUP) * (NTOT - M1 - NRESUP - SN)**2
ITOT = NTOT
INITIAL = ITEST + 1
DO 30 NRESUP = INITIAL, ITOT
30 VAR = VAR + P(NRESUP) * SN**2
RETURN
END

SUBROUTINE SNAVEC(NTOT, M1, P, PO, SN)
DOUBLE PRECISION NTOT, M1, P, PO, SN
ITEST = NTOT - M1
SN = PO * (NTOT - M1)
IF(ITEST .LT. 1) GOTO 20
DO 10 NRESUP = 1, ITEST
SN = SN + P(NRESUP) * (NTOT - M1 - NRESUP)
10 CONTINUE
20 RETURN
END

SUBROUTINE MNAVE(NTOT, M1, P, PO, MN)
DOUBLE PRECISION NTOT, M1, P, PO, MN
INITIAL = NTOT - M1 + 1
MN = 0
IF(INITIAL .LE. 0) MN = PO * NTOT
ITOT=NTOT

IF(INITIAL.LE.0) INITIAL=1

DO 10 NRESUP=INITAL,ITOT

10 MN=MN+(NTOT-NRESUP)*P(NRESUP)

MN=MN+M1*PO

ITEST=INITAL+1

IF(ITEST.LT.1)GOTO 30

DO 20 NRESUP=1,ITEST

20 MN=MN+M1*P(NRESUP)

30 RETURN

END

SUBROUTINE VAR4CNTOT,M1,MN,P,VAR,PO)

DOUBLE PRECISION NTOT,M1,MN,P(101),VAR,PO

INITIAL=NTOT-M1

ITEST=INITAL+1

VAR=PO

DO 20 NRESUP=1,ITEST

20 VAR=VAR+P(NRESUP)

VAR=VAR*(M1-MN)**2

IF(INITAL.EQ.0)VAR=VAR+PO*(NTOT-MN)**2

IF(INITAL.EQ.0)INITIAL=1

ITOT=NTOT

DO 10 NRESUP=INITAL,ITOT

10 VAR=VAR+P(NRESUP)*(NTOT-NRESUP-MN)**2

RETURN

END
SUBROUTINE FLOW(X, J, IX, IY)
C CHECKS FOR OVERFLOW AND UNDERFLOW -- PRINTS WARNING MESSAGE IF ONE IS FOUND.

DOUBLE PRECISION X

IF(X.LT.-174.9) IX = IX + 1
IF(X.GT.174.9) IY = IY + 1
IF(X.LT.-174.9) PRINTS, J
IF(X.LT.-174.9) X = -174.9

5 FORMAT('**** UNDERFLOW***** J=', J, 'I3)
IF(X.GT.174.9) PRINT10, J

10 FORMAT('**** OVERFLOW***** J=', J, 'I3)
IF(X.GT.174.9) X = 174.9

RETURN

END

SUBROUTINE VAR1(PO, M1, NTO, T, VAR, P, NB)
DOUBLE PRECISION P(101), PO, M1, NTO, T, VAR, NB

ITOT = NTO

INITAL = NTO + M1 + 1
VAR = 0.0

IF(INITAL.EQ.0) VAR = VAR + PO*(M1-NTO-NB)**2
IF(INITAL.EQ.0) INITAL = 1

DO 10 NRESUP = INITAL, ITOT

10 VAR = VAR + P(NRESUP)*(M1-NTO+NRESUP-NB)**2

ITEST = NTO + M1

DO 20 NRESUP = 1, ITEST

A-24
20 VAR = VAR + NB ** 2 * P(NRESUP)
VAR = VAR + PO * NB ** 2
RETURN
END

SUBROUTINE VAR2(DER1, P, VAR, PO, NTOT)
DOUBLE PRECISION P(101), DER1, VAR, PO, NTOT
ITOT = NTOT
VAR = PO * DER1 ** 2
DO 10 NRESUP = 1, ITOT
10 VAR = VAR + P(NRESUP) * (NRESUP - DER1 ** 2)
RETURN
END

SUBROUTINE PVALC(PNTQT, MO, RO, P0)
DOUBLE PRECISION DLOG, DEXP, PO, MO, RO, X, C, B, A, P(101), NTOT
ITEST = NTOT
PO = DEXP(-MO * RO)
X = PO
DO 10 NRESUP = 1, ITEST
B = 0.0
DO 30 J = 1, NRESUP
A = J
30 B = B + DLOG(A)
P(NRESUP) = (MO * RO) ** NRESUP * DEXP(-MO * RO) / DEXP(B)
X = X + P(NRESUP)
PRINT20, NRESUP, P(NRESUP)
10 CONTINUE
20 FORMAT(*,F(*,13d) = *,D20.14)
PRINT40,X
40 FORMAT(*,THE TOTAL OF THE PossION PROBABILITIES IS*,D20.14)
RETURN
END

SUBROUTINE NBSTK(NB,STOCK,NTOT,MO,MI,RO,PO)
DOUBLE PRECISION DLOG,DEXP,NTOT,MI,MO,STOCK,MO,RO,PO
M=NTOT-MI+1
ITEST=100
NB=STOCK=O.O
IF(M.LE.O)NB=PO
IF(M.LE.O)M=1
DO 50 NRESUP=M,ITEST
C=O.O
IF(NRESUP.LE.1)GOTO 65
DO 60 J=2,NRESUP
D=J
60 C=C+DLOG(D)
65 IF(NRESUP+LOG(MO*RO)-C-MO*RO*LT<10021.100)GOTO111
5O NB=NB+(M1-NTOT+MRESUP)*DEXP(NRESUP+LOG(MO*RO)-C)*DEXP(-MO*RO)
111 CONTINUE
PRINT100,NB
100 FORMAT(*,OTHER ARE*,D20.14*,BACKORDERS*)
RETURN
END

ENTRY
20 .2 15 17
/LLOGOFF
CONSERVATIVE SINGLE SERVER
INVENTORY MODEL
(FINITE POPULATION)
PROGRAM LISTING
/LOGON "NC"
/DO RUN.WATFIV
$JOB (L8NTJQQ) KOVACS

DOUBLE PRECISION P(101),PA(101),DER2(101),NE,NE1,PO,PO1,Z,NTOT,RO
IM1,MO,ANS36,DER1,DER3,VARA,NBSQ,VARN,RESFSQ,VARS,USQ,MONSQ.

Z=0.0
PRINT5
READK1,K2,K3,K4
5 FORMAT("10.5X","CONSERVATIVE SINGLE SERVER INVENTORY MODEL ")
READMIN,MAX,RO,MO,M1
PRINT10,MIN,MAX,RO,MO,M1
M1=M1+D20.14)
DO 100 ITOT=M1
NTOT=ITOT
IF(K4.EQ.1)PRINT150
IX=IY=0
IF(MO.EQ.RO.LE.1.0)CALL PVAL(P,NTOT,MO,RO,PO,K2)
IF(MO.EQ.RO.GT.1.0)CALL PVAL1A(P,NTOT,MO,RO,PO,IX,IY,K2)
C IX AND IY ARE TOTALS FOR UNDER AND OVERFLOWS IN THE PROGRAM.
150 FORMAT("CP(N) IS THE PROBABILITY THAT N ITEMS ARE IN RESUPPLY AT 
TIME NEARS INFINITY.")
C T IS THE TOTAL OF THE PROBABILITIES -- SHOULD EQUAL 1.000.
T=0.
DO 100 I=1,ITOT

A-28
100 T=T+P(I)
   T=T+PQ
   IF(K2.EQ.1)PRINT160,T
160 FORMAT('THE TOTAL OF ALL THE P VALUES IS ',D20.14)
   CALL_SUB17(NTOT,M1,P,K1)
   CALL_SUB18(NTOT,M1,MO,RO,NE,K1)
   I6=NTOT-M1+1
   Z=0.0
   DO 300 I7=I6,ITOT
300 Z=Z+P(I7)
   C ZIS THE TOTAL OF ALL THE PROBABILITIES FROM I=NTOT-M1+1 TO I=NTOT
   IF(K1.EQ.1)PRINT301,Z
301 FORMAT('THE PROBABILITY FOR STOCKOUT IS ',D20.14)
   CALL_STOCK(MO,RO,M1,NTOT,K1)
   IX=IY=0
   K5=0
   IF(MO*RO.LE.1.0)CALL_PVAL(PA,NTOT+1,MO,RO,PO,I,K5)
   IF(MO*RO.GT.1.0)CALL_PVAL1A(PA,NTOT+1,MO,RO,PO,IX,IY,K5)
   I79=ITOT+1
   PRINT220,IX,IY
220 FORMAT('THERE WERE ',I3, ' UNDERFLOWS AND ',I3, ' OVERFLOWS')
   T=0.0
   DO 102 I1=I79
102 T=T+PA(I)
   CALL_SUB16(NTOT+1,M1,MO,RO,NE,K5)
   CALL_SUB36(MO,RO,PO,PO1,ANS36)
   IF(K4.EQ.1)PRINT400,ITOT,ITOT+1,ITOT,ANS36
400 FORMAT('ONBC ',I3.') = NB(',I3.') / NB(',I3.') = ',D20.14)
        IF(K4.EQ.1)PRINT304,(NB-NB1)/NB
304 FORMAT('THE LEFT SIDE OF THE EQUATION GIVES ',D20.14)

        CALL SUB38(NTOT,P,DER1)
        IF(K3.EQ.1)PRINT310,DER1
310 FORMAT('THE EXPECTED NUMBER OF UNITS IN RESUPPLY IS ',D20.14)
        IF(K4.EQ.1)PRINT320,-DER1/RO
320 FORMAT('THE RESULT OF FORMULA 36 IS ',D20.14)
        CALL SUB37(NTOT,RO,DER1,DER2)
        IF(K4.EQ.1)PRINT333
333 FORMAT('THE ANSWERS FOR FORMULA 37: ')
        IF(K3.EQ.1)PRINT330,DER1/RO
400 FORMAT('THE ANSWERS FOR FORMULA 37: ')
        IF(K4.EQ.1)PRINT330,(J5,DER2,J5,J5=1,NTOT)
        CALL VAR1(PO,M1,NTOT,VAR,P,NB)
        IF(K3.EQ.1)PRINT350,NTOT,VAR
350 FORMAT(' THE VARIANCE OF NB FOR NTOT= ',I3.', IS ',D20.14)
        DER3=0.0
        CALL VAR1(FO,M1,NTOT,NBSQ,P,DER3)
        CALL SUB39(NTOT,M1,NB,DER1,DER3,RO,NBSQ)
        IF(K4.EQ.1)PRINT360,DER3
360 FORMAT(' THE ANSWER TO FORMULA 39 IS ',D20.14)
        IF(K3.EQ.1)PRINT352,NBSQ
352 FORMAT(' THE EXPECTED VALUE OF THE SQUARE OF THE BACKORDER LEVEL IS ',D20.14)
        IF(K3.EQ.1)PRINT410,VARN
410 FORMAT(*THE VARIANCE OF THE EXPECTED NUMBER OF UNITS IN RESUPPLY
  "D20.14)
  DER3=0.
  CALL VAR2(DER3,P,RESPSQ,PO,NTOT)
  IF(K3.EQ.1)PRINT420,RESPSQ

420 FORMAT(*THE EXPECTED VALUE OF THE SQUARE OF THE NUMBER OF UNITS IN
  RESUPPLY IS "D20.14)
  CALL SNAVEC(NTOT,M1,P,PO,SN)
  IF(K3.EQ.1)PRINT430,SN

430 FORMAT(*THE EXPECTED STOCK LEVEL IS "D20.14)
  CALL VAR3(NTOT,M1,P,PO,VARS,SN)
  IF(K3.EQ.1)PRINT440,VARS

440 FORMAT(*THE VARIANCE OF THE STOCK LEVEL IS "D20.14)
  U=0.
  CALL VAR3(NTOT,M1,P,PO,SSQ,U)

425 FORMAT(*THE EXPECTED VALUE OF THE SQUARE OF THE STOCK LEVEL IS 
  "D20.14)
  IF(K3.EQ.1)PRINT425,SSQ
  CALL MNAVEC(NTOT,M1,P,PO,MN)
  IF(K3.EQ.1)PRINT435,MN

435 FORMAT(*THE EXPECTED NUMBER IN SERVICE IS "D20.14)
  CALL VAR4(NTOT,M1,MN,P,VARM,PO)
  IF(K3.EQ.1)PRINT444,VARM

444 FORMAT(*THE VARIANCE OF THE EXPECTED NUMBER IN SERVICE IS "D20.14)
  CALL VAR4(NTOT,M1,U,P,MNSQ,PO)
  IF(K3.EQ.1)PRINT446,MNSQ
446  FORMAT(*THE EXPECTED VALUE OF THE SQUARE OF THE NUMBER IN SERVIE

1  1SI*,D20.14)

1000 CONTINUE

STOP

END

SUBROUTINE PVAL(P,NTOT,MO,RO,PO,K2)

DOUBLE PRECISION P,NTOT,MO,RO,PO,A,B,C,D,E

DOUBLE PRECISION DLOG,DEXP

I = NTOT + 1

DO 5 JK = 1, ITEST

NRESUP = JK - 1

CALL GMORO(MO,RO,G)

M = MO

CALL FACT(M,C)

A = -MO*DLOG(MO) + C

B = NTOT*DLOG(MO*RO)

A = A + B + G

B = 0.0

I = NTOT - MO + 1

DO 10 MS = 1, I

J = MS - 1

10  B = B + DEXP(J*DLOG(MO*RO))

A = -(DLOG(DEXP(A) + B))

IX = IY = 0

IF(NRESUP.EQ.0.0)CALL FLOW(A,NRESUP,IX,IY)

IF(NRESUP.EQ.0.0)PO = DEXP(A)

IF(NRESUP.EQ.0.0 AND K2.EQ.1)PRINT101,NRESUP,PO
IF(NRESUP.EQ.0.0)GOTO 5
IF(NRESUP.LE.NTOT-MO)P(NRESUP)=DEXP(NRESUP-DLOG(MO-RO))+A
D=(NTOT-MO)*DLOG(MO)
M9=NTOT-NRESUP
G=0.0
CALL FACT(M9,G)
E=NRESUP*DLOG(RO)
IF(NRESUP.GT.NTOT-MO)P(NRESUP)=DEXP(C-G+D+E+A)
IF(K2.EQ.1)PRINT101,NRESUP,P(NRESUP)
5 CONTINUE
101 FORMAT(*.P(*13s*) = *D20.14*)
RETURN
END
SUBROUTINE SUB17(NTOT,M1,P,K1)
DOUBLE PRECISION NTOT,M1,P(101),NE,X
NB=0.0
M=M1
DO 10 J=1,M
X=J-1
I=NTOT-X
NB=NB+(M1-X)*P(I)
10 CONTINUE
IF(K1.EQ.1)PRINT4C,NE
40 FORMAT(*OFORMULA 17 GIVES*D20.14s*.BACKORDERS*)
RETURN
END
SUBROUTINE SUB36(MO, RO, PO, P0, ANS36)
    DOUBLE PRECISION MO, RO, PO, P0, ANS36
    ANS36 = 1.0 - MO*RO*PO1/PO
    RETURN
END

SUBROUTINE SUB37(NTOT, RO, DER1, DER2)
    DOUBLE PRECISION NTOT, RO, DER1, DER2(101)
    ITOT = NTOT
    DO 10 NRESUP = 1, ITOT
       10 DER2(NRESUP) = NRESUP/RO - DER1/RO
    PODER = DER1/(0.0 - RO)
    RETURN
END

SUBROUTINE SUB38(NTOT, P, DER1)
    DOUBLE PRECISION P(101), NTOT, DER1
    DER1 = 0.0
    ITOT = NTOT
    DO 10 NRESUP = 1, ITOT
       10 DER1 = NRESUP*P(NRESUP) + DER1
    RETURN
END

SUBROUTINE SUB39(NTOT, M1, NB, DER1, DER2, RO, NBSQ)
    DOUBLE PRECISION NTOT, M1, NB, DER1, DER2, RO, NBSQ
    DER2 = NB/RO*(NTOT-M1)-DER1+NBSQ/NB
    RETURN
END
SUBROUTINE VAR3(NTOT,M1,P,PO,VAR,SN)
DOUBLE PRECISION NTOT,M1,P(101),PO,VAR,SN
ITEST=NTOT-M1
VAR=PO*(NTOT-M1-SN)**2
DO 10 NRESUP=1,ITEST
10 VAR=VAR+P(NRESUP)*(N-M1-NRESUP-SN)**2
ITOT=NTOT
INITAL=ITEST+1
DO 30 NRESUP=INITAL,ITOT
30 VAR=VAR+P(NRESUP)**2
RETURN
END

SUBROUTINE SNAVE(NTOT,M1,P,PO,SN)
DOUBLE PRECISION NTOT,M1,P(101),PO,SN
ITEST=NTOT-M1
SN=PO*(NTOT-M1)
IF(ITEST.LT.1)GOTO 20
DO 10 NRESUP=1,ITEST
SN=SN+P(NRESUP)**2(NTOT-M1-NRESUP)
10 CONTINUE
20 RETURN
END

SUBROUTINE MNAVE(NTOT,M1,P,PO,MN)
DOUBLE PRECISION NTOT,M1,P(101),PO,MN
INITAL=NTOT-M1+1
MN=0
IF(INITAL.EQ.0)MN=PO*(NTOT)
ITOT = NTOT

IF(INITAL .EQ. 0) INITIAL = 1

DO 10 NRESUP = INITIAL, ITOT

10 MN = MN + (NTOT - NRESUP) * P(NRESUP)

MN = MN + M1 * PO

I TEST = INITIAL = 1

IF (I TEST .LT. 1) GOTO 30

DO 20 NRESUP = 1, I TEST

20 MN = MN + M1 * P(NRESUP)

30 RETURN

END

SUBROUTINE VAR4(NTOT, M1, MN, P, VAR, PO)

DOUBLE PRECISION NTOT, M1, MN, P(10), VAR, PO

INITAL = NTOT - M1 + 1

I TEST = INITIAL = 1

VAR = PO

DO 20 NRESUP = 1, I TEST

20 VAR = VAR + P(NRESUP)

VAR = VAR * (M1 - MN)**2

IF(INITAL .EQ. 0) VAR = VAR * PO * (NTOT - MN)**2

IF(INITAL .EQ. 0) INITIAL = 1

ITOT = NTOT

DO 10 NRESUP = INITIAL, ITOT

10 VAR = VAR + P(NRESUP) * (NTOT - NRESUP - MN)**2

RETURN

END
SUBROUTINE FLOW(X,J,IX,IY)
DOUBLE PRECISION X
IF(X.LT.-174.9)IX=IX+1
IF(X.GT.174.9)IY=IY+1
IF(X.LT.-174.9)PRINT5,J
5 FORMAT('****UNDERFLOW**** J=',I3)
IF(X.GT.174.9)PRINT10,J
10 FORMAT('****OVERFLOW**** J=',I3)
IF(X.LT.-174.9)X=-174.9
IF(X.GT.174.9)X=174.9
RETURN
END

SUBROUTINE VAR1CPO,J1,TOT,VAR,P,NB)
DOUBLE PRECISION P(101),PO,M1,TOT,TOT,VAR,NB
INITAL=NTOT=M1+1
VAR=0.0
IF(INITIAL.EQ.0)VAR=VAR+PO*(M1-NTOT-NB)**2
IF(INITIAL.EQ.0)INITAL=1
DO 10 NRESUP=INITAL,ITOT
10 VAR=VAR+P(NRESUP)*(M1-NTOT+NRESUP-NB)**2
ITEST=NTOT=M1
DO 20 NRESUP=1,ITEST
20 VAR=VAR+NE**2*P(NRESUP)
VAR=VAR+PO*NE**2
RETURN
END
SUBROUTINE VAR2(P, DER1, VAR, PQ, NTOT)

DOUBLE PRECISION P, DER1, VAR, PQ, NTOT

ITOT = NTOT

VAR = DER1 * 2

DO 10 NRESUP = 1, ITOT

10 VAR = VAR + P(NRESUP) * (NRESUP - DER1)**2

RETURN

END

SUBROUTINE SUBV19(MO, RO, NTOT, M1, ANS19)

DOUBLE PRECISION MO, RO, NE, M1, NTOT, ANS19, G, BA

DOUBLE PRECISION LOG, EXP

CALL GMOR(MO - 1.0, RO, G)

G = EXP(G)

M = MO

IF(MO = 1.0, LE, 0.0) G = 0.0

CALL FACT(M, A)

B = MN - NB - 1.0 / RO

IF(B = LE, 1.0, AND, B = GT, 0.0) B = 0.0

A = ((1.0 - MO * RO) * EXP(-MO * LOG(MO) + A) * G + (M1 - MO * RO) * (M1 - NB)) * EXP

1((1.0 - MO) * LOG(MO) + (1.0 - MO) * LOG(MO)) + (1.0 - MO) * EXP(-MO * LOG(MO) + (1.0 - MO) * LOG(MO)) * EXP(-M1)

LOG(MO * RO) / (1.0 - MO * RO) - (M1 - MO) * EXP(-MO * LOG(MO) + (1.0 - MO) * LOG(MO))

ANS19 = (LOG(A) - LOG(NB)) / (-LOG(MO * RO))

RETURN

END

SUBROUTINE FACT(M, A)

DOUBLE PRECISION A, B

DOUBLE PRECISION LOG

A-38
A=0.0
IF(M.LE.1)GOTO 20
DO 10 J=2,M
B=J
10 A=A+DLOG(B)
20 RETURN
END

SUBROUTINE GMORO(MO,RO,A)
DOUBLE PRECISION MO,RO,A,B,DLOG,DEXP
INTEGER X
M=MO
A=0.0
DO 10 J=1,M
X=J-1
B=0.0
CALL FACT(X,B)
10 A=A+DEXP(-X*DLOG(RO)-B)
A=DLOG(A)
RETURN
END

SUBROUTINE SUB21(MO,RO,NB,M1)
DOUBLE PRECISION MO,RO,NB,M1,M2,A,B,ANS
DOUBLE PRECISION DLOG,DEXP
IF(MO.NE.M1)GOTO 20
CALL GMORO(MO-1,0,RO,0)
G=DEXP(G)
IF(MO-1.GE.LE.0)G=0.0
M=MO
CALL FACT(M0,A)
B=M1-NB-1.0/RO
IF(B.LT.1E-12.AND.B.GT.0.0)B=0.0
A=((MO-(MO+RO)+...-NB)*DEXP(1.0-MO)*DLG(MO+RO))+(DEXP(-MO-DLG)
1(MO)+A)*G((1.0-MO+RO)*B)
ANS=(DLG(NB)-DLG(A))/DLG(MO+RO))
PRINT10,ANS
10 FORMAT(1OFORMULA 21 GIVES D20.14)
20 RETURN
END

SUBROUTINE PAL1AC(P,NTOT,MO,RO,P0,IX,IY,X2)
DOUBLE PRECISION P,NTOT,MO,RO,P0,IX,IY,X2
DOUBLE PRECISION DLG,DEXP
ITEST=NTOT+1
DO 5 JK=1,ITEST
NRESUP=JK-1
CALL GMORO(MO,RO,P0)
M=MO
CALL FACT(M0,C)
A=-MO*DLG(MO)+C+8
B=0.0
I=NTOT-MO+1
IF(I.LE.0)GOTO 20
DO 10 MS=1,i
J=MS-1
10 B=B+DEXP((J-NTOT)*DLG(MO+RO))
20  A = -(DLOG(DEXP(A)+B))

IF(NRESUP.EQ.0.0) CALL FLOW(A,NRESUP,IX,IY)

IF(NRESUP.EQ.0.0) D = DEXP(-NTOT*DLOG(MO*RO)+A)

IF(NRESUP.EQ.0.0) IF(K2.EQ.1) PRINT101,NRESUP,PO

IF(NRESUP.EQ.0.0) GOTO 5

IF(NRESUP.LE.NTOT-MO) P = DEXP((NRESUP-NTOT)*DLOG(MO*RO)+A)

D = MO*DLOG(MO)

M9 = NTOT-NRESUP

F = 0.0

CALL FACT(M9,F)

E = (NRESUP-NTOT)*DLOG(RO)

IF(NRESUP.GT.NTOT-MO) P = DEXP(C+F+D+E+A)

IF(K2.EQ.1) PRINT101,NRESUP,P(NRESUP)

5 CONTINUE

101 FORMAT(*,P(*,13*),=,*D20.14)

RETURN

END

SUBROUTINE SUB13CNTOT,M1,MO,RO,NB,K1)

DOUBLE PRECISION NTOT,M1,MO,RO,NB,K1,GM,AM,GAMMA,BETA,AS,BS,CS,AS,BS,CS,AS,BS,CS

M = MO

G = 0.0

IF(MO.LE.1) GOTO 5

CALL GM0*(MO-1,RO,G)

G = DEXP(G)

A-41
5 CALL FACT(M, U)

A = (M1 - 1.0/R0) * DEXP(-MO * DLOG(MO) + U) = 0
B = DEXP(DLOG(M1) + (1.0 - MO) * DLOG(MO - R0))

GAMMA = M1 - MO
BETA = 1.0 - MO * R0
C = 0.0
M9 = GAMMA - 1

IF (GAMMA LT 2.0) GOTO 20
DO 10 J = 1, M9
K = J - 1
CALL FACT(M9 + 2, D)

CALL FACT(K, E)

J5 = GAMMA - K - 1
CALL FACT(J5, F)

H = K + 2

10 C = C + (-1)**(K + 1) * BETA**2 * DEXP(E - F - DLOG(H))

20 R = 0.0

IF (BETA EQ 0.0 OR GAMMA EQ 0.0) GOTO 25
R = (-1)**(M9 + 2) * BETA**2 * M9 * GAMMA

25 IF (M1 EQ 1.0) Q = C + R
IF (M1 NE 1.0) Q = DEXP((1.0 - M1) * DLOG(MO - R0)) *(C + R)
J26 = NTOT - MO + 1

S = 0.0

DO 30 J27 = 1, J26
J = J27 - 1

30 S = S * DEXP((J - NTOT)* DLOG(MO - R0))

CALL GMORO(MO, RO, G1)

T = DEXP(-MO * DLOG(MO) + U + G1)
NB=DEXP(DLOG(A+B)+Q)-DLOG(S+T))

IF(K1.EQ.1)PRINT50,NB

IF(K1.EQ.1)PRINT70+DEXP(DLOG(A+B)+Q)-DLOG(T))

50 FORMAT(*'FORMULA 18 GIVES: D20.14: BACKORDERS')

70 FORMAT(*'FOR N=INFINITY, NB = D20.14')

RETURN

END

SUBROUTINE STOCK(MO,RO,M1,NTOT,K1)

DOUBLE PRECISION MO,RO,M1,NTOT,A,B,C,D,E,F,STOK

DOUBLE PRECISION DLOG,DEXP

M=MO

CALL FACT(M,A)

CALL GMORO(MO,RO,G)

B=DEXP(-MO*DLOG(MO)+A+G)

C=0.0

I=M1-MO

IF(I.LT.1)GOTO 20

DO 10 N=1,I

J=N-1

IF(M1-J.EQ.1.0)C=C+1

IF(M1-J.EQ.1.0)GOTO 10

C=C+DEXP((J-M1+1)*DLOG(MO+RO))

10 CONTINUE

20 DO 30 N=1,MO

DO 30 N=1,MO

J=N-1

30 CONTINUE

DO 30 N=1,MO

RETURN

END
30 D=DEXP(J*LOG(MO/RO))

E=DEXP(NTOT*LOG(MO/RO))

F=DEXP(-NTOT*LOG(MO/RO))

IF(MO/RO.LE.1)STOK=DEXP(LOG(E+C)-LOG(D+F+B))

IF(MO/RO.GT.1.0)STOK=DEXP(LOG(E*C)+LOG(D+B+E))

IF(K1.EQ.1)PRINT50,STOK

50 FORMAT(90,THE STOCKOUT PROBABILITY IS*.D20.14)

RETURN

END

ENTRY

1111

1015.51113

/LOGOFF
PROBLEM 1
PROGRAM LISTING
DOUBLE PRECISION N, NBO, U, L

INTEGER DNL1, DNL2
READ RO, M1, MO, NBO, NMAX, A21
NMIN = 1
N = NMAX / 2
IF (N .LT. 1) N = 1
A, B, C, D, R01, R02, R03, R04 = 0, 0
5 IF (N .EQ. 2) GOTO 15
CALL FINO (N-1, R01, M1, M0, NBO, A21)
A = (N-1) * (1+R0/R01)
10 CALL FINO (N, R02, M1, M0, NBO, A21)
B = N * (1+R0/R02)
IF (N .EQ. NMAX) GOTO 30
CALL FINO (N+1, R03, M1, M0, NBO, A21)
C = (N+1) * (1+R0/R03)
30 IF (NMAX-1, D, EQ, N) GOTO 40
CALL FINO (N+2, R04, M1, M0, NBO, A21)
D = (N+2) * (1+R0/R04)
40 FN1 = C - B
FN2 = B - A
FN3 = D - C
Q = N
10 Q1 = B
Q2 = R02
Q3 = N - 1
Q4 = A
Q5 = R01
Q6 = N + 1
Q7 = C
Q8 = R03
IF (FN1 .GE. 0.0, AND, FN2 .LE. 0.0) GOTO 100
100 IF (N .EQ. NMAX, AND, FN1 .LE. 0.0) GOTO 100
100 IF (N .EQ. 1.0, AND, FN1 .GE. 0.0) GOTO 100
Q = N + 1
Q1 = C
Q2 = R03
Q3 = N
Q4 = B
Q5 = R02
Q6 = N + 2
Q7 = D
Q8 = R04
100 IF (FN3 .GE. 0.0, AND, FN1 .LE. 0.0) GOTO 100
100 IF (FN1 .LE. 0.0, AND, NMAX-1.0, EQ, N) GOTO 100
Q = N + 2
Q1 = D
Q2 = R04
Q3 = N + 1
Q4 = C
Q5 = R03
Q6 = Q7, Q8 = 0, 0
IF (NMAX < 2.0, EQ. N, AND, FN3, LE, 0.0) GOTO 100
Q = 1
Q1 = A
Q2 = R01
Q3 = Q4 = Q5 = 0.0
Q6 = 2
Q7 = B
Q8 = R02
IF (N, EQ. 2.0, AND, FN2, GE, 0.0) GOTO 100
IF (FN1, EQ. 0.0) PRINT, 'ERROR #1'
IF (FN2, EQ. 0.0) PRINT, 'ERROR #2'
DN1 = 1 / (1 - (FN3 / FN1))
DN2 = 1 / (1 - (FN1 / FN2))
IF (DN1, LT, DN2, AND, -DN1, GT, DN2) THEN N = N + DN1
IF (DN1, LT, DN2, AND, -DN1, LE, DN2) THEN N = N + DN2
IF (DN2, LE, DN1, AND, -DN2, LE, DN1) THEN N = N + DN2
IF (N, LE, 0) THEN
IF (N, GT, NMAX) THEN N = NMAX
PRINT, NP, R02
GOTO 5
100 PRINT 200, Q3, Q4, Q5
PRINT 200, Q6, Q1, Q2
PRINT 200, Q6, Q7, Q8
U = (1 + (Q3 - Q2) / Q2) / ((Q4 - Q3) / Q2 - 1)
L = (1 - (Q2 - Q5) / Q2) / ((Q7 - Q2) / Q2 - 1)
PRINT 210, U, L
PRINT 220, U * Q2, U * Q2
210 FORMAT ('X', 20.14, '< (RO/R01) <', 'R20.14')
220 FORMAT ('X', 20.14, '< (RO <', 'R20.14')
200 FORMAT ('X', 12 = 'R20.14'/', 'R1 = ', 'R20.14')
PRINT 9900
900 FORMAT ('1')
PAUSE
STOP
END
SUBROUTINE SUB21(NTOT, P, DER1)
C FINDS THE EXPECTED NUMBER OF UNITS IN RESUPPLY.
D.P. NTOT, P(101), DER1
DER1 = 0.0
NTOT = NTOT
DO 10 NRESUP = 1, NTOT
10 DER1 = NRESUP * P(NRESUP) + DER1
RETURN
END
SUBROUTINE NINE(NTOT, M0, P, PD, RU, NB, M1)
C FINDS THE PROBABILITIES FOR J = 0 TO J = NTOT - M0 (USES NATURAL LOGSTO ALLOW FOR LARGER NTOT).
DOUBLE PRECISION P(101), NTOT, P, PD, RU, A, B, D, E, F, G, H, Z, Y, NB, M1
ITEST = NTOT - M0 + 1
111 = ITEST

A-47
IF(I11.LT.1)I11=1
DO 10 J1=1,1TEST
J=J1-1
Y,Z=0,0
IF(I1TEST.LT.1)GOTO 21
DO 20 N1=1,1TEST
NRESUP=N1-1
A=(NRESUP-J)*LOG(10*R0)
C,D=0,0
IF(J.LE.1)GOTO 35
DO 30 K=2,J
B=K
30 C=C+LOG(B)
35 IF(NRESUP.LE.1)GOTO 45
DO 40 K=2,NRESUP
B=K
40 D=D+LOG(B)
45 E=A+C-D
CALL FLOW(E,J,Ix,IY)
IF (E.LT.-174.996)GOTO 20
Z=Z+EXP(E)
20 CONTINUE
21 IF(I1TEST.LE.1)I1TEST=1
I1OT=I1OT+1
I1TEST=I1TEST+1
DO 50 N1=1,1TEST,I1OT
NRESUP=N1-1
C,D=0,0
M=MO
IF(M.LE.1)GOTO 65
DO 60 K=2,M
B=K
60 C=C+LOG(B)
65 IF(J.LE.1)GOTO 75
DO 70 K=2,J
B=K
70 D=D+LOG(B)
75 A=F=0.0
IF(NRESUP.LE.1)GOTO 85
DO 80 K=2,NRESUP
B=K
80 A=A+LOG(B)
85 N=NTOT-NRESUP
IF(N.LE.1)GOTO 95
DO 90 K=2,N
B=K
90 F=F+LOG(B)
95 G=(NTOT-MO-J)*LOG(10)
H=(NRESUP-J)*LOG(R0)
E=C+D+C+H-A-F
CALL FLOW(E,J,Ix,IY)
IF (E.LT.-174.996)GOTO 50
Y=Y+EXP(E)
90 CONTINUE
95 CONTINUE
50 CONTINUE
IF(J.EQ.0) P0=1.0/(Y+Z)
IF(J.EQ.0) GOTO 10
P(J)=1.0/(Y+Z)
10 CONTINUE
CALL TEN(NTOT,M0,P,RO)
CALL SIXTEEN(NTOT,M1,P,RO,PC)
RETURN
END

SUBROUTINE TEN(NTOT,M0,P,RO)
C FINDS THE PROBABILITIES FOR J=NTOT-MO+1 TO J=NTOT
USES NATURALLOGS TO ALLOW FOR LARGER NTOT.
D.P.P(101),NTOT,M0,RO,A,B,C,D,E,F,G,H,Z,Y
INITIAL=NTOT-MO+1
IF(INITIAL.LT.1) INITIAL=1
ITOT=NTOT
DU 10 J=INITAL,ITOT
Y+Z=0.0
DU 20 N1=1, INITIAL
NRESUP=N1-1
A,C,H,G=0.0
IF(J.LE.1) GOTO 105
DU 100 K=2,N
B=K
100 G=G+LOG(B)
105 N=NTOT-J
) IF(N.LE.1) GOTO 115
DC 110 K=2,N
B=K
110 H=H+LOG(B)
115 IF(NRESUP.LE.1) GOTO 35
DC 30 K=2,NRESUP
B=K
30 A=A+LOG(B)
35 N=M0
IF(N.LE.1) GOTO 45
DU 40 K=2,N
B=K
40 C=C+LOG(B)
45 D=(NRESUP-J)*LOG(R0)
E=(NRESUP-(NTOT-M0))*LOG(M0)
F=D+E+G+H-A-C
CALL FLOW(F,J,IX,ITY)
IF (F.LT.-174.995) GOTO 20
Y=Y+EXP(F)
20 CONTINUE
DC 50 NRESUP=INITAL, ITOT
A,C,D,E,F=0.0
IF(J.LE.1) GOTO 65
DU 60 K=2,N
B=K
60 A=A+LOG(B)
I=NTOT-J
IF(I.LE.1)GOT0 75
DO 70 K=2,I-1
B=K
C=C+LOG(A)
75 I=NTOT-NRESUP
IF(I.LE.1)GOT0 85
DO 80 K=2,NRESUP
B=K
80 Z=E+LOG(B)
85 IF(NRESUP.LE.1)GOT0 95
DO 90 K=2,NRESUP
B=K
90 F=F+LOG(B)
95 D=(NRESUP-J)*LOG(DP)
B=D+A*C-E-F
CALL FLOWS(J,IX,IY)
IF(B.LT.-174.996)GOT0 50
Z=Z+EXP(B)
50 CONTINUE
P(J)=1.0/(Y+Z)
10 CONTINUE
RETURN
END

SUBROUTINE SIXTEEN (NTOT,M1,P,NB,PD)
C FINDS THE EXPECTED NUMBER OF BACKGROUNDS.
DOUBLE PRECISION NTOT,M1,P(101),NB,PD,3
ITOT=NTOT
NB=0.
INITIAL=NTOT-M1+1
IF(INITIAL.LE.0)NB=PD
IF(INITIAL.LE.0)INITIAL=1
DO 10 NRESUP=INITIAL,ITOT
B=(M1-(NTOT-NRESUP))*P(NRESUP)
10 NB=NB+6
RETURN
END

SUBROUTINE FLOWS(X,J,IX,IY)
C CHECKS FOR OVERFLOW AND UNDERFLOW.-PRINTS WARNING MESSAGE IF
ONE IS FOUND.
D,P,X
IF(X.LT.-174.996)IX=IX+1
IF(X.GT.174.996)IY=IY+1
IF(X.LT.-174.996)PRINT5,J
5 FORMAT('****UNDERFLOW****** J=',13)
IF(X.GT.174.996)PRINT10,J
10 FORMAT('****OVERFLOW****** J=',13)
C SETS X EQUAL TO MAXIMUM VALUE WITH NO OVERFLOW IF AN OVERFLOW
IS FOUND (LN(X) > 10**76).
RETURN
END
SUBROUTINE NBSTK(NB,NTOT,MC,M1,RO)
D.O. NTOT, M1, NB, STOCK, D, C, MC, RO, PG
M=NTOT-M1+1

ITEST=100

NB=0.0
IF(M.LE.0) NB=EXP(-MD*RO)

IF(M.LE.0) M=1

DG 50 NRESUP=M, ITEST
C=0.0
IF(NRESUP.LE.1) GOTO 65

DG 60 J=2, NRESUP
D=J

20 C=C+LOG(D)

50 NB=NB+(M1-NTOT+NRESUP)*EXP(NRESUP*LOG(MD*RO)-C)*EXP(-MD*RO)

CONTINUE

RETURN

END

SUBROUTINE FIND(NTOT, RO1, M1, MD, NB2, A)

DOUBLE PRECISION RO, RO1, NTCT, COUNT, MU, PB(I91), PDB, NB2, PC(I91), POC, NB3, M1, DER5, DER1C, DR, EPSILN, A, NB2A

RO, RO1 = 1.0

1032 IF(A.NF.0.0) CALL NINE(NTOT-1, MU, PB, PDB, RO1, NB2, A)
    IF(A.NE.0.0) CALL NINE(NTOT, MU, PC, POC, RO1, NB3, M1)
    IF(A.NF.0.0) CALL SUB21(NTOT, PC, DER1C)
    IF(A.EQ.0.0) CALL NBSTK(NB3, NTOT, MO, M1, RO1)
    IF(A.EQ.0.0) CALL NBSTK(NB2, NTOT-1, MO, M1, RO1)
    IF(A.EQ.0.0) DER1C=M0*RO1
    COUNT=COUNT+1

DER5=(-DER1C/RO1)*(NB3-NB2)

DRO=(NB2A-NB3)/DER5

RO1=RO1+DRO

EPSILN=DRO/RO1

IF(EPSILN.LE.2E-14 .AND. EPSILN.GE.-2E-14) GOTO 1031

IF(COUNT.GE.100) GOTO 1031

GOTO 1031

1031 RETURN

END
PROBLEM 2
PROGRAM LISTING
/LOGON "NC"
/DO RUN=WATFIV
$JOB (LNNTJQQ)

DOUBLE PRECISION N, ROPI, ROPI, ROQ, ROA, ROB, ROC, DRO1P, DRO2P, A1, A3
DOUBLE PRECISION A, B, C, DR01, RO2, RO3, RO4, ROZ, ROA, A21, M1, MO, FN1, FN2
IFN3, P(101), PO, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8

INTEGER DN1, DN2
READ 5, 5) RO, MO, ZO
DO 1000 M3=1,2
M1=MO
A21=M3=1.0
PRINT RO, MO, ZO, A21
NMIN=1
NMAX=20
IF(NMAX.EQ.ZO) NMAX=ZO-1.0
I=ZO/2
IF(A21.EQ.0.0) N=1
IF(N.LT.1) N=1
A=B=C=D=RO1=RO2=RO3=RO4=0.0
5 IF(N.EQ.1) GOTO 10
RO1=RO*(N-1)/(ZO-(N-1))
IF(A21.EQ.0.0) CALL NBSTK(A*N-1, MO, M1, RO1)
IF(A21.NE.0.0) CALL NINEK(N-1, MO, PO, RO1, A, M1)
10 RO2=RO*N/(ZO-N)
IF(A21.EQ.0.0) CALL NBSTK(B*N, MO, M1, RO2)
IF(A21.NE.0.0) CALL NINEK(N, MO, PO, RO2, B, M1)
IF(N.EQ.NMAX) GOTO 30
R03=R0*(N+1)/(Z0-(N+1))

IF(A21.EQ.0.0) CALL NBSTK(Cn+1,MO,M1,R03)
IF(A21.NE.0.0) CALL NINE(N+1,MO,PO,R03,C,M1)

30 IF(NMAX=1.0.EQ.N) GOTO 40
R04=R0*(N+2)/(Z0-(N+2))

IF(A21.EQ.0.0) CALL NBSTK(Dn+2,MO,M1,R04)
IF(A21.NE.0.0) CALL NINE(N+2,MO,PO,R04,D,M1)

40 FN1=C-B
FN2=B-A
FN3=D-C
Q=N
Q1=B
Q2=R02
Q3=N-1
Q4=A
Q5=R01
Q6=N+1
Q7=C
Q8=R03
IF(FN1.GE.0.0.AND.FN2.LE.0.0) GOTO 100
IF(N.EQ.NMAX.AND.FN2.LE.0.0) GOTO 100
IF(N.EQ.1.0.AND.FN1.GE.0.0) GOTO 100
Q=N+1
Q1=C
Q2=R03
Q3=N
Q4=B
Q5=R02
Q6=N+2
Q7=D
Q8=R04

IF(FN3·GE·0·0·AND·FN1·LE·0·0)GOTO 100

IF(FN1·LE·0·0·AND·NMAX·LE·0·0·EQ·N)GOTO 100

Q=N+2
Q1=D
Q2=R04
Q3=N+1
Q4=C
Q5=R03

Q6=Q7=Q8=0.0

IF(NMAX·LE·2·0·EQ·N·AND·FN3·LE·0·0)GOTO 100

Q=1
Q1=A
Q2=R01
Q3=Q4=Q5=0.0
Q6=2
Q7=B
Q8=R02

IF(N·EQ·2·0·AND·FN2·GE·0·0)GOTO 100

IF(FN1·EQ·0)PRINT:"ERROR #1"

IF(FN2·EQ·0)PRINT:"ERROR #2"

DN1=1/(1-(FN3/FN1))
DN2=1/(1-(FN1/FN2))
IF(IABS(DN1)*GE*IABS(DN2))ND=DN2
IF(IABS(DN2)*GT*IABS(DN1))ND=DN1
IF(ND*NE*0)GOTO 98
IF(FN1*GT*0)ND=-1
IF(FN1*LT*0)ND=2
98 IF(ND*EQ*1)ND=2
IF(FN1*LT*0*AND*ND*LT*0)ND=2
N=ND
IF(N*LE*0)N=1
IF(N*GT*NMAX)N=NMAX
PRINT,ND,RO2
GOTO 5
100 PRINT200,Q3,Q4,Q5
PRINT200,Q1,Q2
PRINT200,Q6,Q7,Q8
200 FORMAT(*CN=*D20,14*5X,*NB=*D20,14*/*,RO1=*D20,14*)
N=Q
ROP1=RO
COUNT=0*0
300 ROA=(N+1)*ROP1/(ZC-N-1)
ROB=N*ROP1/(ZC-N)
COUNT=COUNT+1
CALL HELP(N+1,M0,ROA,A1,M1,A21)
CALL HELP(N,M0,ROB,A3,M1,E21)
IF(A*ROA-B*ROB)*EQ*0*0)PRINT3C1
IF((A*ROA-B*ROB)*EQ*0*0*AND*(A1-A3)*LE*2E-14)Q=0*0
IF((A1-A3)*GE*-2E-14*AND*Q*EQ*0*0)ROP1=0*0
IF(ROP1*EQ*0*0)GOTO 310
301 FORMAT(*ERROR #1*)

IF(COUNT.EQ.1.0)GOTO 309
Q2=-(A1-A3)/(RA+ROB)/ROP1-DR1P
IF(Q2.LE.1E-14.AND.COUNT.GE.5.0)ROP1=0.0
IF(ROP1.EQ.0.0)GOTO310

309 DR1P=-(A1-A3)/(RA+ROB)/ROP1
IF(COUNT.GT.2)GOTO 307
IF(DR1P.LT.0.0)COUNT=3.0
IF(COUNT.EQ.2.0)GOTO 307
IF(COUNT.EQ.1.0)DRO1P=DR1P
IF(COUNT.EQ.1.0)Q7=A1-A3
IF(COUNT.EQ.1.0)GOTO 307
IF(A1-A3).LT.0.0.AND.Q7.LT.0.0)GOTO 307
IF(A1-A3).LT.0.0.AND.Q7.GT.0.0)GOTO 307

DRO1P=DR1P
COUNT=1.0
IF(ROP1+DRO1P).LE.0.0.AND.ROP1.EQ.1E-5)GOTO 310

307 ROP1=ROP1+DRO1P
IF(ROP1.LE.0.0)ROP1=1E-5
IF(ROP1.EQ.1E-5.AND.COUNT.GE.6.0)ROP1=0.0
IF(ROP1.EQ.0.0)GOTO 310
IF(DRO1P.LE.2E-14.AND.DRO1P.GE.-2E-14)GOTO 310
PRINT,ROP1,DRO1P,N,Z0
GOTO 300

310 ROP2=RO
COUNT=1.0
320  \( \text{ROC} = (N-1) \frac{\text{ROP2} \times (Z0-N+1)}{\text{Z0-N+1}} \)

\( \text{ROB} = N \frac{\text{ROP2}}{\text{Z0-N}} \)

\( Q = 0.0 \)

CALL HELP(N=1,MO,ROA,A1,M1,C,A21)

CALL HELP(N,MO,ROB,A3,M1,B,A21)

IF((C=ROC-B=ROB).EQ.0.0)PRINT302

IF((C=ROC-B=ROB).EQ.0.0)ROP2=1

IF(ROP2.EQ.1.0)GOTO 330

302  FORMAT(*ERROR #2*)

ROP2=-(A1-A3)/((C=ROC-B=ROB)/ROP2)

IF(COUNT.GT.2.0)GOTO 306

IF(DRO2P.GT.0.0)COUNT=3.0

IF(COUNT.EQ.3)GOTO 306

IF(COUNT.EQ.1.0)DRO2P=-DRO2P

IF(COUNT.EQ.1.0)Q=A1-A3

IF(COUNT.EQ.1.0)GOTO 306

IF((A1-A3).LT.0.0.AND.Q.GT.0.0)GOTO 306

IF((A1-A3).GT.0.0.AND.Q.LT.0.0)GOTO 306

DRO2P=-DRO2P

COUNT=1.0

306  ROP2=ROP2+DRO2P

IF(ROP2.GE.1E+04)ROP2=100

IF(ROP2.EQ.100)GOTO 330

IF(ROP2.LT.R0)ROP2=COUNT*RO

IF(COUNT*RO.EQ.ROP2)COUNT=0.0

IF(DRO2P.LE.2E-14.AND.DRO2P.GE.-2E-14)GOTO 330

A-58
COUNT=COUNT+1
PRINT, ROP2, DRO2P, N
GOTO 320
330 PRINT340, ROP1, ROP2
PRINT900
900 FORMAT('')
1000 CONTINUE
STOP
END

SUBROUTINE NINE(NTOT, MO, PO, RO, NE, M1)
DOUBLE PRECISION LOG, DEXP
ITEST=NTOT-MO+1
111=ITEST
IF(111.LT.1)111=1
DO 10 J1=1, ITEST
J=J1-1
Y=Z=0.0
IF(ITEST.LT.1)GOTO 21
DO 20 NI=1, ITEST
NRESUP=NI-1
A=(NRESUP-J)*DLOG(MO/RO)
C=D=0.0
IF(J.LE.1)GOTO 35
DO 30 K=2, J
B=K
30 C=C+LOG(B)
35 IF(NRESUP.LE.1)GOTO 45

DO 40 K=2,NRESUP

B=K
40 D=D+LOG(B)
45 E=A+C=D

CALL FLOW(E,J,JX,IY)

IF (E.LT.174.996)GOTO 20

Z=Z+EXP(E)
20 CONTINUE

21 IF(ITEST.LE.1)ITEST=1

ITOT=ITOT+1

ITEST=ITEST+1

DO 50 M1=ITEST,ITOT

NRESUP=M1-1

C=D=0.0

M=0.0

IF(M.LE.1)GOTO 65

DO 60 K=2,M

B=K
60 C=C+LOG(B)
65 IF(J.LE.1)GOTO 75

DO 70 K=2,J

B=K
70 D=D+LOG(B)
75 A=F=0.0

IF(NRESUP.LE.1)GOTO 85
DO 80 K=NRESUP
   B=K
80  A=A+DLOG(B)
85  N=NTOT-NRESUP
    IF(N.LE.1)GOTO 95
    DO 90 K=N
       B=K
90   F=F+DLOG(B)
95   G=(NTOT-MO-J)*DLOG(MO)
    H=(NRESUP-J)*DLOG(RO)
    E=C+D+G+H-A=F
    CALL FLOW(E,J,IX,IY)
    IF(E.LT.174.996)GOTO 50
    Y=Y+DEXP(E)
50   CONTINUE
    IF(J.EQ.0)PO=1.0/(Y+Z)
    IF(J.EQ.0)GOTO 10
    P(J)=1.0/(Y+Z)
10   CONTINUE
    CALL TEN(NTOT,MO,PO,RO)
    CALL SIXTEN(NTOT,M1,P,NB,PO)
RETURN
END
SUBROUTINE TEN(NTOT,MO,PO,RO)
DOUBLE PRECISION P(I),NTOT,PO,MO,RO,A,B,C,D,E,F,G,H,Z,Y
DOUBLE PRECISION DLOG,DEXP
INITIAL=NTOT+MO+1
IF(INITIAL.LT.1)INITIAL=1
ITOT=NTOT
DO 10 J=INITAL,ITOT
Y=Z=0.0
DO 20 N1=1,INITAL
NRESUP=N1-1
A=G=H=G=0.0
IF(J.LE.1)GOTO 105
DO 100 K=2,J
B=K
100 G=G+DLOG(B)
105 N=NTOT-J
IF(N.LE.1)GOTO 115
DO 110 K=2,N
B=K
110 H=H+DLOG(B)
115 IF(NRESUP.LE.1)GOTO 35
DO 30 K=2,NRESUP
B=K
30 A=A+DLOG(B)
35 N=MO
IF(N.LE.1)GOTO 45
DO 40 K=2,N
B=K
40 C=C+DLOG(B)
45 D=(NRESUP-J)*DLOG(RO)
E=(NRESUP-(NTOT-MO))*DLOG(MO)
F=D+E+G+H=A-C
CALL FLOW(F, J, IX, IY)

IF (F.LT.-174.996) GOTO 20

Y = Y + DEXP(F)

20 CONTINUE

DO 50 NRESUP = INITIAL, ITOT

A = C = D = E = F = 0.0

IF (J.LE.1) GOTO 065

DO 60 K = 2, J

B = K

60 A = A + DLOG(B)

65 I = NTOT - J

IF (I.LE.1) GOTO 75

DO 70 K = 2, I

B = K

70 C = C + DLOG(B)

75 I = NTOT - NRESUP

IF (I.LE.1) GOTO 85

DO 80 K = 2, I

B = K

80 E = E + DLOG(B)

85 IF (NRESUP.LE.1) GOTO 95

DO 90 K = 2, NRESUP

B = K

90 F = F + DLOG(B)

95 D = (NRESUP = J) * DLOG(RO)

B = D + A + C - E - F

CALL FLOW(B, J, IX, IY)
IF(B.LT.174.996)GOTO 50
Z=Z+DEXP(B)
50 CONTINUE
F(J)=1.0/(Y+Z)
10 CONTINUE
RETURN
END

SUBROUTINE SIXTEEN(NTOT,M1,P,NB,PO)
DOUBLE PRECISION NTOT,M1,P,101,NB,PO,B
ITOT=NTOT
NB=0.
INITIAL=NTOT-M1+1
IF(INITIAL.LE.0)NB=PO
IF(INITIAL.LE.0)INITIAL=1
DO 10 NRESUP=INITIAL,ITOT
B=(M1-(NTOT-NRESUP))*F(NRESUP)
10 NB=NB+B
RETURN
END

SUBROUTINE FLOW(X,J,IX,IY)
DOUBLE PRECISION X
IF(X.LT.-174.996)IX=IX+1
IF(X.GT.174.996)IY=IY+1
IF(X.LT.-174.996)PRINT5,J
5 FORMAT(*,****UNDERFLOW**** J='13)
IF(X.GT.174.996)PRINT10,J
SUBROUTINE NBSTK(NB, NTOT, MO, M1, RO)
DOUBLE PRECISION NTOT, M1, NB, STOCK, D, C, MO, RO, PO
DOUBLE PRECISION DLOG, DEXP
M = NTOT - M1 + 1
I TEST = 100
NB = 0.0
IF(M LE.0)NB = DEXP(-MO*RO)
IF(M LE.0)M = 1
DO 50 NRESUP = M, I TEST
C = 0.0
IF(NRESUP LE.1)GOTO 65
DO 60 J = 2, NRESUP
D = J
60 C = C + DLOG(D)
65 IF(NRESUP * DLOG(MO*RO) - C - MO*RO LT. - 100) GOTO 111
50 NB = NB + (M1 - NTOT + NRESUP) * DEXP(NRESUP * DLOG(MO*RO) - C - DEXP(-MO*RO))
111 CONTINUE
RETURN
END

SUBROUTINE HELP(N, MO, RO, A3, M1, B, A2)
DOUBLE PRECISION N, MO, RO, A3, M1, B, A2, P, C, D, E
IF(A21 EQ.0.0)CALL NBSTK(A2, N - 1, MO, M1, RO)
IF(A21 EQ.0.0)CALL NBSTK(A3, N, MO, M1, RO)
IF(A21.EQ.0.0) DER1 = MO * ROB
IF(A21.NE.0.0) CALL NINE(N-1, MO, P, PO, ROB, A2, M1)
IF(A21.NE.0.0) CALL NINE(N, MO, P, PO, ROB, A3, M1)
IF(A21.NE.0.0) CALL SUB21(N, P, DER1)
B = -(DER1/ROB)*(A3-A2)
RETURN
END

SUBROUTINE SUB21(N, P, DER1)
DOUBLE PRECISION N, P(101), DER1
DOUBLE PRECISION DLOG, DEXP

DER1 = 0.0

ITOT = N

DO 10 NRESUP = 1, ITOT
 10 DER1 = DER1 * NRESUP * P(NRESUP)

RETURN
END

ENTRY
 5  4  6
/LLOGOFF
APPENDIX B
EXAMPLE NUMERICAL RESULTS
## Example Problem 2 Results

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Output Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_0$</td>
<td>$N_{opt}$</td>
</tr>
<tr>
<td>0.01</td>
<td>1</td>
</tr>
<tr>
<td>0.01</td>
<td>1</td>
</tr>
<tr>
<td>1.00</td>
<td>1</td>
</tr>
<tr>
<td>0.01</td>
<td>5</td>
</tr>
<tr>
<td>1.00</td>
<td>5</td>
</tr>
<tr>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
</tr>
<tr>
<td>1.00</td>
<td>25</td>
</tr>
<tr>
<td>0.01</td>
<td>15</td>
</tr>
<tr>
<td>1.00</td>
<td>35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$N_{opt}$</th>
<th>$[N_1]_{opt}$</th>
<th>$(\rho_1)_{opt}$</th>
<th>$(\rho_1)_{min}$</th>
<th>$(\rho_1)_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>1</td>
<td>5.5</td>
<td>0.02667</td>
<td>0.25614X10^-2</td>
</tr>
<tr>
<td>1.00</td>
<td>125</td>
<td>25</td>
<td>0.05250</td>
<td>0.0034220</td>
</tr>
<tr>
<td>0.01</td>
<td>0.5</td>
<td>10.5</td>
<td>0.54176</td>
<td>0.49710</td>
</tr>
<tr>
<td>1.00</td>
<td>6.2500</td>
<td>0.92308</td>
<td>0.92308</td>
<td>0.92308</td>
</tr>
<tr>
<td>0.01</td>
<td>0.3245X10^-14</td>
<td>0.04833</td>
<td>0.04833</td>
<td>0.04833</td>
</tr>
<tr>
<td>1.00</td>
<td>6.25727</td>
<td>0.94444</td>
<td>0.94444</td>
<td>0.94444</td>
</tr>
<tr>
<td>0.01</td>
<td>10.00000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>0.50000</td>
</tr>
<tr>
<td>1.00</td>
<td>6.28137</td>
<td>0.67647</td>
<td>0.67647</td>
<td>0.67647</td>
</tr>
</tbody>
</table>
The purpose of the Air Force freight distribution system is to meet spare parts demand requirements at minimum cost. Budgetary constraints have suggested that total expected backorder level for items at user installations be minimized subject to a given dollar expenditure level for inventory investment. LOGAIR (a dedicated Air Force air transport service) is a major transportation sub-system to support spare parts delivery requirements of users of high priority items. A two-echelon inventory system for spare parts

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
backorder level inventory modeling optimal resupply time level
conservative system investment items resupply system modeling
demand process modeling logistics system queueing models
gradient technique optimal investment level

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of the Air Force freight distribution system is to meet spare parts demand requirements at minimum cost. Budgetary constraints have suggested that total expected backorder level for items at user installations be minimized subject to a given dollar expenditure level for inventory investment. LOGAIR (a dedicated Air Force air transport service) is a major transportation sub-system to support spare parts delivery requirements of users of high priority items. A two-echelon inventory system for spare parts
20. Abstract

delivery exists with centralized, specialized inventories at the Air Logistics Centers (ALCs) and decentralized, broad profile inventories located at user installations. LOGAIR provides transport with low order and ship time to reduce resupply time in the maintenance of inventory safety levels at user bases. A systems approach is used to formulate a cost/benefit model which recognizes the impact of the Air Force resupply system upon the total spare parts distribution system in terms of total inventory investment level, total system cost, and backorder level. Given a total expenditure level available for allocation between inventory investment and transportation, the problem is to determine the optimal fractional allocation to be made to transportation (the remaining fraction to be allocated to inventory investment) such that total expected backorder level is minimized. A conceptual framework for trade-off analysis for minimizing total system cost in terms of inventory investment level and resupply time level for a given backorder level is presented. This conceptual framework also allows for the minimization of backorder level in terms of inventory investment level and resupply time level for a given total dollar expenditure level. Queueing models exhibiting various demand and resupply processes are explored and compared to determine the impact of inventory investment level and resupply time level upon backorder level. Specific solution procedures are developed for and are applied to the trade-off analyses mentioned above.