

AD-A096 338

ARMY INVENTORY RESEARCH OFFICE PHILADELPHIA PA
TWO METHODOLOGIES FOR COMBAT UNIT STOCKAGE, (U)
FEB 81 M COHEN, S FRAZZA, L LEBLANC, A KAPLAN

F/G 15/5

UNCLASSIFIED

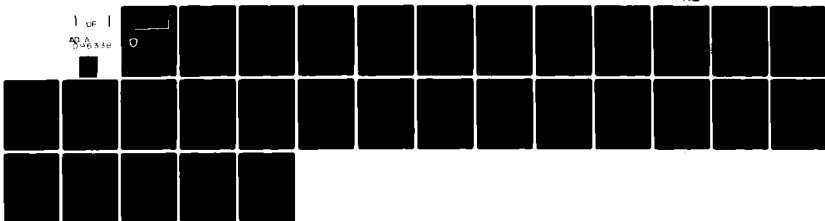
IRO-TR-81-1

NL

1 OF 1

AD-A096 338

0



AD A 096338

AD-
TECHNICAL REPORT
TR 81-1

LEVEL II

10

**TWO METHODOLOGIES FOR
COMBAT UNIT STOCKAGE**

DTIC
MAR 16 1981
E



**U.S. ARMY
INVENTORY
RESEARCH
OFFICE**

FEBRUARY 1981

**ROOM 800
U.S. CUSTOM HOUSE
2nd and Chestnut Streets
Philadelphia Pa. 19106**

Approved for Public Release; Distribution Unlimited

81 3 13 002

DEC FILE COPY

Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the U.S. Army Materiel Command unless so stated.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
	AD-A096	338
4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED	
TWO METHODOLOGIES FOR COMBAT UNIT STOCKAGE	Technical Report	
6. AUTHOR(s)	7. PERFORMING ORG. REPORT NUMBER	
Martin Cohen Sally Frazza Larry LeBlanc, Vanderbilt University	Alan Kaplan	
8. CONTRACT OR GRANT NUMBER(s)		
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
US Army Inventory Research Office, ALMC Room 800, US Custom House, 2nd & Chestnut Sts. Philadelphia, PA 19106		
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE	
US Army Materiel Development & Readiness Command 5001 Eisenhower Avenue Alexandria, VA 22333	February 1981	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES	
(10) 31	27	
	15. SECURITY CLASS. (of this report)	
	UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)		
Approved for Public Release; Distribution Unlimited		
(14) ITC-7K-81-1		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the US Army Materiel Development & Readiness Command unless so stated.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Linear Programming Multi-Echelon Systems LaGrange Model Allowance Lists Operational Readiness		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
The purpose of the report is to present two new algorithms for developing stockage lists of support items for weapon systems at Army Organization level, and to present empirical results of their comparison. Developing economic stockage lists may be viewed as a multi-dimensional "fly-away kit" problem - the multi-dimensionality due to performance requirements for several weapon systems simultaneously. Complicating factors include a constraint on the number of lines stocked, and commonality of support items among the multiple systems.		
(cont)		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403544

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

One algorithm is a Lagrangian approach with a heuristic search for the multiplier. The other is a linear programming formulation. Their comparison made use of three separate data bases: one for a mechanized infantry company, another for an armor company, and the third an artificial data base contrived to reveal algorithm behavior in cases with much commonality.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	Page
TABLE OF CONTENTS.....	1
CHAPTER I INTRODUCTION	
1.1 Description of Problem.....	2
1.2 What Report Covers.....	3
CHAPTER II METHODOLOGIES	
2.1 Computation of System Availability.....	4
2.2 LP Problem Formulation.....	5
2.3 LP Implementation Considerations.....	6
2.4 Lagrangian Approach.....	8
2.5 Lambda Search Routine.....	9
2.6 Alternative Search Routines.....	10
CHAPTER III EMPIRICAL COMPARISONS	
3.1 Data.....	12
3.2 Comparison of Algorithms.....	12
3.3 Possible Additional Research.....	14
3.4 Other Considerations.....	15
APPENDIX A BACKORDER COMPUTATION.....	16
APPENDIX B AVAILABILITY COMPARISON.....	18
APPENDIX C UPPER BOUND ON NON-INTEGERSOLUTIONS IN LP.....	20
APPENDIX D SEARCH ROUTINE SPECS.....	22
REFERENCES.....	24
DISTRIBUTION.....	25

A

CHAPTER I

INTRODUCTION

1.1 Description of Problem

IRO was asked to develop a method of preparing stockage lists for combat, i.e., lists of repair parts to be kept on hand in case of war. The stockage levels were to be based on assumed war time usage of the end items to be supported. It was required that these end items should be 90% available (not unoperational because of shortage of repair parts) during the 15-day time horizon assumed for combat.

A standard cost-minimizing model called SESAME is available for this purpose, but it has been programmed for use with only one end item type at a time. Since it was expected that many of the parts would have application to more than one of the end items, SESAME would be inappropriate. It was decided to use the basic philosophy of SESAME, but not the multi-echelon feature, so as to simplify the model to permit the multiple-application or part-commonality problem to be managed. The loss of the ability to automatically optimize the two echelons would be made up by assuming a constant fill rate for upper echelon resupply in computing lower-echelon stock. While not optimal this seemed to promise a smaller error than ignoring part commonality.

The supply system to be modelled is organized as follows. There is one DSU, a second-echelon organization with both supply and maintenance missions for the parts under consideration. A possibly large number, perhaps 30, ORGs are supported by the DSU. The ORGs can merely replace parts on the end items and cannot repair these parts under the war time conditions we consider. The ORGs are isolated from each other, but can acquire replacement parts in two days from the DSU if in stock. If not, then the end item needing the part is assumed down for the entire 15-day horizon. All parts are assumed to be in stock at the DSU for a fraction of time equal to the fill rate. All parts are assumed to be essential to the operation of the end items to which they apply. A part is ordered from the DSU as soon as it fails, and failure inter-arrival times are distributed exponentially. Thus the end item availability can be computed given the stockage level at ORG and the failure rates and application information for the parts.

An additional complication was introduced to insure mobility and ease supply management of the ORGs. In some cases a constraint was imposed on the

number of different parts (i.e. lines) permitted on the ORG stockage list. If the availability constraints could be met simultaneously with the lines constraint, i.e. if the problem were feasible, this resulted in higher stockage costs than when lines were not constrained. The requirement that the implemented model must handle line constraints was another of the reasons that SESAME was not used.

Technically, the problem to be solved might be described to be of the multi-dimensional "fly-away kit" type, with a constraint on lines stocked, a range of stock levels possible for each part, and multiple systems supported with part commonality.

1.2 What Report Covers

Two alternate algorithms were developed to compute the stockage lists for the ORGs. For the DSU stockage levels a standard algorithm is used, which will not be discussed in this report. The purpose of this report is to present and compare the performance of the two algorithms used to generate the lists for the ORGs.

Briefly, the two algorithms are a Lagrangian heuristic approach and a linear programming (LP) approach. The heuristic employs a search algorithm to adjust the values of a set of Lagrange parameters, one for each end item, until the availabilities are close to the target value. The LP approach is actually an approximate solution to an integer program model, which was too large to solve exactly. This procedure tries many combinations of stockage levels of the different components until end item availabilities are all at least equal to the target; if more than one solution satisfies the availability (and other) constraints, the one which produces the lowest dollar value of stock is chosen.

In this report we first present some definitions and background on availability needed in describing the algorithms. Next we present the LP approach and discuss its implementation, followed by an explanation of the heuristic and its implementation. The data base is then described, and finally the results of the comparison.

CHAPTER II

METHODOLOGIES

2.1 Computation of System Availability

Let B_i be the expected number of backorders for part i at the ORG at a random point in time; i.e. if $B_i(t)$ is expected number at time t , then for a 15-day horizon,

$$B_i = \int_0^{15} B_i(t) dt / 15 \quad (1.1)$$

Computation of B_i is described in Appendix A; clearly it is a function of the stockage level of part i as well as demand for part i and resupply times from DSU to ORG.

Let F_{ij} be that fraction of demand for part i caused by a randomly chosen system of type j . For example, let system type 3 refer to the M60A1 tank, and suppose part i is used only on the M60A1. If an organization supports 50 M60A1's, then $F_{i3} = 1/50 = .02$. Now suppose that part i is also used on a weapon system of type 4 and that the type 4 system accounts for 1/4 of the part's expected demand. Then $F_{i3} = (3/4)(1/50) = .015$.

Defining B_{ij} as the expected number of backorders of part i for a randomly chosen system of type j at a random point in time,

$$B_{ij} = (F_{ij})(B_i) \quad (1.2)$$

Further, if only one part of type i can ever fail on a system of type j , B_{ij} must be less than one, and in fact equals the probability of a backorder, e.g. if $B_{ij} = .02$, this means 98% of the time there are no backorders and 2% of the time there is 1 backorder ($.02 \times 1 = .02$ backordered). It is assumed that B_{ij} can be used to represent the probability part i is available for system j , recognizing that if the number of applications of part on system exceeds one, this is only an approximation. LMI [5] shows how to make a more precise calculation which adjusts for number of part i applications per system, but requires some additional assumptions.

Defining A_j as the percent of time a randomly chosen system of type j is available - or at least is not unavailable for lack of a part - we have

$$A_j = \sum_i (1 - B_{ij}) \quad (1.3)$$

This formula assumes there is no redundancy, and each part is critical, so that the system is operable only if no parts are missing.

A more precise expression for A_j would be:

$$A_j = \int_0^{15} \prod_i (1 - F_{ij} B_{it}) dt / 15 \quad (1.4)$$

2.2 LP Problem Formulation

A key to the linear programming approach is that only a limited number of stock levels, besides 0, are considered for each part. The LP chooses among these levels. The subscript, ℓ , is used below as both an index and as the actual quantity to be stocked.

Let $x_{i\ell}$ refer to stockage of part i at the ℓ^{th} level; i.e. $x_{i\ell}$ equals 1 if part is stocked at that level, and 0 if not. Let C_i be the inventory investment (quantity times unit price) and $B_{ij\ell}$ the backorders attributed to system j when part i is stocked at the ℓ^{th} level. Let TAR_j be the availability target for system j .

We then have this integer programming problem for the decision variables

$x_{i\ell}$:

$$\text{Minimize } \sum_i \sum_{\ell} x_{i\ell} C_{i\ell} \quad (2.1a)$$

$$\text{Subject to } \sum_i \sum_{\ell} \log(1 - B_{ij\ell}) x_{i\ell} \geq \log TAR_j \quad \text{for all } j \quad (2.1b)$$

$$\sum_{\ell} x_{i\ell} = 1 \quad \text{for all } i \quad (2.1c)$$

$$\text{where } x_{i\ell} = 0 \text{ or } 1$$

Condition (2.1b) is derived by taking the log of the condition that achieved availability must at least equal the target; see equation 1.3 for achieved availability. Condition 2.1c insures that exactly one level is chosen for each part, possibly x_{i0} denoting 0 stock with backorders B_{ij0} . Note that if part i is not used on system j , $B_{ij\ell}$ is always 0.

The integer programming problem may be given an alternative form which eliminates explicit consideration of the 0 levels, x_{i0} :

$$\text{Minimize } \sum_i \sum_{\ell > 0} x_{i\ell} C_{i\ell} \quad (2.2a)$$

Subject to

$$\sum_i \sum_{l>0} x_{il} [\log (1-B_{ijl}) - \log (1-B_{ij0})] \geq \quad (2.2b)$$

$$\log \text{TAR}_j - \sum_i \log (1-B_{ij0})$$

$$\sum_{l>0} x_{il} \leq 1 \quad (2.2c)$$

To see the equivalence between (2.1b) and (2.2b), note that in (2.2b):

If $x_{il} = 1$ for some $l>0$ then $\log (1-B_{ijl})$ is on both sides of the equation, and can be cancelled out.

If $x_{il} = 0$, for all $l>0$, then $\log (1-B_{ij0})$ is subtracted from the R.H.S., which is equivalent to adding it to L.H.S. giving the same result as (2.1b) with $x_{il} > 0$ for $l=0$.

If there is a constraint on the number of lines which may be stocked, a third condition is added to (2.2):

$$\sum_i \sum_{l>0} x_{il} \leq \text{MAXLINES} \quad (2.2d)$$

As an integer programming problem (2.2) may be time consuming to solve. However, it may be solved as a continuous linear program with quite acceptable results as shown in section 2.3 of this report. There are two reasons for this. One is that the target availability constraints are not rigid - if in rounding non-integer answers down there is a minor loss in availability, e.g. less than 1%, this is acceptable to the model users. The second reason is that the number of non-integer levels will be small relative to the size of the stockage list and in fact cannot exceed the number of weapons systems, or that number plus 1 if MAXLINES is an active constraint. This is shown in Appendix C. Thus, in a typical problem with 30 weapon systems, and about 250 lines stocked, rounding would not affect more than $31 : 250$ or typically about 10% of the parts.

2.3 LP Implementation Considerations

The linear program described in the previous section was solved using a standard simplex LP code, APEX II. The number of columns, i.e. solution variables

x_{1l} in the previous section, depends on the precision needed. Of course, there must be a compromise between accuracy and problem size. If the number of candidate non-zero stockage levels is constant for all parts, then the number of columns equals the product of this number and the number of parts. This is what was done in order to simplify problem setup. In the chapter on results it is shown that eight stockage levels gave acceptable answers at not too great a cost of running the algorithm.

The trial stock levels (designated as l in the formulation above) are selected as follows:

Let m = expected demand

N = number of trial levels desired

Then if $m + 8\sqrt{m} \leq N \{l\} = 1, 2, \dots, N$,

if $m + 8\sqrt{m} > N \{l\} = [(m + 8\sqrt{m})/N](k)$

where $k = 1, 2, \dots, N$

and the N values are rounded to the nearest integer.

Since $0 \leq x_{1l} \leq 1$ may be non-integer, the stockage level is calculated as

$$S_1 = \sum_l l x_{1l}$$

which is then further treated below.

Although the number of non-integer levels found by the LP code will be few, they must still be disposed of in some way to yield a feasible (if slightly non-optimal) solution. This is accomplished by a rounding rule. Each stockage quantity, S_1 , is adjusted to the nearest integer. This rounding could cause the final availability to be lower than the target in some cases. If S_1 is less than one but greater than zero, it must be decided whether or not to stock the part. The rounding rule was modified in this case to eliminate the part only if it does not cause excessive reduction of availability. The rule also considers the cost of the part: If rounding down a fractional stockage value less than .5 would not reduce any end-item availability by more than .005, and rounding up would increase cost by more than \$50., then round to zero; otherwise stock one unit. Thus, to prevent a single part from reducing availability by one-half percent we would stock one when simple rounding would have stocked zero. Cheap parts are also stocked. In addition

a check is made on the overall availability reduction; a part will be stocked rather than allow the cumulative availability to drop by more than one percent.

In order to run the LP code, a data input deck must first be prepared. To facilitate this job a program was written that gets data for each part from the data base, computes the terms of the sums on both sides of expression (2.2b), and writes out the card images in the correct format for APEX II.

2.4 Lagrangian Approach

Let the λ_j be a set of non-negative real numbers; let S_i and $B_{ij}(S_i)$ be the stockage quantity and backorders given that stockage for part i , and let UP_i be the unit price of item i . Let $\gamma(S_i)$ be a function with value 1 if $S_i > 0$, else value 0.

The Lagrangian approach involves solving problems of the form:

$$\text{Maximize: } - \sum_i S_i UP_i - \lambda_0 \gamma(S_i) + \sum_j \sum_i \lambda_j \log(1 - B_{ij}(S_i)) \quad (4.1)$$

or equivalently:

$$\text{Maximize } \sum_i \left\{ - S_i UP_i - \lambda_0 \gamma(S_i) + \sum_j \lambda_j \log(1 - B_{ij}(S_i)) \right\} \quad (4.1')$$

This is a separable sum of concave functions of the S_i and hence easily solved. Note that the objective in maximizing is to reduce costs and lines stocked and increase availabilities. If there is no line constraint, $\lambda_0 = 0$.

By a Theorem of Everett [1], it is known that if $\{S_i^*\}$ maximizes (4.1) for some set of $\{\lambda_j\}$ with resulting system availabilities $\{A_j^*\}$, then there is no cheaper solution possible with availabilities $\{A_j\}$ such that for all j , $A_j' \geq A_j^*$. Hence if by a manipulation of the λ_j we could get a solution such that $A_j = \text{TAR}_j$ for all j this must be the least cost solution. In general, because of the discreteness of the S_i , there is no solution with all $A_j^* = \text{TAR}_j$. In fact, even if such a solution exists there is no guarantee that there are λ_j which will result in that solution. (This is termed a duality gap). This would occur, for example, if there were two items with identical unit price and failure characteristics and in order to just meet the target availability only one should be stocked. A maximization of (4.1) must result in both items with identical characteristics stocked, or both not stocked. A duality gap would also occur if stocking a less cost effective part was preferred because

stocking the more cost effective part raised availability "needlessly" above target.

In the next section a heuristic is described which attempts to find λ_j such that the resulting A_j are as close as possible to the targets. While this heuristic is logically based on the characteristics of the problem, no general theoretical claims can be made for it. The empirical tests of its effectiveness are presented in section (3.1).

2.5 Lambda Search Routine

Motivation. If there were only one system, and no constraint on lines stocked, a standard procedure would be:

- a. Start with some λ , solve (4.1) and compare availability, denoted $A(\lambda)$, to target availability.
- b. If availability is high, lower λ , and visa versa, since availability is a monotonic function of λ .
- c. Once availability is bracketed, i.e. $A(\lambda_1) < \text{TAR}$ and $A(\lambda_2) > \text{TAR}$, perform a binary search until the bracketing λ 's are sufficiently close together.

Generalization. If the procedure just described were applied to a problem with two systems and target availabilities of 90%, we might find:

Iteration	System 1		System 2	
	λ_1	A_1	λ_2	A_2
1	10	88%	30	87%
2	20	96%	60	92%
3	15	94%	45	88%
4	12.5	93%	52.5	91%
5	11.25	92%	48.75	89%

On each iteration we update λ_1 and λ_2 independently and simultaneously and then solve 4.1 again to get the corresponding availabilities. Unfortunately, it may be true that if λ_2 is in the interval $[48.75, 52.5]$, then for any value of $\lambda_1 \geq 10$, $A_1 \geq 91\%$, the results of iteration 1 notwithstanding. This could occur if there were a part used on both systems. Referring to (4.1), stockage for this part will depend on both λ_1 and λ_2 and therefore A_1 will depend in part on λ_2 .

The heuristic attempts to keep the search on track by recalibrating every

so often; e.g. at iteration 6, the bracketing λ values for system 1 are 10 and 11.25 and for system 2 they are 52.5 and 48.75 with 10 and 52.5 the values found first. In recalibration $\lambda_1 = 10$ and $\lambda_2 = 52.5$ are tried.

Thus, the heuristic is a single system search procedure with the recalibration to account for possible interdependence.

If there is a line constraint, λ_0 (cf 4.1) is found implicitly. Suppose the constraint is 240 lines, and that as a result of an iteration 300 lines qualify for stockage. The items are ranked by their contribution to the objective function, i.e. by

$$\text{Value}_i = -S_i UP_i + \sum_j \lambda_j \left\{ \log (1-B_{ij}(S_i)) - \log (1-B_{ij}(0)) \right\}$$

The top 240 are considered stocked, and the corresponding system availabilities found.

Implementation Details. The search process ends when the search process for each system has ended. This occurs if:

- a. Availability is bracketed and bracketing lambdas are within 1% of each other.
- b. Availability is not bracketed but availability is found which exceeds target by less than 1%, or underachieves target by less than 0.3%.

Calibration is done every 5th iteration. When bracketing lambdas have not been found and availabilities are too high, new lambda is set to 50% of old lambda if availability is off by more than 5%; otherwise it is set to 90% of old. Respective figures when availability is too low are 200%, 111%.

Additional specifications for the search routine are in Appendix D.

2.6 Alternative Search Routines

We first briefly summarize an approach developed by Logistics Management Institute [5]. In the single system case they first compute backorder levels for all potentially optimum stockage levels for all parts. All units of stock are then ranked by their marginal contributions to system availability per dollar spent to buy the unit. In determining marginal contribution, stocking 1 unit of part i is compared to not stocking part i, stocking 2 units of part i is compared to stocking only 1 unit, and so on. LMI then adds units in order of their rank until the desired system availability is achieved.

When there are multiple systems, the systems are first treated independently. If part i is common to systems 1 and 2, it appears in the ranked

lists of both systems, with backorders and cost for each stockage level allocated based on percent of demand accounted for by each system. After the solutions for each system are obtained, a heuristic is used to make adjustments if the stockage level chosen for part i on system 1 is not the same as the level chosen for system 2.

The LMI approach has not been applied to problems with constraints on the number of lines stocked.

Given an explicit Lagrangian approach, a common procedure to find appropriate lambda values is to make use of the dual problem (cf Lasdon). This procedure uses the dual gradient (or subgradient) as a search direction. New lambda values are obtained by "stepping off" from the old lambdas in this direction. If the step size parameters are suitably chosen, this procedure has been proven to converge. Under such a procedure, each new set of $\{\lambda\}$ accepted provides a better (lower) upper bound on solutions of (4.1) which will also satisfy the system availability constraints. As in linear programming, we are thereby protected against cycling back to previous solutions. Unlike linear programming it can be difficult to find a better solution at each iteration.

CHAPTER III

EMPIRICAL COMPARISONS

3.1 Data

Three data bases were used for testing the models. Two consisted of data from the Materiel Readiness Commands (MRCs) for two companies, a mechanized infantry company and a tank company. The end items and necessary end item data for each company were taken from mission profiles provided by TRADOC.

The third data base was created to provide a smaller source of data with a higher degree of commonality. A smaller data base was sufficient for model comparisons and made computer runs faster and easier. The commonality apparent on the MRC data consisted mostly of a few pairs of weapon systems with largely identical parts. This did not test the capacity of the models for handling more intricate situations where one item might appear on more than 2 weapon systems, or weapon systems have some but not all items in common.

The manufactured data base consists of 215 part applications for 5 end items. 123 of these applications are common to (at least) 2 of the end items; 18 of them to 3. (The number of parts (regardless of application) is 151.)

The 5 end items were taken from TACOM data. However, part applications were added to and deleted from each weapon system to induce the desired degree of commonality, hence the original end item identification is meaningless.

3.2 Comparison of Algorithms

Three questions were studied:

- a. Are the results of the two algorithms comparable?
- b. How do the computer costs differ?
- c. How many levels are sufficient in the LP algorithm?

For the first question, we used the final interpolated, rounded solution levels of the LP algorithm and computed end item availabilities, and compared these with availabilities computed (in the same way) by the heuristic algorithm for its list. The LP algorithm usually resulted in stockage costs a few percent above those of the heuristic algorithms. The costs were always within three percent. The results are shown in a table below for the three different sets of data. Results are also broken out for the unconstrained and constrained cases, i.e. whether the number of part lines was prevented from exceeding

some value (110 for COMMON data base, 250 for INFANTRY, and 332 for TANKS).

COST/LINES/TIMES*

	HEUR	LP
COMMON-UN	\$1353/135/112	\$1386/136/186
COMMON-CON	\$15257/118/560	\$15513/119/191
INFANT-UN	\$ 7072/289/262	\$ 7052/287/746
INFANT-CON	\$ 7857/256/**	\$ 7914/256/785
TANKS-UN	\$19189/434/792	\$19368/430/1335
TANKS-CON	\$22460/331/**	\$22968/332/1436

* LP Time is for algorithm only. Add about 20% setup.

** Needed manual intervention as hit time limit.

The third number, computer time, in each entry is the number of "system seconds", a weighted measure of computer resource usage. The current day rate is about \$260 per hour; this represents the resource-weighted fraction of allocatable costs the user reimburses the ARRADCOM S+E Computer Facility. It is probably a serious understatement of the commercial value of the data processing usage.

As can be seen the LP algorithm tends to use considerably more computer time than the heuristic for the unconstrained case. Note that the times for the LP algorithm are understated because they do not include the additional data set up and result processing. This is estimated to total about 20% of the times given for the large data bases and 80% for the small. The constraint on part lines hardly increases the time required by the LP algorithm, but causes a marked increase in the heuristic because many iterations are needed. In fact, for the large data bases, manual intervention was required and no time could be estimated.

The LP algorithm is feasible only because it limits attention to a few stockage levels from the infinity of possible values. As described in Chapter II we have implemented the algorithm using a fixed number of levels for each part.

The higher the demand rate, the more these levels spread out. For most of the parts, the demand rates were so low that complete coverage of levels was achieved up to a large maximum. For a few high-demand parts, however, the stockage levels were spread out; and thus the optimal value may not have been tried.

Since a continuous LP model was used, non-integer stockage levels were obtained for a small number of parts. Linear interpolation was used to force integer solutions in these cases; this introduced additional departure from optimality. These two effects are possibly the cause of the slightly better cost performance of the heuristic as compared to the LP algorithm. However, cost results between LP and heuristic are not completely comparable because achieved end item system availabilities differ. In running the heuristic, availabilities of 89.7% were considered acceptable, since this would be true if the heuristic were implemented, and this leeway permits the heuristic to run faster. The LP solution always gave availability of at least 90%, although after rounding availability could drop. A detailed comparison of availabilities for the cases run is in Appendix B.

It was found in earlier work that four stockage levels gave significantly less optimal results than eight levels. The table below shows that eight and sixteen levels achieve about the same in terms of optimality. The Infantry data was used to make two unconstrained LP runs, one with eight levels, the other with sixteen. The availabilities were the same within two or three significant digits:

Number of Levels	8	16
Number of Parts Stocked	288	287
Stockage Cost	\$7879	\$7862

Only very limited experimentation was done on the effects of interpolation. This did occasionally introduce a large cost increase; for example in the case of an expensive part which the LP algorithm tried to stock in a fractional quantity. No conclusions were drawn from this work.

3.3 Possible Additional Research

The following ideas to improve the LP algorithm seem promising:

- a. Variable stockage levels, using more levels for higher demand or higher priced parts.
- b. An integer programming code.

At the very least, use of an integer program code would identify how much cost is increased by using the continuous approximations and rounding.

Just as the LP chooses between a limited number of stockage levels for each part, so could the heuristic. The advantage would be an increase in computational efficiency, although whether this would make treatment of a constraint feasible is not clear. Certainly, it would not be a good approach if there were more than one constraint, for example a total weight as well as number of lines. Other variants for running the heuristic would compute levels as they were needed, and store them; at any given time values stored would include the optimum for Lagrangians less than or equal to the highest tested to date.

The inherent cost effectiveness of both rules might have been better compared by modifying the LP rounding rule and heuristic to always achieve at least 90% availability. The disadvantage is that such a rigid specification, with no tolerance, was not part of our problem.

3.4 Other Considerations

With a constraint on the number of lines the problem may not be feasible. It is possible to run the heuristic in an unconstrained mode but with a high fixed cost charged per line stocked, e.g. \$100,000. The number of lines in the solution constitutes a reasonable minimum practical list size. Similarly, a cost-per-part-stocked could be added to the objective function of the LP formulation for the same purpose. Comparable results can be achieved by LP by using parametric programming.

The LP approach could be extended to two echelons. For each item different budgets would be tried, analogously to the x_{il} of section 2.2, and a standard algorithm would be used to allocate the budget among the two echelons and compute the corresponding minimum ORG level backorders. If there is a line constraint, alternative item budgets with zero ORG level stockage would also have to be tried.

APPENDIX A

BACKORDER COMPUTATION

We are interested in the expected number of backorders over a 15-day period, given a resupply policy with resupply time of two days. The number of backorders depends on the stockage level at the beginning of the 15 days, the expected demands in the period, and the DSU fill rate, the probability of getting resupply in two days.

Let S = stockage level

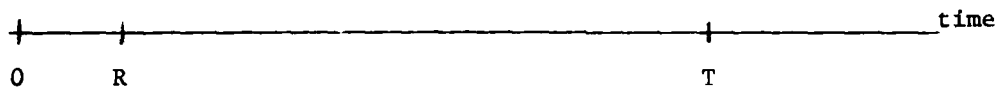
D = demand rate per day

F = fill rate

T = resupply time if DSU not in stock (15 days)
(this will be called "no resupply" below)

R = resupply time if DSU in stock (2 days)

To calculate the average backorders over 15 days, look at the period from 0 to R , during which there is no resupply, and the period from R to T , during which resupply occurs.



At time 0, the stock on hand equals the quantity provided by the PLL. For the first R days this stock is depleted at an average rate D per day. After day R , resupply occurs at the same rate as demand, and thus the expected stock level remains constant over the remaining $T-R$ days.

Assume the following notation:

$BACKO(S,t,D)$ = expected number of backorders at a point in time t

$EVALNR(S,T,D)$ = expected number of total time weighted backorders in T , assuming no resupply

$EVALRE(S,T,D)$ = expected number of time weighted backorders in T , assuming resupply

Then

$EVALRE(S,T,D)$ = (probability of no resupply) $\times$ (backorders with no resupply) + (probability of resupply) $\times$ (backorders with resupply).

For the expected backorder computation with resupply, no resupply is received in the first R days, so the backorders for this period are as for no resupply, but for R rather than T days = $EVALNR(S,R,D)$. For the remaining $T-R$ days, the expected backorders are constant at the day R level = $BACKO(S,R,D)$.

Therefore,

$$\begin{aligned} \text{EVALRE}(S,T,D) &= (1-F)[\text{EVALNR}(S,T,D)] \\ &\quad + F [\text{EVALNR}(S,R,D) + (T-R) \text{BACKO}(S,R,D)] \end{aligned}$$

We assume demands occur according to a Poisson probability distribution.

$p(x;\lambda t)$ = prob of x demands in time t , given a mean rate of λ per unit time t

$$P(r;\lambda t) = \sum_{x=r}^{\infty} p(x;\lambda t) = \text{prob of } r \text{ or more demands}$$

$$\begin{aligned} \text{BACKO}(S,t,\lambda) &= \sum_{j=S}^{\infty} (j-S)p(j;\lambda t) \\ &= \lambda t P(S-1;\lambda t) - SP(S;\lambda t)^* \end{aligned}$$

BACKO is the expression for backorders at a random point in time t for a continuous inventory system with random demands, where it is assumed that an order is placed each time there is a demand. S is the inventory position and λ is the demand rate per day.

To get the expected number of backorders over 15 days (T), we integrate BACKO over t from 0 to T .

$$\begin{aligned} \text{EVALNR}(S,T,\lambda) &= \lambda \int_0^T t P(S-1;\lambda t) dt - S \int_0^T P(S;\lambda t) dt \\ &= \lambda \left[(T^2/2) P(S-1;\lambda T) - (1/2\lambda^2) (S)(S-1)P(S+1;\lambda T) \right]^{**} \\ &\quad - S[TP(S;\lambda T) - (S/\lambda)P(S+1;\lambda T)] \\ &= (\lambda T^2/2)P(S-1;\lambda T) - STP(S;\lambda T) + (S(S+1)/2\lambda)P(S+1;\lambda T) \end{aligned}$$

*Equation (4-112), page 205, Hadley & Whitin

**Equation 19, page 443, Appendix 3, Hadley & Whitin, used twice to evaluate the integrals.

APPENDIX B
AVAILABILITY COMPARISON

INFANTRY

END ITEM	UNCONSTRAINED		CONSTRAINED	
	HEURISTIC	LP ALGORITHM	HEURISTIC	LP ALGORITHM
1	.9014	.9014	.9033	.9021
2	.9061	.9058	.9026	.9024
3	.9090	.9090	.9119	.9090
4	.9002	.9022	.9047	.9006
9	.9056	.9053	.9037	.9039
10	.9011	.9019	.9011	.9019
11	.9306	.9306	.9306	.9306
12	.9601	.9601	.9854	.9601
13	.9029	.9012	.8995	.8992
14	.9046	.9021	.9012	.9028
15	.9007	.8983	.9311	.9319
17	.9565	.9565	.9565	.9565
18	.9038	.9038	.9038	.9038
19	.9255	.9106	.9255	.9106
23	.9092	.9092	.9092	.9092
25	.9092	.8997	.9092	.8997
26	.9123	.9154	.9101	.9131
27	.9368	.9298	.9368	.9298
COST	\$7072	\$7052	\$7857	\$7914

COMMON ITEM DATA BASE

1	.9039	.9039	.8995	.8995
2	.8967	.9024	.8979	.8979
3	.9021	.9028	.8971	.9028
4	.9090	.9053	.9062	.9020
5	.8987	.9048	.8995	.9007
COST	\$13537	\$13866	\$15257	\$15513

ARMOR

END ITEM	UNCONSTRAINED		CONSTRAINED	
	HEURISTIC	LP ALGORITHM	HEURISTIC	LP ALGORITHM
1	.9053	.9053	.9119	.9031
2	.9032	.9032	.9050	.9046
3	.9021	.9021	.8955	.9032
4	.9083	.9028	.9119	.9000
5	.9009	.9009	.9078	.9068
8	.9058	.9058	.8988	.9050
9	.9002	.9002	.9010	.8984
10	.9192	.8835	.9301	.9192
11	.9614	.9614	.9922	.9922
12	.9062	.9058	.9181	.8993
13	.9020	.9007	.8980	.8972
14	.9078	.9021	.8981	.9012
15	.8986	.9013	.8993	.9008
16-19	.9717	.9718	.9719	.9718
20-22	.9341	.9226	.9044	.9031
29	.8994	.8994	.9001	.9004
30	.9910	.9910	.9944	.9947
31	.9269	.9269	.9615	.9269
32	.9506	.9506	.9861	.9506
COST	\$19189	\$19368	\$22460	\$22968

APPENDIX C

UPPER BOUND ON NON-INTEGER SOLUTIONS IN LP

Let W be the number of weapon systems, and P the number of parts. Assume, for the moment, there is a line constraint. There are therefore, (referring to Section 2.2) $W + 1 + P$ constraints.

We use a well known result of linear programming (cf Hadley): If there is an optimum solution, there must be a basic optimum solution. Since the rank, r , of the coefficient matrix cannot exceed the number of constraints, and a basic solution has only r non-zero variables (or even fewer if the basic solution is degenerate), an optimum basic solution to our problem has at most $W + 1 + P$ positive variables. The LP code provides basic solutions.

From the above, we can infer that at most $W + 1$ parts may have non-integer solutions. Consider the P constraints of the form:

$$(C1) \quad \sum_{\ell} x_{i\ell} + k_i = 1$$

where k_i is a slack variable.

Then for each i in a basic, or any other, solution

(1) Some $x_{i\ell} = 1 \rightarrow$ exactly one of variables in (C1) is $\neq 0$

(2) All $x_{i\ell} = 0 \rightarrow$ exactly one of the variables in (C1) is $\neq 0$, namely k_i .

(3) An $x_{i\ell} > 0$, but not equal 1 $\rightarrow$ two or more of the variables in C1 are $\neq 0$, since all variables must sum to 1.

Thus, for every part there is at least one non-zero variable in the solution, and two or more if the part is not stocked at an integer level (case 3).*

Just as there are slack variables in C1, there are slack variables in the W constraint equations relating to weapon system availability and in the line constraint. Let S be the number of these slack variables which are non-zero in the final solution, and let NI be the number of parts for which there are positive non-integer $x_{i\ell}$.

Then

$$(C2) \quad S + 2 NI + (P - NI) \leq W + 1 + P$$

* It can be shown that in fact at most k_i plus two $x_{i\ell}$ will be non-zero.

The left hand side are the total non-zero variables, and the right hand is the number of variables in the (basic) solution.

From (C2), by algebra,

$$(C3) \quad NI \leq W + 1 - S$$

This, therefore, bounds the number of non-integer stock levels in the LP solution.

If there is not a line constraint, we can derive:

$$(C3') \quad NI \leq W - S$$

APPENDIX D

SEARCH ROUTINE SPECS

When SEARCH is called, it has stored in memory for each system.

λ_{OLD}	AV_{OLD}
λ_{OLDEST}	AV_{OLDEST}

which resulted from previous iterations. λ_{OLD} , which resulted in a system availability of AV_{OLD} was found in the last iteration, but λ_{OLDEST} could have been several iterations ago.

Input to SEARCH is a $\lambda_{CURRENT}$ and $AV_{CURRENT}$ for each system, from current iteration.

SEARCH has two functions: update the stored λ 's and availabilities with the current values input. Determine a (different) λ_{NEW} to be evaluated for each system.

See diagram for flow.

Update Logic

If $\lambda_{CURRENT}$ resulted from calibration run (see below) λ_{OLDEST} drops out.

If AV_{OLD} and AV_{OLDEST} are on same side of target availability, e.g. both above it, λ_{OLDEST} drops out.

If AV_{OLD} and AV_{OLDEST} bracket target, replace whichever one is on same side as $AV_{CURRENT}$ i.e. make sure the two availabilities stored after update also bracket target.

Relabel, as necessary, after update so λ_{OLD} always has value which was input to search routine, while λ_{OLDEST} has what used to be either λ_{OLD} or λ_{OLDEST} .

Choice of λ_{NEW}

Calibration Run: every 5th run set $\lambda_{NEW} = \lambda_{OLDEST}$. This is done because AV_{OLDEST} may no longer be obtained if λ_{OLDEST} is rerun (see Section 2.5 of main report).

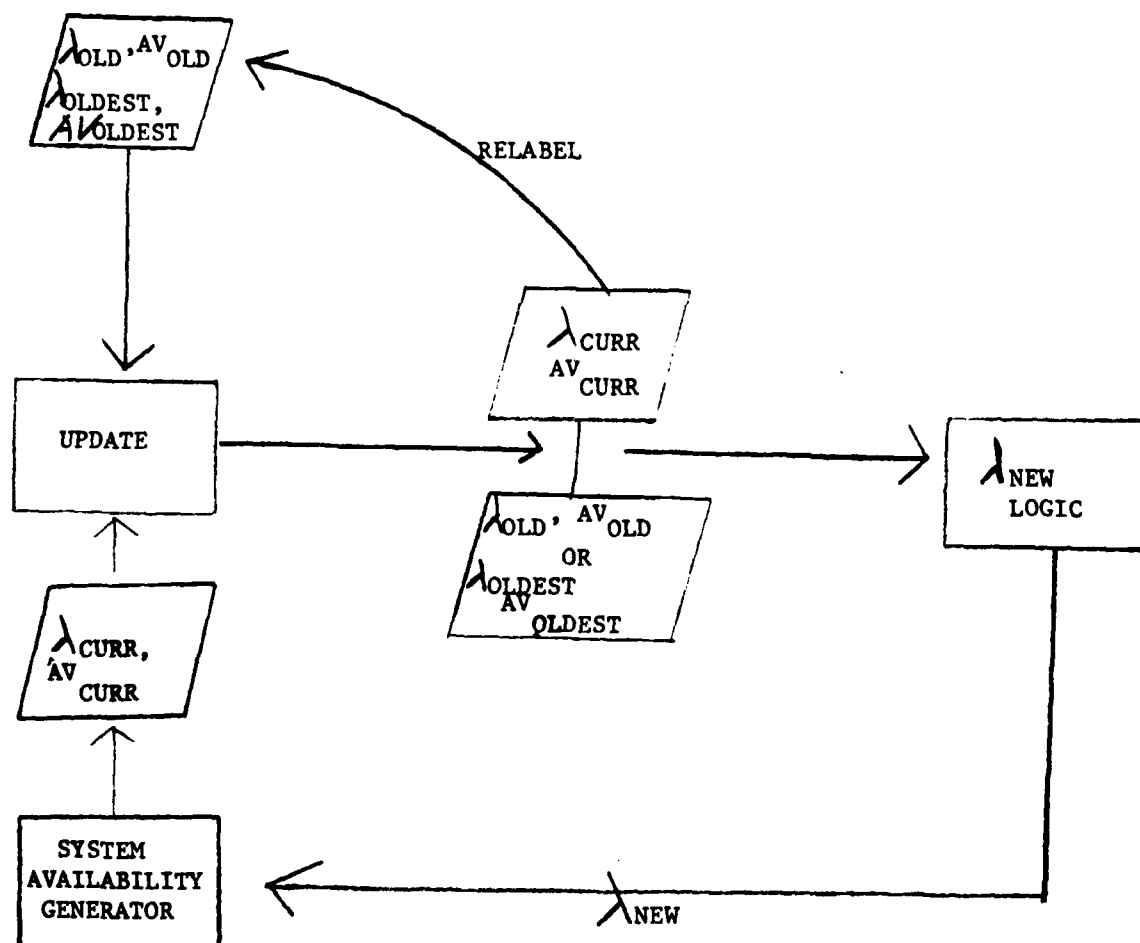
Target Is Not Bracketed: Multiply λ_{OLD} by $FAC(>1)$ if AV_{OLD} is below target or $1/FAC$ if AV_{OLD} is above target. To get FAC

If less than 6 iterations have been done set $FAC = 10$, or if more, set $FAC = 2$.

But if $|AV_{OLD} - TAR| < 5\%$ set $FAC = 1.1$ and if $|AV_{OLD} - TAR| < 1\%$,
 set $FAC = 1$, i.e. $\lambda_{NEW} = \lambda_{OLD}$.*

Target Is Bracketed: A test is made to see if $|\lambda_{OLD} - \lambda_{OLDEST}| : \text{Min}$
 $(\lambda_{OLD}, \lambda_{OLDEST})$ is less than 1%. If it is, λ 's are sufficiently close:
 set λ_{NEW} to λ_{OLD} or λ_{OLDEST} , depending on which gave an availability
 above target. Else, use a binary search; i.e. $\lambda_{NEW} = 1/2 (\lambda_{OLD} + \lambda_{OLDEST})$.

SEARCH PROCESS



* For experimental runs, to get better compatibility with LP runs, a FAC of 1.02 was used if $AV_{OLD} < TAR$ and $|AV_{OLD} - TAR| > 0.3\%$.

REFERENCES

- 1 Everett, H., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," Operations Research, Vol 11, 1963, p. 399-417.
- 2 Hadley, G., Linear Programming, Addison-Wesley, Reading, MA, 1962.
- 3 Hadley, G. and T. M. Whitin, Analysis of Inventory Systems, Prentice Hall, Englewood Cliffs, NJ, 1963.
- 4 Lasdon, L. S., Optimization Theory for Large Systems, MacMillan, New York, 1970.
- 5 O'Malley, T. J., "A Method of Treating Common Recoverable Components in the LMI Military Essentiality Model," Logistics Management Institute, Washington, DC, 1976.

DISTRIBUTION

COPIES

<u>1</u>	Deputy Under Sec'y of the Army, ATTN: Office of Op Resch
	Headquarters, US Army Materiel Development & Readiness Command
<u>1</u>	DRCPA-S
<u>1</u>	DRCMS
<u>1</u>	DRCDMR
<u>1</u>	DRCPS-M
<u>1</u>	DRCPS-P, ATTN: Ms. Lamb
<u>1</u>	DRCPS-S
<u>1</u>	DRCMM
<u>1</u>	DRCMM-R
<u>1</u>	DRCMM-S
<u>1</u>	DRCMM-RS
<u>1</u>	DRCMM-M
<u>1</u>	DRCMM-MS
<u>1</u>	DRCMM-E
<u>1</u>	DRCDE
<u>1</u>	DRCPM
<u>1</u>	DRCQA
<u>1</u>	Dep Chf of Staff for Logistics, ATTN: DALO-SMS, Pentagon,
	Wash., DC 20310
<u>1</u>	Dep Chf of Staff for Logistics, ATTN: DALO-SML, Pentagon,
	Wash., DC 20310
<u>1</u>	Commandant, US Army Logistics Mgt Center, Ft. Lee, VA 23801
<u>1</u>	Office, Asst Sec'y of Defense, ATTN: MRA&L-SR, Pentagon,
	Wash., DC 20310
<u>2</u>	Defense Logistics Studies Info Exchange, DRXMC-D
<u>10</u>	Defense Technical Information Center, Cameron Station, Alexandria,
	VA 22314
	Commander, USA Armament Materiel Readiness Cmd, Rock Island, IL 61201
<u>1</u>	ATTN: DRSAR-MM
<u>1</u>	ATTN: DRSAR-SA
	Commander, USA Communications & Electronics Materiel Readiness Cmd,
	Ft. Monmouth, NJ 07703
<u>1</u>	ATTN: DRSEL-MM
<u>1</u>	ATTN: DRSEL-SA
	Commander, USA Missile Command, Redstone Arsenal, AL 35898
<u>1</u>	ATTN: DRSMI-S
<u>1</u>	ATTN: DRSMI-D
	Commander, USA Troop Support & Aviation Materiel Readiness Command,
	St. Louis, MO 63120
<u>1</u>	ATTN: DRSTS-SP
<u>1</u>	ATTN: DRSTS-SPSS
<u>1</u>	ATTN: DRSTS-BA(1)
<u>1</u>	Commander, US Army Tank-Automotive Command, Warren, MI 48090
<u>1</u>	ATTN: DRSTA-F
<u>1</u>	ATTN: DRSTA-S
<u>1</u>	Commander, US Army Armament Research & Development Cmd, ATTN: DRDAR-SE,
	Dover, NJ 07801

COPIES

<u>1</u>	Commander, US Army Aviation Research & Development Cmd, 4300 Goodfellow Blvd, St. Louis, MO 63120
<u>1</u>	Commander, US Army Aviation Research & Development Cmd, ATTN: DRDAV-BD (Marl Glenn), 4300 Goodfellow Blvd, St. Louis, MO 63120
<u>1</u>	Commander, US Army Communications Research & Development Command, ATTN: DRSEL-SA, Ft. Monmouth, NJ 07703
<u>1</u>	Commander, US Army Electronics Research & Development Command, ATTN: DRDEL-AP, Adelphi, MD 20783
<u>1</u>	Commander, US Army Mobility Equipment Research & Development Cmd, ATTN: DRDME-O, Ft. Belvoir, VA 22060
<u>1</u>	Commander, US Army Natick Research & Development Command, ATTN: DRXNM-O, Natick, MA 01760
<u>1</u>	Commander, US Army Logistics Center, Ft. Lee, VA 23801
<u>1</u>	Commander, US Army Logistics Evaluation Agency, New Cumberland Army Depot, New Cumberland, PA 17070
<u>1</u>	Commander, US Air Force Logistics Cmd, WPAFB, ATTN: AFLC/XRS, Dayton, Ohio 45433
<u>1</u>	US Navy Fleet Materiel Support Office, Naval Support Depot, Mechanicsburg, PA 17055
<u>1</u>	Mr. James Prichard, Navy SEA Systems Cmd, ATTN: PMS 3061, Dept of US Navy, Wash., DC 20362
<u>1</u>	George Washington University, Inst of Management Science & Engr., 707 22nd St., N.W., Wash., DC 20006
<u>1</u>	Naval Postgraduate School, ATTN: Dept of Opns Anal, Monterey, CA 93940
<u>1</u>	Air Force Institute of Technology, ATTN: SLGQ, Head Quantitative Studies Dept., Dayton, OH 43433
<u>1</u>	US Army Military Academy, West Point, NY 10996
<u>1</u>	Librarian, Logistics Mgt Inst., 4701 Sangamore Rd., Wash., DC 20016
<u>1</u>	University of Florida, ATTN: Dept of Industrial Systems Engr., Gainesville, FL 32601
<u>1</u>	RAND Corp., ATTN: S. M. Drezner, 1700 Main St., Santa Monica, CA 90406
<u>1</u>	US Army Materiel Systems Analysis Activity, ATTN: DRXS-CL, Aberdeen Proving Ground, MD 21005
<u>1</u>	Commander, US Army Logistics Center, ATTN: Concepts & Doctrine Directorate, Ft. Lee, VA 23801
<u>1</u>	ALOG Magazine, ATTN: Tom Johnson, USALMC, Ft. Lee, VA 23801
<u>1</u>	Commander, USDRC Automated Logistics Mgt Systems Activity, P.O. Box 1578, St. Louis, MO 63188
<u>1</u>	Director, DARCOM Logistics Systems Support Agency, Letterkenny Army Depot, Chambersburg, PA 17201
<u>1</u>	Commander, Materiel Readiness Support Activity, Lexington, KY 40507
<u>1</u>	Director, Army Management Engineering Training Agency, Rock Island Arsenal, Rock Island, IL 61202
<u>1</u>	Defense Logistics Agcy, ATTN: DLA-LO, Cameron Sta, Alexandria, VA 22314
<u>1</u>	Dep Chf of Staff (I&L), HQ USMC-LMP-2, ATTN: MAJ Sonneborn, Jr., Wash., DC 20380
<u>1</u>	HQ, Dept of the Army, (DASG-HCL-P), Wash., DC 20314
<u>1</u>	Operations Research Center, 3115 Etcheverry Hall, University of California, Berkeley, CA 94720

COPIES

1 Dr. Jack Muckstadt, Dept of Industrial Engineering & Operations
Research, Upson Hall, Cornell University, Ithaca, NY 14890

1 Prof Herbert P. Galliher, Dept of Industrial Engineering,
University of Michigan, Ann Arbor, MI 48104

1 Scientific Advisor, ATCL-SCA, Army Logistics Center, Ft. Lee, VA
23801

1 Commandant, USA Armor School, ATTN: MAJ Harold E. Burch,
Leadership Dept, Ft. Knox, KY 40121

1 Prof Robert M. Stark, Dept of Stat & Computer Sciences,
University of Delaware, Newark, DE 19711

1 Prof E. Gerald Hurst, Jr., Dept of Decision Science, The Wharton
School, University of Penna., Phila., PA 19174

1 Logistics Studies Office, DRXMC-LSO, ALMC, Ft. Lee, VA 23801

1 Procurement Research Office, DRXMC-PRO, ALMC, Ft. Lee, VA 23801

1 Dept of Industrial Engr. & Engr. Management, Stanford University,
Stanford, CA 94305

1 Commander, US Army Communications Command, ATTN: Dr. Forry,
CC-LOG-LEO, Ft. Huachuca, AZ 85613

1 Commander, US Army Test & Evaluation Cmd, ATTN: DRSTE-SY,
Aberdeen Proving Ground, MD 21005

1 Prof Harvey M. Wagner, Dean, School of Business Adm, University
of North Carolina, Chapel Hill, NC 27514

1 DARCOM Intern Training Center, Red River Army Depot, Texarkana,
Tx 75501

1 Prof Leroy B. Schwarz, Dept of Management, Purdue University,
Krannert Bldg, West Lafayette, Indiana 47907

1 US Army Training & Doctrine Command, Ft. Monroe, VA 23651

1 Operations & Inventory Analysis Office, NAVSUP (Code 04A) Dept
of Navy, Wash., DC 20376

1 US Army Research Office, ATTN: Robert Launer, Math. Div.,
P.O. Box 12211, Research Triangle Park, NC 27709

1 Prof William P. Pierskalla, 3641 Locust Walk CE, Philadelphia,
PA 19104

1 US Army Materiel Systems Analysis Activity, ATTN: DRXSY-MP,
Aberdeen Proving Ground, MD 21005

1 Air Force Logistics Management Center, Gunter Air Force Station, AL 36144
ATTN: AFLMC/LCY

1 ATTN: AFLMC/XRP, Bldg. 205

1 Engineer Studies Center, 6500 Brooks Lane, Wash., DC 20315

1 US Army Materiel Systems Analysis Activity, ATTN: Mr. Herbert Cohen,
DRXSY-MP, Aberdeen Proving Ground, MD 21105

1 Commander, US Army Missile Cmd, ATTN: Ray Dotson, DRSMI-DS, Redstone
Arsenal, AL 35898

1 Prof Peter Kubat, Graduate School of Mgmt, University of Rochester,
Rochester, NY 14627

1 Logistics Management Institute, 4701 Sangamore Rd., Wash., DC 20016
ATTN: Mike Slay/T. J. O'Malley

1 Asst Sec'y of Defense (MRA&L), ATTN: Russ Shorey, Room 2B322,
Pentagon, Wash., DC 20310