

MATHEMATICAL MODEL AND OPTIMIZATION  
OF AN INTERLEAVING WAREHOUSE LAYOUT

by

Bhoopatsinh Udaysinh Jadeja

Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Industrial Engineering and Operations Research

APPROVED:

S. Balachandran, Chairman

G. V. Loganathan

S. C. Sarin

November, 1985

Blacksburg, Virginia

MATHEMATICAL MODEL AND OPTIMIZATION  
OF AN INTERLEAVING WAREHOUSE LAYOUT

by

Bhoopatsinh Udaysinh Jadeja

(ABSTRACT)

This research is devoted to the development of a mathematical model for an Interleaving Warehouse Layout. The space allocated to the items in the warehouse is most commonly determined on the basis of inventory cost of the items. Once the space requirements for the items are computed, the actual assignment of items to locations in the warehouse is carried out independently. An Interleaving Warehouse Layout is presented in this research to incorporate both reorder quantity and location of each item in a single comprehensive mathematical model of the warehouse. The advantage of this approach is that it considers the quantity and location problems encountered in a warehouse layout simultaneously.

The mathematical model developed for the warehouse layout is optimized utilizing the computer code GRG2.5. The numerical results for two warehouses are summarized, discussed and compared with the data available in the literature.

## ACKNOWLEDGEMENTS

The author wishes to express his appreciation to the following individuals for their aid and encouragement during the completion of this thesis:

Dr. S. Balachandran, the chairman of his committee for providing excellent suggestions, invaluable advice and consultations.

The other members of his graduate committee, Dr. G. V. Loganathan and Dr. S. C. Sarin for their assistance, cooperation, constructive criticisms and advice.

Dr. R. D. Dryden, Dr. W. J. Fabrycky, and Dr. J. W. Schmidt for providing assistance in clearing numerous administrative hurdles.

Mrs. Paula Davis for her efficient and expert typing. Without her help, this thesis would never have been completed on time.

## TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT . . . . .	ii
ACKNOWLEDGEMENTS . . . . .	iii
 CHAPTER I: INTRODUCTION	
1.1 Introduction . . . . .	1
1.2 Problem Description . . . . .	2
1.3 An Overview of the Thesis . . . . .	3
1.4 Research Contributions . . . . .	5
 CHAPTER II: BACKGROUND LITERATURE REVIEW	
2.1 Introduction . . . . .	6
2.2 Economic Order Quantity . . . . .	7
2.3 The ABC Curve . . . . .	12
2.4 Stock Location . . . . .	13
2.4.1 Linear Programming Model of the Warehouse . . . . .	15
2.4.2 Comparison of the COI and LP Method . . . . .	17
2.5 Storage Assignment Rules Using the Out and Back Order Picking Method . . . . .	18
2.5.1 Continuous Distance and Turnover Functions . . . . .	19
2.6 Storage Assignment Using the Interleaving Order Picking Method . . . . .	25
2.7 The Warehouse Layout Problem . . . . .	29
2.7.1 The Warehouse Layout Problem Formulated as a General Assignment Problem . . . . .	30
2.7.2 The Discrete Warehouse Layout Problem Using The Factoring Assumption . . . . .	33
2.7.3 Problem Solution Using Factoring Assumption . . . . .	34
2.8 Quantity and Location Problems Considered Simultaneously . . . . .	36
2.8.1 Determination of Reorder Quantity and Stock Location Independently . . . . .	37
2.8.3 Solution Technique . . . . .	39
2.9 Summary . . . . .	44
 CHAPTER III: THE MODELING APPROACH	
3.1 Introduction . . . . .	46
3.2 The Model Assumptions . . . . .	46
3.3 Profile of Related Problems . . . . .	52
3.4 Model Formulation . . . . .	53
3.4.1 The Modified Location ABC Curve . . . . .	54
3.4.2 Formulation of Travel Distances . . . . .	59
3.4.2.1 Travel to the Storage Location . . . . .	59
3.4.2.2 Interleave Travel Distance . . . . .	59
3.4.2.3 Retrieval Travel Distance . . . . .	60
3.4.3 Order Picking Cost Expressions . . . . .	60

3.4.3.1	Storage Travel Costs . . . . .	61
3.4.3.2	Interleaving Travel Costs . . . . .	61
3.4.3.3	Retrieval Travel Costs . . . . .	62
3.4.4	Inventory Costs . . . . .	63
3.4.5	The Total Cost Equation . . . . .	63
3.4.6	Constraints . . . . .	64
3.5	Summary and Conclusions . . . . .	65
CHAPTER IV: PROPOSED SOLUTION METHODOLOGY		
4.1	Introduction . . . . .	66
4.2	Description of the Computational Logic . . . . .	66
4.2.1	Main Calling Program . . . . .	66
4.2.2	Layout Subroutine . . . . .	71
4.3	GRG2.5 Computer Code . . . . .	71
4.3.1	Subroutine GCOMP . . . . .	74
4.3.2	GRGSUB Optimization Routine . . . . .	75
4.4	Summary and Conclusions . . . . .	76
CHAPTER V: RESULTS, CONCLUSIONS AND RECOMMENDATIONS		
5.1	Introduction . . . . .	78
5.2	Profile of Input Parameters . . . . .	78
5.3	GRGSUB Optimization Routine . . . . .	82
5.3.1	Bounds for Variables and Functions . . . . .	83
5.3.2	Argument Values for GRGSUB Routine . . . . .	84
5.4	Results for the 5 Item Problem . . . . .	85
5.5	Results for the 25 Item Problem . . . . .	88
5.5.1	Comparison with Solutions from Literature . . . . .	93
5.6	Conclusions . . . . .	94
5.7	Recommendations for Further Research . . . . .	95
REFERENCES	. . . . .	97
VITA	. . . . .	99

LIST OF FIGURES

	<u>Page</u>
Figure 2.2a. Inventory Level Over Time . . . . .	10
Figure 2.2b. Order Preparation Cost Curve . . . . .	10
Figure 2.2c. Holding Cost Curve . . . . .	11
Figure 2.2d. Total Variable Cost Curve . . . . .	11
Figure 2.3. The ABC Curve . . . . .	14
Figure 2.5a. Continuous Representation of Storage Rack Hausman, Schwarz and Graves[10] . . . . .	20
Figure 2.5b. Two-Class Storage Assignment . . . . .	24
Figure 2.8. Wilson's Flow Chart . . . . .	43
Figure 3.1. Warehouse Layout . . . . .	47
Figure 3.2. Unique Address Classification . . . . .	49
Figure 3.3. Preference Classification . . . . .	50
Figure 4.1. Flowchart of General Logic for One Cycle of Optimization of the Interleaving Warehouse Layout Model . . . . .	67
Figure 4.2. Flowchart of General Logic for the LAYOUT Routine . . . . .	68
Figure 4.3. Flowchart for General Logic for the GCOMP Routine . . . . .	69
Figure 4.4. Flowchart for General Logic of One Optimization Cycle of GRGSUB Routine . . . . .	70
Figure 5.4. Cost Chart for 5 Item Problem . . . . .	89
Figure 5.5. Cost Chart for 25 Item Problem . . . . .	92

LIST OF TABLES

	<u>Page</u>
Table 5.2.1. Input Data for 5 Item Problem . . . . .	80
Table 5.2.2. Input Data for 25 Item Problem . . . . .	81
Table 5.4.1. Results for the 5 Item Problem - I . . . . .	86
Table 5.4.2. Results for the 5 Item Problem - II . . . . .	87
Table 5.5.1. Results for the 25 Item Problem - I . . . . .	90
Table 5.5.2. Results for the 25 Item Problem - II . . . . .	91

# CHAPTER I

## INTRODUCTION

### 1.1 Introduction

A large fraction of the Gross National Product of the United States of America is locked up in inventory during each year. Therefore, mathematical models for optimizing the total warehouse operating cost are very useful. The cost of handling goods in a warehouse can amount to over half of the total operating expenses. The total revenue could be significantly increased by minimizing the cost of handling inventory in the warehouse [1]. The two major factors that determine the material handling expense in most warehouses are the order picking method and the location of inventory items. This research is devoted to the development of a mathematical model which incorporates both reorder quantity and location of each item simultaneously for optimizing Interleaving Warehouse Layout.

An item arriving at the warehouse is moved out of the receiving dock to it's storage location and is then retrieved to be delivered to the shipping dock. This routing of the items through the warehouse has an associated handling cost which is determined by the picking discipline used. Although a variety of picking disciplines can be chosen based on the characteristics of the items stored, warehouse design and distribution of orders, the out and back order picking method and interleaving order picking method are used most widely. In the out and back order picking method, only one visit is made to a storage or retrieval location before returning to the shipping dock

(input/output point). On the other hand the interleaving order picking method visits both a storage and a retrieval location between returns to the input/output point.

The space requirements for items stored in the warehouse are determined by their respective order quantities. Hence, reducing reorder quantities while still meeting their demand, will reduce space requirements for the items stored. This allows the items to be moved closer to the input/output point, which in turn reduces their order picking costs. The reduction in travel costs achieved by decreasing the reorder quantity is partly offset by the increase in inventory costs as the EOQ reorder quantity is no longer utilized. Thus in order to achieve a minimum total cost, tradeoffs in handling costs among various stock arrangements, order picking disciplines and reorder quantities have to be evaluated.

## 1.2 Problem Description

The order picking method and space requirement for the inventory items are the two primary aspects of a typical warehouse layout problem. Generally, space requirements of the inventory items are determined by simple economic order quantity model. When these space requirements are assumed to be available, storage locations for the inventory items are primarily dictated by the travel costs. Wilson [2] points out that it is essential to solve the quantity and location problems simultaneously so that the total cost can be minimized. Thus, both order picking costs and inventory costs affect the overall warehouse layout and should not be determined independently.

The order picking method has a direct bearing on the total cost as it determines the amount of travel within the warehouse. The out and back order picking method, in which only one location is visited between returns to the input/output point, achieves lower throughput than that allowed by interleaving method [3]. Interleaving systems, also known as "dual command" or "dual address" systems, allow visits to two storage locations in a round trip. Thus in an interleaving system the order picking vehicle would visit a storage location, store the item it carried, interleave travel to another location, perform a retrieval operation and return to the input/output point.

In this research an interleaving warehouse layout model is presented to include both reorder quantity and location of each item in a single mathematical model, while continuing to meet demand requirements. This model is solved using the computer code GRG2.5 [4]. The EOQ reorder quantities are used in the starting point for the optimization technique. The optimal reorder quantities thus obtained lower the total cost of the warehouse and increase the throughput of the warehouse.

### 1.3 An Overview of the Thesis

The research carried out for this thesis is organized and presented in the following chapters:

In Chapter II, the literature related to the problem chosen for this research is comprehensively reviewed. The review progresses from a discussion of basic economic order quantity model to the concept of subdivision of inventory by the ABC principle. Next, various stock

allocation rules and travel times for different storage assignment rules are considered. The warehouse layout problem is considered next, followed by discussion of solution procedure that incorporates quantity and location problems simultaneously.

The mathematical model for the warehouse layout is presented in Chapter III together with the detailed definitions of the variables and the different components of the total cost function. The assumptions made in formulating the mathematical model are listed and discussed in detail in this chapter. The assumptions emphasize the limitations of this particular study and also reveal that these assumptions are not very restrictive for the design of interleaving warehouse layout. A description of types of warehouse layouts that can be solved using proposed model is given next. Modified Location ABC curve which allows approximation of the probability of a particular location access is introduced. A formulation of travel distance between storage locations is considered next. The total cost equation is derived and additional information is provided as to how the mathematical model can be expanded, if necessary, to include constraints on some or all of the variables so that the mathematical model will be a constrained non-linear optimization problem.

In Chapter IV, the GRG2.5 computer code selected for solving the mathematical model in this research is presented. The options available in the GRG2.5 computer code are briefly reviewed and a solution methodology is presented for solving the mathematical model developed in this research using the GRG2.5 computer code.

In Chapter V, two warehouse layout problems are presented. Using the solution methodology presented and discussed in Chapter IV, numerical results obtained in this research will be summarized and compared with those obtained by Kyle [5].

#### 1.4 Research Contributions

(i) This research formulates the design of the interleaving warehouse layout as an optimization problem. This approach has not been utilized before. Thus, this research provides a new approach for interleaving warehouse layout.

(ii) Two warehouse layouts are selected and optimized using the mathematical model presented in this research and the computer code GRG2.5. The numerical results obtained in this research are compared with those available from the literature. This analysis of the numerical results establishes that the optimization methodology presented in this research is a viable tool for achieving efficient interleaving warehouse layout.

(iii) The limitations of this research are discussed in detail, suggestions are provided for modification of the mathematical model and recommendations are made for further research in this area.

## CHAPTER II

### BACKGROUND LITERATURE REVIEW

#### 2.1 Introduction

Warehouse layout models that identify efficient methods for assignment of inventory items to storage locations within the warehouse provide a very important tool to the warehouse management. Wilson [2] points out that traditionally, the amount of space allocated to each item in a warehouse is determined on the basis of inventory cost considerations. The inventory cost is usually obtained from simple economic order quantity inventory model. Assuming that the space requirements obtained from the economic order quantity considerations are available, the assignment of inventory items to various storage locations within the warehouse is then carried out independently.

One criterion for assigning inventory items to locations within the warehouse is the cube-per-order index (COI) rule, which was first proposed by Heskett. Harmatuck [9] has shown that the cube-per-order index rule produces an optimal solution to the linear programming formulation of the stock location problem where the objective is to minimize the order picking costs. The justification for focusing on order picking costs has been discussed by Kallina and Lynn [12].

The analytical results of Francis [13], and Mallette and Francis [14], support the use of dedicated storage to minimize travel time of the order picking vehicles based on single command cycles. Graves, Hausman, and Schwarz [3], propose a model for maximizing the effectiveness of dual command cycles, involving both storage and

retrieval during a trip, to minimize travel time. The problem of pairing storages and retrievals can be formulated as an assignment problem.

Francis and White [11] formulate the warehouse layout problem as an assignment problem. They also show that when there is only one dock, the warehouse layout problem may be solved by using the factoring assumption.

Wilson [2] assumes an out-and-back order picking method and the simple economic order quantity inventory model and demonstrates that the quantity and location problems must be considered simultaneously in order to achieve a minimum total cost (order picking cost plus inventory cost). He uses an iterative search procedure, based on a gradient search technique to obtain the optimal location and reorder quantity for each item.

This literature review proceeds from the economic order quantity inventory model, where inventory costs and reorder quantities are obtained, to consideration of the quantity and location problems simultaneously. The major emphasis in the literature review presented in this chapter is on the methods for producing a minimum total cost solution to the warehouse layout problem.

## 2.2 Economic Order Quantity

In a typical warehouse, it is necessary to procure greater quantities of any item than the quantities being shipped at the moment. This results in carrying certain inventory of that item with its associated inventory carrying cost. As the size of the replenishment

order determines the carrying cost, a balance between the ordering cost and the carrying cost needs to be achieved so that the total cost of the inventory is kept to a minimum. Figure 2.2a depicts a simplified inventory model which assumes that (i) shortages are not allowed, (ii) the demand for an item is uniform and (iii) the lead time is zero.

The total cost CT for a specified time period is the sum of the item cost CI for the period, the ordering cost CP for the period, and the holding cost CH for the period. Thus

$$CT = CI + CP + CH.$$

Expressing above component costs as the sums of their respective individual item costs,

$$CI = \sum CI_j, CP = \sum CP_j, CH = \sum CH_j$$

and

$$CT = \sum CT_j.$$

Now, the item cost per year for the jth item can be expressed in terms of cost per unit  $V_j$  and yearly demand  $D_j$ :

$$CI_j = V_j * D_j$$

Also, ordering cost for the item j for the period will be the cost per order  $Cp_j$  times the number of orders per year N, or

$$CP_j = Cp_j * N$$

If  $Q_j$  is the order quantity for the item j then,

$$N = D_j / Q_j,$$

so that

$$CP_j = Cp_j * D_j / Q_j.$$

With the assumptions made about the model, the time interval  $t$  begins with  $Q_j$  units in stock and ends with none, so that the average inventory during the cycle is  $Q_j/2$ . If  $Ch_j$  is the yearly holding cost per unit of item  $j$ , then

$$CH_j = Ch_j * Q_j/2,$$

and the total cost for the period to provide  $Q_j$  items is

$$CT_j = V_j * D_j + Cp_j * D_j/Q_j + Ch * Q_j/2.$$

Now, the graph of ordering cost  $CP_j$  against order quantity  $Q_j$  shows that  $CP_j$  decreases as the lot size is increased (Fig. 2.2b). Similarly Figure 2.2c shows the linear relationship between the holding cost  $CH_j$  and the lot size  $Q_j$ , where the holding cost increases as the lot size is increased.

The curve for the variable cost  $CV_j$ , where cost  $CV_j = CP_j + CH_j$ , is shown in Figure 2.2d. This total variable cost curve is convex. Hence we can obtain the minimum economic lot size by differentiating  $CV_j$  with respect to  $Q_j$ , equating the resulting expression to zero and solving for  $Q_j$ . Thus,

$$CV_j = Cp_j * D_j/Q_j + Ch_j * Q_j/2$$

and

$$\frac{\partial CV_j}{\partial Q_j} = -(Cp_j * D_j/Q_j^2) + Ch_j/2 = 0$$

$$Q_j = [2 Cp_j * D_j/Ch_j]^{1/2}.$$

The second order derivative of  $CV_j$  with respect to  $Q_j$  is always positive. Therefore the above  $Q_j$  minimizes the total variable cost  $CV_j$ .

Inventory  
Level of  
Item j

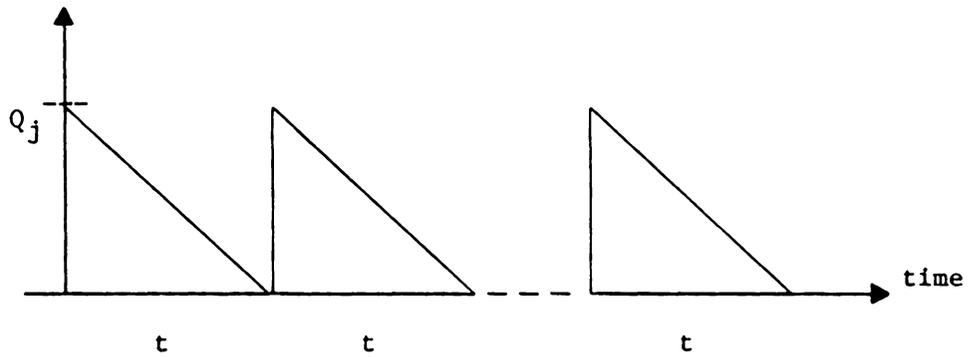


Figure 2.2a. Inventory Level Over Time.

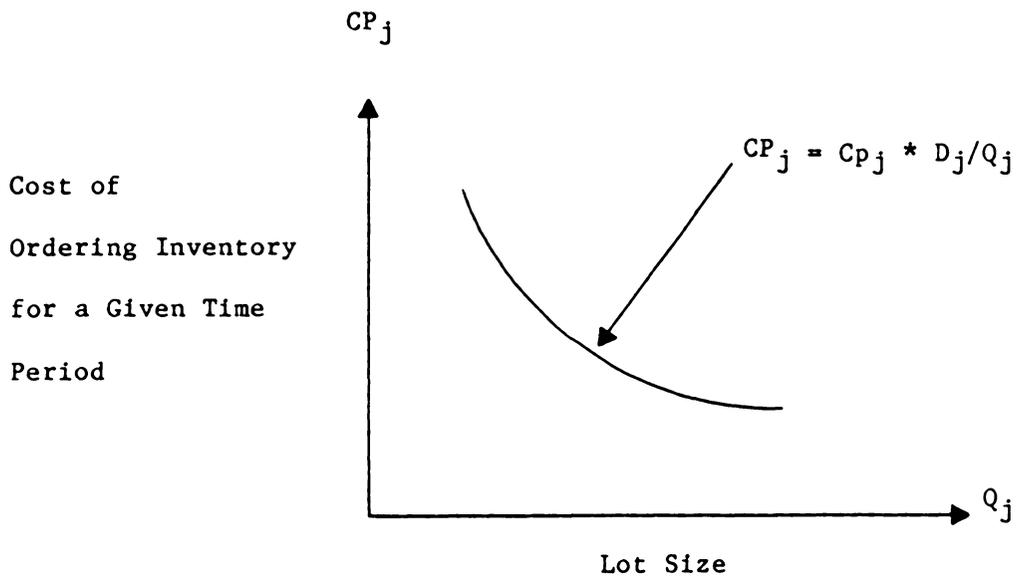


Figure 2.2b. Order Preparation Cost Curve.

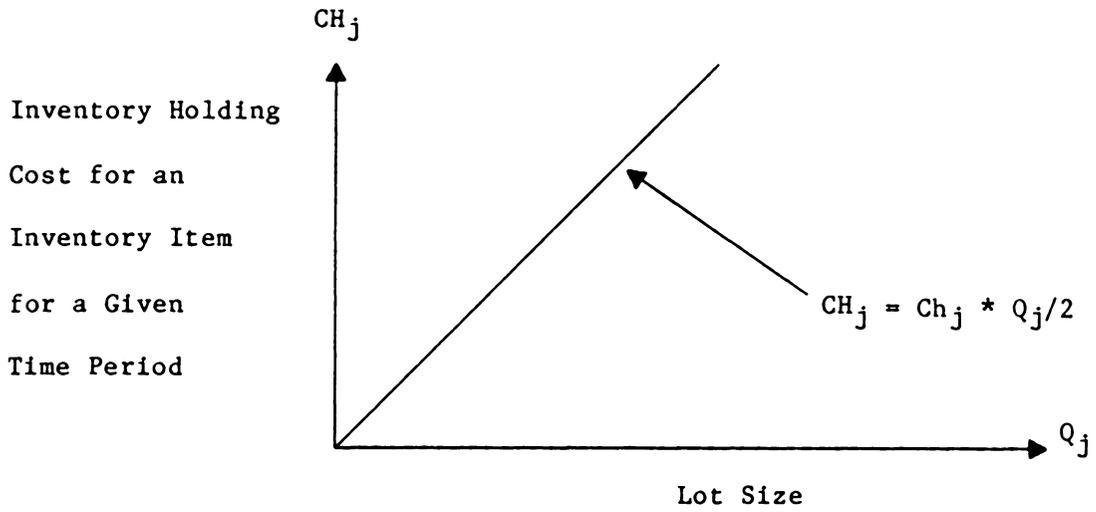


Figure 2.2c. Holding Cost Curve.

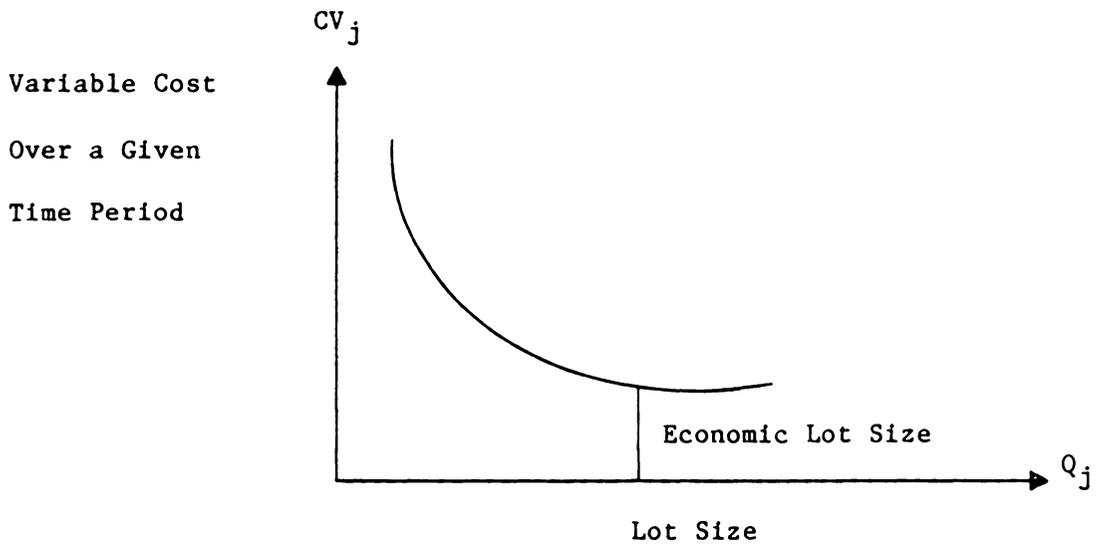


Figure 2.2d. Total Variable Cost Curve.

The optimal total cost for item j is:

$$CT_j = V_j * D_j + (Cp_j * D_j * Ch_j/2)^{1/2} + (Ch_j * Cp_j * D_j/2)^{1/2}.$$

Adding up the above inventory cost for all items stored, the total inventory cost per period is obtained for the whole warehouse inventory.

### 2.3 The ABC Curve

The ABC principle is a very effective and useful tool for efficient control of the inventory system within a warehouse. The importance of the principle lies in the fact that the principle emphasizes the relative importance of different items in the warehouse. It points out the need for sub-dividing the total inventory for the purpose of more efficient inventory management.

If the importance of an inventory item is measured by the inventory investment or the dollar useage of that item, a system that provides uniform control over all items will over control the low value items while providing adequate control for the high value items. On the other hand, a uniform control system that provides adequate control for the low value items will under control the high value items. Thus, Zimmerman [6] suggests that efficient management occurs when the amount of management effort is directly proportional to the importance of the item being managed.

Lee and Dobler [7] give the following procedure for conducting an ABC analysis of the inventory items when demand characteristics of individual items, as measured by their average inventory investment or dollar useage, are known. All items are ranked in the order of their

demand. The demand for each individual item is then expressed as a percentage of the total. The ABC curve can now be plotted by successively adding the individual percentages for each item and plotting these cumulative percentages of the demand against the percentage of items inventoried as shown in Figure 2.3. From this curve, we can identify which items make up any specified percent of the investment.

To conclude the ABC analysis of the above curve, the inventory items are divided into three classes, A, B and C. Class A contains the items that deserve greatest attentions from management as a small percentage of the total items account for a large percentage of the total demand. For class B items, a moderate percentage of the items account for a moderate portion of the total demand, while for class C items a large percentage of the items account for a small portion of the total demand.

Peterson and Silver [8] point out that other criteria for ABC types of classifications are sometimes used. In some large volume distribution warehouses, allocation of space is made using an ABC curve based on cubic feet per unit required. Thus, items with a low ratio of required cubic footage to the number of daily orders are placed nearest the input/output point so that the greatest stock volume moves the shortest distance.

#### 2.4 Stock Location

The location of the items in a warehouse determines the order picking costs associated with assembling orders and transporting them

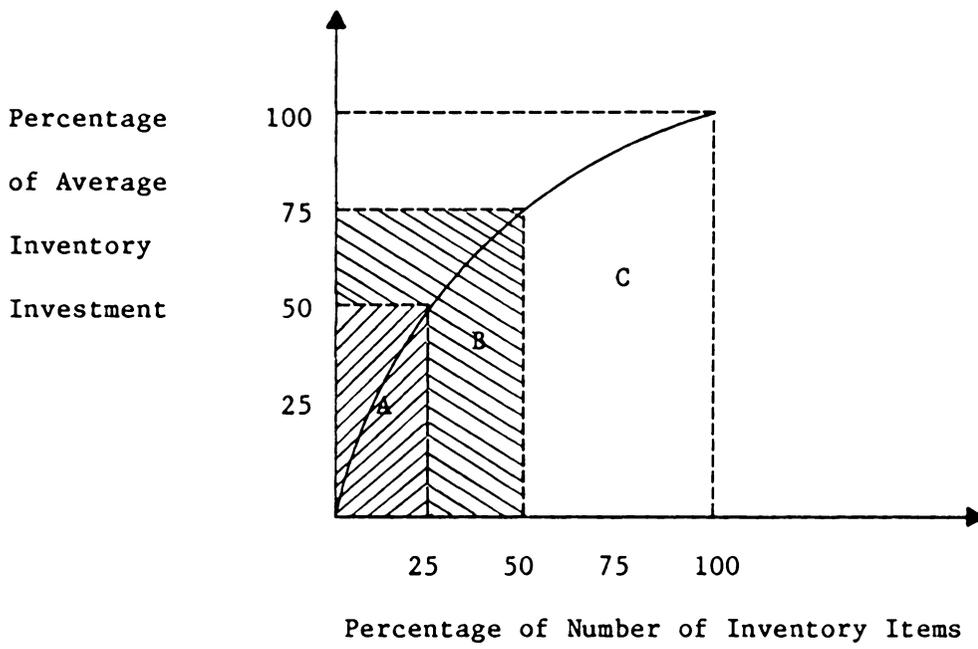


Figure 2.3. The ABC Curve.

to the input/output point. Thus an efficient method of locating items within a warehouse will give an improved warehouse layout.

One of the most common layout methods, as pointed out by Ballou [1], is one based on the product popularity. This method assumes that the total distance traveled within the warehouse determines the total handling cost. Hence, locating faster moving items near the input/output point and the slower moving items to the rear of these items will minimize the total travel distance, thereby minimizing the total handling costs. This reduces the distance traveled per trip, as the items requiring greater number of trips to satisfy demand are located near the input/output point.

A second layout method commonly used is based on the volume of the items stored. In this method, smaller items are located nearest to the input/output point and larger items are located to the rear of these items. This way, a greater number of small items, which will satisfy a greater percentage of demand can be located near the input/output point, thereby reducing the total distance traveled.

A third method, called cube-per-order index rule, combines both turnover rate and size of every item. Items with a low ratio of cubic footage (or the floor space) required to the number of daily orders for the item, are placed nearest the input/output point. Thus, the greatest stock volume will move the shortest distance.

#### 2.4.1 Linear Programming Model of the Warehouse

This model, as presented by Harmatuck [9] minimizes the sum of the handling costs for all items in the warehouse assuming an out-and-back

order picking method. In the out-and-back order picking method, the order picking vehicle performs either a storage or a retrieval operation in one round trip from the input/output point and is able to pick one item during each round trip. The amount of product allocated to any one subarea cannot exceed the capacity of the subarea and enough space needs to be allocated to each item so that the demand requirements for the items are met.

The  $n$  items to be located are characterized by:

$DD_j$  = the time period for which the demand for item  $j$  should be satisfied. This is also the time between two successive replenishments of item  $j$

$VU_j$  = volume of space required by a unit of item  $j$

$AOS_j$  = average order size of an item  $j$

$OP_j$  = the number of orders per period for item  $j$ .

The  $m$  subarea are characterized by:

$CPA_i$  = the volume capacity of location  $i$

$CM_i$  = the cost of moving an order of any item from location  $i$  to the input/output point.

The locations are ordered so that  $CM_i < CM_{i+1}, \dots$  where  $i = 1, 2, \dots, m-1$ .

Haskett proposes a cube-per-order index rule for deciding the location for each item. The index for item  $j$  is defined as

$$COI_j = VU_j * AOS_j * DD_j.$$

The number of units of item  $j$  assigned to location  $i$ ,  $X_{ij}$  is determined by the minimization of the following objective function:

$$Z = \sum_{i=1}^m \sum_{j=1}^n (CM_i / AOS_j * DD_j) * X_{ij}.$$

Subject to

$$\sum_{j=1}^n VU_j * X_{ij} < CAP_i, \quad i = 1, 2, \dots, m.$$

$$\sum_{i=1}^m X_{ij} = AOS_j * OP_j * DD_j \quad j = 1, 2, \dots, n.$$

$$X_{ij} > 0 \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n.$$

#### 2.4.2 Comparison of the COI and LP Methods

Harmatuck [9] has shown that the cube-per-order index (COI) rule produces an optimal solution to the linear programming formulation of the stock location problem based on the following assumptions:

1) For picking purposes, an order consists of a quantity of a single item retrieved by means of a simple out-and-back selection procedure.

2) The amount of space required for the  $n$  items may be allocated in any manner. More than one item may be stored in a particular location and an item may be split among several locations.

3) The cost of moving an order of an item from a particular location to the shipping area is identical for all items and depends only on the location of the item.

4) The system will continue to operate in steady state indefinitely.

To demonstrate that the two methods produce the same optimal solution, we can show that the violation of the COI rule will never lower the total costs. If item 1 is located in subarea A and item 2 is located in a more remote subarea B, then by the definition of the

cube-per-order index rule,

$$COI_1 < COI_2$$

$$CM_A < CM_B.$$

For each cubic foot of item 1 and item 2 exchanged between subareas, the change in cost for item 1 is

$$(CM_B - CM_A)/COI_1$$

and the change in cost for item 2 is

$$(CM_A - CM_B)/COI_2.$$

Summing and simplifying the above two expressions, the total additional cost for the two items is obtained as

$$(CM_B - CM_A) * (1/COI_1 - 1/COI_2),$$

which is non-negative. Therefore, a violation of the COI rule will never lower costs.

## 2.5 Storage Assignment Rules Using the Out and Back Order Picking Method

Randomized storage assignment is one of the most widely used storage assignment rules in warehouses using computer controlled inventory systems. Incoming items are assigned to a pallet and the contents of the pallet are then communicated to a minicomputer. The computer assigns the pallet to a location in the warehouse and records the assigned location. When there is a demand for an item, the computer obtains the pallet location from its memory storage and directs an order picking vehicle to retrieve the pallet. The randomized storage system achieves a very efficient throughput because of reduced space requirement achieved as a result of random storage.

Consider a warehouse using randomized storage and an out-and-back order picking method in which storages alternate with retrievals. The closest available location will be the location at which the preceding retrieval has taken place, or a location of equal distance from the input/output point. Thus the closest open location rule, in which a pallet is stored in the closest location to the input/output point regardless of the pallet turnover rate, is approximated by the random storage system. In such a warehouse, Hausman, Schwarz and Graves [10] derived the expected one way travel time for random storage assignment. These researchers showed that the expected one way travel time is minimized when the turnover based assignment rule, that places the highest turnover rate pallet closest to the input/output point, is followed.

Next section deals with continuous representations of the distance and turnover functions. These continuous representations considerably reduce the difficulty of the analysis and yield approximately the same results as those obtained by the discrete analysis.

### 2.5.1 Continuous Distance and Turnover Functions

Hansman, Schwarz and Graves [10] derived the expressions for the one way travel time  $y(i)$  to the location  $i$  and the turnover rate  $\lambda(j)$  for location  $j$  under the following assumptions. The storage rack is assumed to be square where the co-ordinates  $x_1$  and  $x_2$  of each point are the horizontal and vertical travel times for the crane to reach that point. Without loss of generality, a unit of travel time is selected to yield a rack with unit sides, as shown in Figure 2.5a.

Following assumptions are made:

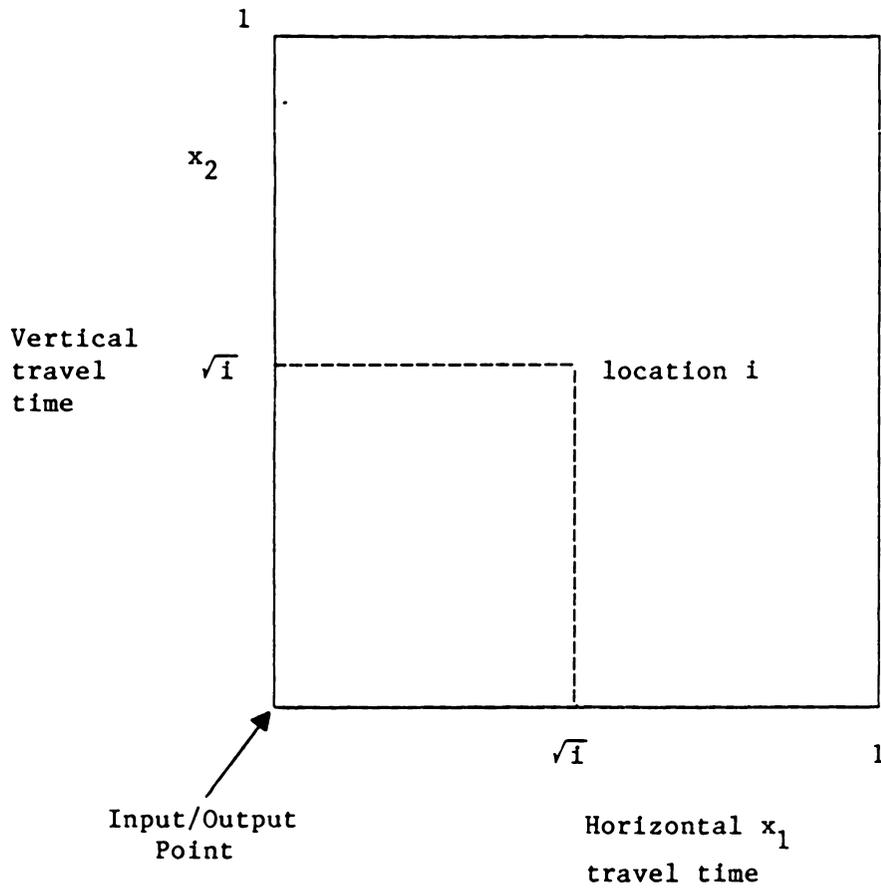


Figure 2.5a. Continuous Representation of Storage Rack  
Hausman, Schwarz and Graves [10].

1) Each pallet holds only one item type.

2) All storage locations are of the same size and all pallets are also of the same size. Thus all storage locations are candidates for storing any pallet load.

3) The system analyzed consists of a single crane serving a two sided aisle.

4) The system is bounded at the point where the crane transfers pallets to the input/output conveyor. The input/output is at the corner of the racks.

5) On each side of the aisle is a storage rack with  $R$  rows and  $C$  columns of locations. The crane travel is measured in time rather than distance. The crane can travel both horizontally and vertically simultaneously, its vertical and horizontal speeds are such that the time to reach the row most distant from the input/output point is same as the time to reach the most distant column. Actual time for the crane to load or unload a pallet at the input/output point or storage location is ignored.

6) An out-and-back order picking method is used.

7) The turnover frequency of each item is known and is constant throughout time. Turnover frequency is the number of times a given item requires storage and retrieval in certain time period. It is also the reciprocal of the average length of storage time for the item. This assumption is relaxed when class based turnover system is considered.

8) Short term dynamic considerations are ignored as the long-run average behavior of the system is investigated.

Hausman, Schwarz and Graves [10] derive the continuous distance  $y(i)$  as;

$$y(i) = i^{1/2} \quad 0 < i < 1.$$

The ABC principle for inventories and the basic EOQ model are utilized to estimate the distribution of the pallet turnover. The ABC curve is represented by the function  $G(i)$  where the demand is measured in full pallet loads.

$$G(i) = i^s \quad 0 < s < 1.$$

Now with  $D(i)$  = demand rate (pallets per unit time) of item  $i$  and  $Q(i)$  = the economic order quantity of item  $i$ ,

$$D(i) = s * i^{s-1} \quad 0 < i < 1$$

and  $Q(i) = (2K * D(i))^{1/2}$  where  $K$  is the ratio of ordering cost to the holding cost. The average inventory in pallet loads of item  $i$  is;

$$Q(i)/2 = (2K * D(i))^{1/2}/2$$

each with an average turnover of,

$$2D(i)/Q(i) = (2D(i)/K)^{1/2}.$$

Next, using the demand rate  $D(i)$  of item  $i$ , the total number of rack locations required for the average inventory of all items is calculated. These researchers then determine the time index  $i$  of the  $j$ th pallet  $i(j)$ . Substituting this into the demand rate  $D(i)$ , the turnover rate for the  $j$ th pallet,  $\lambda(j)$ , is derived as

$$\lambda(j) = (2D'_j/K)^{1/2} = (2s/K)(j^{(s-1)/(s+1)})$$

where

$$D'_j = D_{i(j)} = s_j^{2(s-1)/(s+1)}$$

the demand rate for the item on pallet  $j$ .

Utilizing the above results the expected one way travel time under random storage assignment, the expected one way travel time based on full turnover rate assignment and the percentage improvement over random storage assignment are computed.

Using different ABC curves and their corresponding  $s$  values, Hausman et al. [10] have shown that the improvement in the one way travel time based upon the full turnover rate assignment increases considerably as the skewness of the inventory distribution increases.

In practice, it is unrealistic to assume that the turnover rate of every pallet in the system is known, and is constant over time. The class-based turnover assignment rule permits this assumption. Racks and pallets are partitioned into  $K$  classes, based on the one way travel time  $y(i)$  and turnover rates. Pallets are assigned to a class of storage according to their class of turnover (i.e., highest turnover to closest location). Pallets are assigned randomly within any given class.

In a two class system, the symbol  $R$  represents the partitioning of the unit square into two classes as shown in Figure 2.5b. In this system, class 1 is used for higher turnover pallets and class 2 for lower turnover pallets. Hausman et al. [10] have derived the one way travel time under this two class system as a function of  $R$ .

Testing a wide range of inventory distribution they found that the two class system yields approximately 70% of potential gain of a full

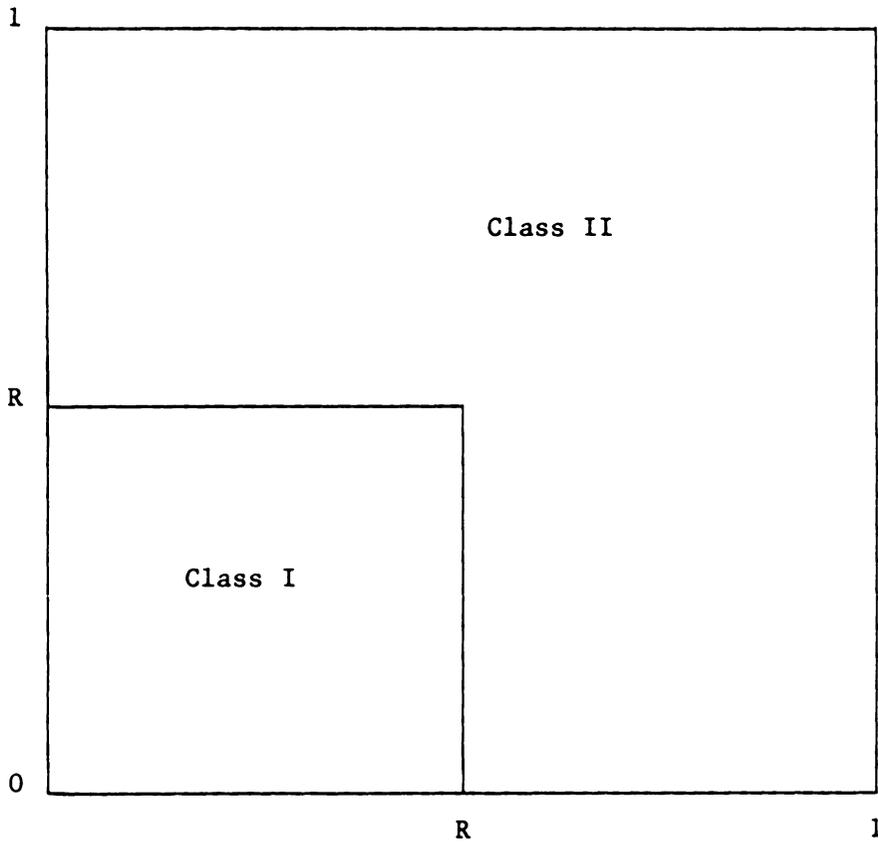


Figure 2.5b. Two-Class Storage Assignment.

turnover based system. The most highly skewed inventory distribution yields the largest improvement. For the three-class turnover-based system, a potential gain of approximately 85% of the full turnover-based system was found.

Hausman et al. [10] have developed a computer program to determine the relationship between the continuous model and the discrete model. The results indicate that as the measure of skewness of the inventory distribution increases the error involved in the continuous approximation increases significantly. Thus the true potential improvement is less than that indicated by the continuous approximation because of the increased importance given to the first few pallets and their discreteness relative to the continuous approximation.

This section thus shows that given an out-and-back order picking method, significant potential reduction of the crane travel times in an automatic warehouse system are possible using a class-based turnover assignment policy, rather than the closest open location policy.

## 2.6 Storage Assignment Using the Interleaving Order Picking Method

The interleaving systems, also known as dual command or dual address systems allow visits to up to two rack locations during a round trip from the input/output point. Thus, after completion of a given storage request, the order picking vehicle can move to the location of the next retrieval before returning to the input/output point. As can be expected, the interleaving systems permit potentially higher throughputs than those obtained by the out-and-back order picking methods. Only an optimal combined storage assignment/interleaving

policy will achieve the maximum throughput potential of the interleaving system.

The operating performance is measured in terms of the expected round trip travel time which is the expected time for the system to complete one storage and one retrieval operation before returning to the input/output point. For mandatory interleaving policies (MIL), the expected round trip travel time is twice the expected one way travel time plus the expected interleave time (the weighted sum of the order picking vehicle travel times between all storage locations, weighted by the probability of the corresponding interleaves). For the non-interleaving policy (NIL), the expected round trip travel time is four times the expected one way travel time [3].

Graves, Hausman and Schwarz [3] have studied the following storage assignment and interleaving rules:

- 1) Random Storage Assignment (RAN).
- 2) Class-based Storage Assignment (C2 or C3).

The C2 represents a two class system while C3 represents three class system.

- 3) Full-Turnover-Based Assignment (FULL).

The highest turnover pallet is assigned to the location closest to the input/output point. This is the limiting case of the class-based storage assignment with "square-L" boundaries in which each ranked location is a unique class. This case will estimate the maximum potential of the class-based storage assignment rules. The above storage assignment rules need to be considered together with the

following interleaving rules to obtain an optimal combined storage assignment/interleaving policy:

- 1) Noninterleaving Interleaving Rule (NIL).
- 2) Mandatory Interleaving with FCFS Queue Discipline of the Retrievals (MIL/FCFS).

The retrievals will be selected on a first come first serve (FCFS) basis from the queue of retrievals.

- 3) Mandatory Interleaving with Selection Queue of K retrievals (MIL/Q=K). Retrievals need not be chosen FCFS within the first K retrievals.

Graves et al. [3] have presented results for the following policies:

- 1) RAN/MIL/FCFS
- 2) FULL/MIL/FCFS
- 3) C2/MIL/FCFS
- 4) C3/MIL/FCFS
- 5) C2/MIL/Q=K
- 6) C3/MIL/Q=K

The best class-based storage assignment policies use classes whose outer boundaries are square in time, which are called "square-L" boundaries. This is due to the fact that crane velocities are square in time and all travel in NIL policies are to and from the input/output point.

In the interleaving systems, travel also occurs between the storage locations. Thus, the best class boundary shape in terms of expected round trip travel time need not be of the "square L" type.

The exact shape of the optimal boundaries are quite difficult to specify.

Graves et al. [3] investigated the sensitivity of the expected crane travel time to class boundary shape by making discrete evaluations of the "square-L", concentric-square and other boundary shapes. They were unable to find any boundary shape yielding expected round trip travel times lower than those resulting from the "square-L" boundaries.

The storage classifications  $R_1$  and  $R_2$  are chosen numerically to minimize the expected round trip travel time for different values of  $s$ . In the two class system,  $R_1$  and  $R_2$  indicate the partitioning points between classes.

The results differ most when the skewness of the turnover distribution is largest. Again this is due to the fact that increasing importance is given to the first few pallets and their "discreteness" relative to the continuous representation.

In general, class-based storage reduces round trip travel time over random storage. More rack locations are needed for the class-based storage than random storage due to the default probability (probability of not being able to store a pallet in the correct class) of the system. For a fixed number of rack locations, C2 and C3 systems will have higher default probabilities. Assuming an infinite channel queueing model, an increase of approximately 2% to 3% in rack size is needed for a 2-class system in order to keep the probability of default below 0.005; an increase of 4% to 5% is needed for the three class system.

All other factors remaining the same, MIL systems will reduce expected round trip travel times over NIL systems. In MIL systems, if a required queue for storage or retrieval is empty, round trip travel times may not involve interleaving. Thus, actual travel times would be weighted average of NIL and MIL performances.

By adopting the selection queue (no longer taking retrievals FCFS), system performance is further improved. Nearly all the benefits of this type of rule can be obtained by considering only the first few requests, thereby degradation in customer service is kept to a minimum.

## 2.7 The Warehouse Layout Problem

The warehouse layout problem consists of finding the optimal locations of the items stored, in order to minimize the order picking (handling) costs associated with assembling orders and transporting useful to think of  $S_j$  as specifying the location of item  $j$  and  $A_j$  as the area taken up by item  $j$ . The average distance that the item  $j$  travels between dock  $k$  and it's storage region is given by:

$$\sum_{i \in S_j} (1/A_j) * d_{ki}$$

The above assumes that for a given storage region for an item, the item is equally likely to travel between dock  $k$  and any grid square taken up by the item  $j$ . This expression has dimensions of average distance per trip between dock  $k$  and region  $i$ . Let  $w_{jk}$  be a known total cost per unit of average distance incurred in transporting item  $j$  between dock  $k$  and it's storage region for a given time period. Assuming items are stored on pallets,  $w_{jk}$  would be directly proportional to the number of pallet loads of item  $j$  moving between dock  $k$  and storage region of item

$j$  for some given time period. The total average cost of transporting item  $j$  between dock  $k$  and the storage region of item  $j$  may be written as;

$$w_{jk} * \sum_{i \in S_j} (1/A_j) * d_{ki}. \quad (2.71)$$

It follows that the dimensions of the above equation are cost per time period. The total average cost per time period due to transporting items to and from storage, written as a function of item location is;

$$F(S_1, S_2, \dots, S_n) = \sum_{j=1}^n \sum_{k=1}^p w_{jk} \left( \sum_{i \in S_j} (1/A_j) * d_{ki} \right). \quad (2.72)$$

Given that no more than one item can take up more than one grid square, the warehouse layout problem is to find storage assignments for all items to minimize the total cost equation given above. Since sets  $S_1, S_2, \dots, S_n$  determine the storage regions of the items, the collection of all sets  $(S_1, S_2, \dots, S_n)$  represents a warehouse layout. The problem is to obtain a layout that minimizes the total cost.

### 2.7.1 The Warehouse Layout Problem Formulated as a General Assignment Problem

To convert the layout problem to a generalized assignment problem, define the variable  $x_{ij}$  for  $i = 1, 2, \dots, m$  and  $j = 1, 2, \dots, n$  as;

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ takes up grid square } i \\ 0 & \text{if item } j \text{ does not take up grid square } i. \end{cases}$$

Assume that the total number of grid squares to be taken up by items is the same as the total number of grid squares, i.e.;

$$\sum_{j=1}^n A_j = m.$$

Since item  $j$  takes up a total of  $A_j$  grid squares,

$$\sum_{i=1}^m x_{ij} = A_j, \quad j = 1, 2, \dots, n.$$

Also since each grid square is also taken up by one item.

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, m.$$

Now  $S_j$  is the collection of all grid squares  $i$  for which  $x_{ij} = 1$ ; i.e.,  $S_j = \{i: x_{ij} = 1\}$ , so that equation (2.71) equals;

$$w_{jk} * \sum_{i=1}^m (1/A_j) * d_{ki} * x_{ij}.$$

The total cost expression can now be written as;

$$\sum_{j=1}^n \sum_{k=1}^p w_{jk} \left( \sum_{i=1}^m (1/A_j) * d_{ki} * x_{ij} \right).$$

Define  $c_{ij}$  as;

$$c_{ij} = (1/A_j) \sum_{k=1}^p w_{jk} * d_{ki} \quad j = 1, 2, \dots, n \text{ and } i = 1, 2, \dots, m.$$

The total cost expression can then be rewritten as;

$$\sum_{j=1}^n \sum_{i=1}^m c_{ij} * x_{ij}.$$

The generalized assignment version of the layout problem may now be stated as;

$$\begin{aligned} \text{minimize:} & \quad \sum_{j=1}^n \sum_{i=1}^m c_{ij} * x_{ij} \\ \text{subject to:} & \quad \sum_{i=1}^m x_{ij} = A_j \quad j = 1, \dots, n \\ & \quad \sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, m \\ & \quad x_{ij} \in \{0, 1\} \quad j = 1, \dots, n \text{ and } i = 1, \dots, m. \end{aligned}$$

If each  $A_j$  equals 1, then the above model is the assignment problem.

When some  $A_j$  are positive integers greater than 1, the above model

becomes a generalized assignment problem which is a special case of the

transportation problem.

Now, consider the case where the total number of grid squares to be taken up by items is less than the total number of grid squares available. That is;

$$\sum_{j=1}^n A_j < m.$$

In this case a "dummy item"  $n+1$  is defined to take up;

$$A_{n+1} = m - \sum_{j=1}^n A_j$$

grid squares. The constraints now change to;

$$\sum_{i=1}^m x_{ij} = A_j \text{ for } j = 1, \dots, n+1$$

$$\sum_{j=1}^{n+1} x_{ij} = 1 \text{ for } i = 1, \dots, m$$

$$x_{ij} \in \{0,1\} \text{ } j = 1, \dots, n+1 \text{ and } i = 1, \dots, m.$$

The objective function remains the same. In a general linear programming context, the inclusion of a dummy item is equivalent to the inclusion of a slack variable and one redundant constraint. In the transportation problem context, the inclusion of the dummy item is equivalent to the inclusion of a dummy destination. Another way of obtaining  $c_{ij}$  is to define the terms;

$$w_{n+1,1}; w_{n+1,2}, \dots, w_{n+1,p}$$

to be zero and use the equation;

$$c_{ij} = 1/A_j \sum_{k=1}^p w_{jk} * d_{ki} \text{ for } j = 1, \dots, n+1 \text{ and } i = 1, \dots, m$$

which uses the fact that the dummy item has no exchange with any dock.

That is,

$$w_{n+1,k} = 0 \text{ for } k = 1, \dots, p.$$

### 2.7.2 The Discrete Warehouse Layout Problem Using The Factoring Assumption

In this section, an assumption that permits a least cost layout to be found by a simpler procedure than the generalized assignment approach is made. To state the assumption, it is useful to make the following definition: the matrix  $W = (w_{jk})$  having  $n$  rows and  $p$  columns, will be said to factor if and only if there exist numbers  $u_1, u_2, \dots, u_m$  and  $v_1, v_2, \dots, v_p$  such that

$$w_{jk} = u_j * v_k \text{ for } j = 1, 2, \dots, n \text{ and } k = 1, 2, \dots, p.$$

The factoring assumption is always satisfied when there is only one dock.

With the factoring assumption, the total cost equation can be rewritten as;

$$F(S_1, S_2, \dots, S_n) = \sum_{j=1}^n u_j / A_j \sum_{i \in S_j} f_i$$

where

$$f_i = \sum_{k=1}^p v_k * d_k \text{ for } i = 1, 2, \dots, m.$$

An equivalent condition for the matrix  $W$  to factor is;

$$w_{jk} = c_j * w_k \text{ for } j = 1, 2, \dots, n \text{ and } k = 1, 2, \dots, p$$

where

$$c_j = \sum_{k=1}^p w_{jk} \text{ for } j = 1, 2, \dots, n$$

and

$$w_k = \sum_{j=1}^n w_{jk} / \sum_{k=1}^p \sum_{j=1}^n w_{jk} \text{ for } k = 1, 2, \dots, p.$$

This condition is equivalent to the original factoring condition.

If  $u_j = c_j$  and  $v_k = w_k$  and the matrix  $W$  factors,

$$\sum_{k=1}^p w_{jk} = u_j \sum_{k=1}^p v_k$$

$$\sum_{j=1}^n w_{jk} = v_k \sum_{j=1}^n u_j$$

$$\sum_{k=1}^p \sum_{j=1}^n w_{jk} = \left( \sum_{j=1}^n u_j \right) \left( \sum_{k=1}^p v_k \right).$$

Solving for  $u_j$  and  $v_k$  and using the above expression,

$$u_j v_k = \left( \sum_{k=1}^p w_{jk} \right) \left( \sum_{j=1}^n w_{jk} \right) / \left( \sum_{k=1}^p \sum_{j=1}^n w_{jk} \right) = c_j w_k.$$

This is an important intuitive result that states that the factoring assumption is satisfied whenever the total number of pallets of item  $j$  traveling in and out from dock  $k$  per time period can be obtained by multiplying the total number of pallets of item  $j$  (traveling in and out of storage per time period) by the percentage of all pallets traveling in and out of storage from dock  $k$ . That is, all individual items have the same distribution of dock usage.

### 2.7.3 Problem Solution Using Factoring Assumption

To motivate the procedure to be developed for finding a least cost layout, consider the following two sequence of numbers: (1,3,4,6,8) and (5,6,2,9,7). Consider rearranging the second sequence of numbers so that the vector multiplication will maximize the scalar product, (1,3,4,6,8) and (2,5,6,7,9). Since the largest numbers are multiplied together, the second largest numbers are multiplied together, and so on, the maximum scalar product will be obtained which is;

$$8(9) + 6(7) + 4(6) + 3(5) + 1(2) = 155.$$

The following arrangement will minimize the scalar product; (1,3,4,6,8) and (9,7,6,5,2). Thus;

$$1(9) + 3(7) + 4(6) + 6(5) + 8(2) = 100,$$

is the minimum scalar product.

In a warehouse layout formulation, the first of the sequences represents the average distance between a grid square and docks. Let the second sequence represent the numbers

$$c_1/A_1, c_2/A_2, \dots, c_n/A_n.$$

Recalling that  $u_j$  can be replaced by  $c_j$  and  $v_k$  by  $w_k$ , the items will be numbered so that;

$$\frac{c_1}{A_1} > \frac{c_2}{A_2} > \dots > \frac{c_n}{A_n}.$$

The problem of interest is to find the layout  $(S_1^*, S_2^*, \dots, S_n^*)$  such that  $F(S_1^*, S_2^*, \dots, S_n^*) < F(S_1, S_2, \dots, S_n)$  for all  $((S_1, S_2, \dots, S_n))$ . The procedure for finding the least cost layout is as follows:

For  $i = 1, 2, \dots, m$  compute,

$$f_i = \sum_{k=1}^p w_k d_{ki}$$

and let  $i_1, i_2, \dots, i_m$  be a permutation of  $1, 2, \dots, m$  such that

$f_{i_1} < f_{i_2} < \dots < f_{i_m}$ . A least cost layout is then given, on defining

$B_j = \sum_{h=1}^j A_{i_h}$  for  $j = 1, 2, \dots, n$  by;

$$S_1^* = (i_1, \dots, i_{B_1})$$

$$S_2^* = (i_{B_1+1}, \dots, i_{B_2})$$

$$S_n^* = (i_{B_{n-1}+1}, \dots, i_{B_n}).$$

Consider the case of two different items. Item number 1 takes up a large number of grid squares and has a small total number of pallets traveling in and out of storage per time period. Now suppose that item number 2 takes up a smaller number of grid squares and has a large number of pallets traveling in and out of storage per unit time. In that case, item number 2 should be located so that its average distance from the input/output point is less than the average distance from the input/output point of item number 1 to minimize the order picking costs.

## 2.8 Quantity and Location Problems Considered Simultaneously

In the previous sections, only the stock location problem was considered to achieve a minimum order picking cost. The problem is now expanded to include the quantity and location problems simultaneously to achieve a minimum total cost, which includes order picking costs and inventory costs.

Wilson [2] consider the problem of assigning warehouse space to inventory items. Traditionally, the amount of space allocated to each item has been determined on the basis of inventory or production cost considerations. Assuming that the space requirements are available, the actual assignment of items to storage locations within the warehouse is carried out independently. Wilson demonstrates that the quantity and location problems must be considered simultaneously to achieve a minimum total cost (order picking cost and inventory cost). His analysis assumes an out-and-back order picking method and the simple economic order quantity model. Wilson develops and applies an iterative solution procedure, based on a gradient search technique.

### 2.8.1 Determination of Reorder Quantity and Stock Location Independently

The problem addressed here is the assignment of warehouse space to inventory items with an objective of generally minimizing the cost of picking orders and transporting them to the input/output point. It is obvious that the optimal arrangement of stock will depend on the order picking method, turnover rates, product space requirements, and often the demand relationships among products. In most cases, the most desirable space is at a premium and tradeoffs must be made. The results for a simple out-and-back order picking method are presented.

The cube-per-order index is one criterion for assigning inventory items to locations within the warehouse. It has been shown earlier that the COI rule produces an optimal solution to the linear programming formulation of the stock location problem.

Harmatuck [9] observed that the order frequency  $OP_j$  does not appear in the COI formula, therefore, optimal location is independent of the item popularity. This statement is valid only if the length of the inventory period  $DD_j$  remains constant as the demand rate changes. Reorder quantity must change in direct proportion to demand. If reorder quantities are calculated according to the simple EOQ formula (which is not a linear function of demand) it is clear that optimal locations will change with demand rates.

### 2.8.2 Interaction Between Inventory Cost and Order Picking Cost

Assuming an out-and-back order picking method and the simple EOQ model, the total cost objective function becomes:

$$Z = \sum_{i=1}^m \sum_{j=1}^n (CM_i/AOS_j * DD_j) * X_{ij} + (Cp * D_j/Q_j) + (Ch * v_j * (Q_j/2))$$

where

$C_p$  = cost of reordering any item (dollars/order),

$C_h$  = inventory carrying cost (dollars/time/\$value),

$D_j$  = demand rate of item  $j$  (units/time),

$Q_j$  = reorder quantity for item  $j$  (units),

and

$V_j$  = value of item  $j$  (dollars/unit).

A few helpful relationships are given below.

$$D_j = OP_j * AOS_j$$

$$DD_j = Q_j/D_j$$

$$Q_j = \sum_{i=1}^m X_{ij}.$$

The last relationship rests on the simplistic assumption of the EOQ model that the maximum quantity on hand of any product is its reorder quantity.

The total cost function can now be written as,

$$Z = \sum_{j=1}^n OP_j \left( \frac{\sum_{i=1}^m CM_i * X_{ij}}{\sum_{i=1}^m X_{ij}} \right) + \sum_{j=1}^n \left\{ C_p * D_j / \left( \sum_{i=1}^m X_{ij} \right) + (C_h * (V_j/2)) \sum_{i=1}^m X_{ij} \right\}.$$

This function is to be minimized subject to the following constraints:

$$\sum_{j=1}^n VU_j * X_{ij} < CAP_i.$$

The above states that the volume capacity of location  $i$  is not exceeded for all storage addresses  $i = 1, 2, \dots, m$ . The second constraint is:

$$X_{ij} > 0 \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n.$$

This requires that the number of units of item  $j$  assigned to location

$i$ , for each  $i$  and  $j$ , is non negative. Note that the constraint

$$\sum_{i=1}^m X_{ij} = AOS_j * OP_j * DD_j \quad j = 1, 2, \dots, n$$

which states that shortages are not allowed, has been incorporated into the objective function by substituting for  $DD_j$ . Note also that  $DD_j$ , which was constant in the original LP problem, is a variable in the expanded problem.

Inspection of the new objective function reveals the interaction between inventory costs and order picking costs. Since the reorder quantity  $\sum_{i=1}^m X_{ij}$  appears in both terms, it is evident that the application of EOQ will not produce a minimum cost solution.

Intuitively, one expects the optimal reorder quantities to be somewhat less than the EOQs, since reducing space requirements allows all items to be nearer the shipping area, thereby reducing order picking costs.

### 2.8.3 Solution Technique

The objective function along with its associated constraints represents a nontrivial nonlinear programming problem appearing to defy closed form solution. The alternative is a directed search procedure.

The partial derivative of the total cost objective function with respect to  $X_{ij}$  is,

$$\begin{aligned} \partial Z / \partial X_{ij} = & OP_j (CM_i * Q_j - \sum_{i=1}^m CM_i * X_{ij}) / Q_j^2 \\ & + (Ch * V_j / 2) - (Cp * D_j / Q_j^2). \end{aligned}$$

This gives the rate of change of the unconstrained cost function with respect to  $X_{ij}$ ; it disregards the capacity constraints. Given any solution - represented by a set of  $X_{ij}$  values - this derivative gives

the change in cost incurred by changing  $Q_j$ , if the increment in  $Q_j$  could somehow be accommodated in location  $i$ . If the change in  $Q_j$  is positive, the change in actual cost will be algebraically greater. However, since the required space is generally not available in location  $i$ , some product relocation must take place.

Consider a set of positive reorder quantities obtained in the  $k$ th interation;  $Q_1^k, Q_2^k, \dots, Q_n^k$ , which have been allocated by the COI method in quantities

$$x_{ij}^k, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n.$$

Let's say that the item  $p$  has the highest COI of all the items allocated to  $i$  (item  $p$  was the last item allocated to location  $i$ ). The increased picking costs due to moving a unit of item  $p$  from location  $i$  to the location  $i+1$  (assuming that space is available in location  $i+1$ ) is,

$$\begin{aligned} \Delta i' &= (OP_p / Q_p^k)(CM_{i+1})(X_{i+1,p} + 1) + CM_i(X_{i,p} - 1) \\ &\quad - (OP_p / Q_p^k)(CM_{i+1} * X_{i+1,p} + CM_i * X_{i,p}) \\ &= (OP_p / Q_p^k) * (CM_{i+1} - CM_i). \end{aligned}$$

The increased cost per unit volume of item  $p$  which is relocated is,

$$\Delta i = \Delta i' / VU_p = (OP_p / (Q_p^k * VU_p))(CM_{i+1} - CM_i).$$

The increased order picking cost attributed to the relocation resulting from increasing  $Q_p$  by one unit, is the sum of the cost of transferring an equal volume of product between each successive pair of locations more distant from the input/output point than location  $i$ .

When  $Q_j$  is increased by one unit, the increased order picking cost due to product transfer can be written as,

$$VU_j \sum_{i=i_j}^{m-1} \Delta i$$

where  $i_j$  is the greatest  $i$  for which  $X_{ij} > 0$ .

Define  $Z^*(Q_1^k, Q_2^k, \dots, Q_n^k)$  to be the total cost  $Z$  corresponding to a set of reorder quantities  $Q_j^k$ , which have been allocated according to the COI rule. The partial derivative of interest is the sum of the unconstrained partial derivative of the objective function plus the incremental cost due to product relocation,

$$\begin{aligned} \partial Z^*/\partial Q_j &= G_j = VU_j \sum_{i=i_j}^{m-1} \Delta i + OP_j (CM_{i_j} * Q_j \\ &\quad - \sum_{i=1}^m CM_{i_j} * x_{ij}) / (Q_j^k)^2 + Ch * v_j / 2 \\ &\quad - C_p * D_j / (Q_j^k)^2. \end{aligned}$$

This is the  $j$ th element of the gradient vector. Small movements in the direction of the negative gradient will reduce the total cost.

The gradient of  $Z^*$  can be calculated so that a search procedure can be carried out. The COI allocation assures that the capacity constraint is satisfied. In sum, the original constrained optimization of  $X_{ij}$  (the total cost objective function  $Z$ , subject to the appropriate constraints) has been transformed into an unconstrained optimization problem in  $Q_j$ . Since the starting point for each search procedure is the EOQ solution, and the procedure is not likely to approach zero quantities the non-negativity constraint are ignored.

A step size is also needed in order to complete the search

procedure. The beginning step size is defined by the user. This is the maximum change to be made in any  $Q_j$  in one iteration, denoted by  $\Delta Q_{\max}$ . The step size is calculated by,

$$\text{STEP} = \Delta Q_{\max} / \max(|G_j|),$$

from which the new reorder quantities are calculated as,

$$Q_j^{k+1} = Q_j^k - \text{STEP} * G_j, \quad j = 1, 2, \dots, n.$$

The step parameter  $\Delta Q_{\max}$  is left unchanged until an increase in total cost is encountered, then  $\Delta Q_{\max}$  is reduced by a factor of one-half. The iteration continues until the change in reorder quantities is sufficiently small or until some pre-specified number of iterations have been completed. The flow chart shown in Figure 2.8 depicts the logic of the procedure. It is difficult to establish that the local minimum found by this iterative procedure is also global minimum. The solution obtained is the local minimum in the vicinity of the EOQ solution. It appears unlikely that any other local minimum would represent a substantial improvement in terms of cost savings. In any event, the algorithm always gives a lower cost solution than the elementary use of the EOQ and COI rule independently.

The algorithm described here appears to present a practical solution technique for the joint reorder quantity - stock location problem.

The underlying model assumption may not fit a particular real setting in two respects:

- 1) The order picking method may not be of the out-and-back type,

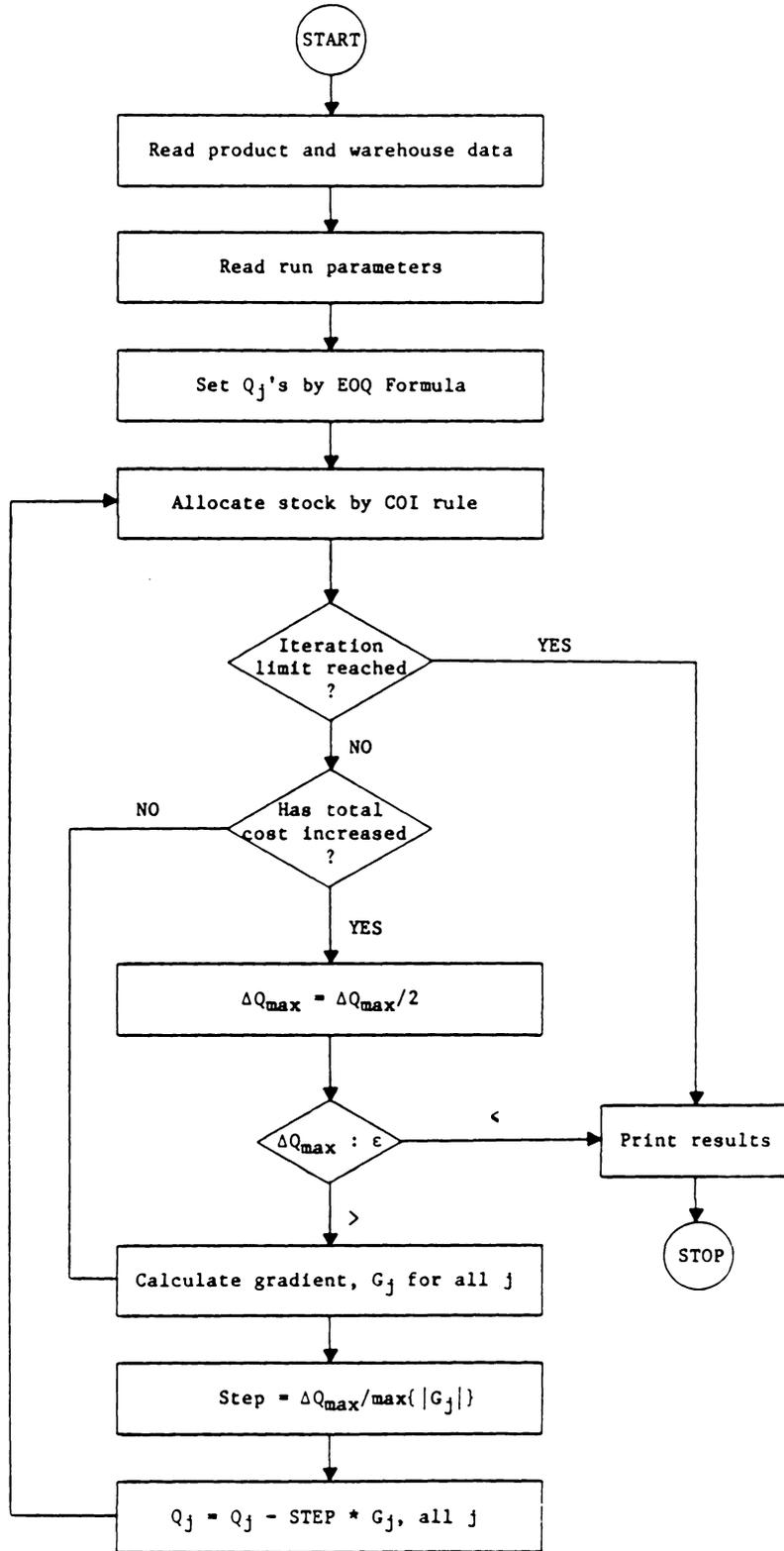


Figure 2.8. Wilson's Flow Chart.

- 2) inventory costs may be more complex than those assumed in the EOQ framework.

The out-and-back order method need not be strictly adhered to. The COI rule is appropriate for any facility wherein the cost of retrieving a quantity of a product from storage is determined primarily by the location of the product and is independent of particular product that is being retrieved and the products that may happen to be contained in the same order.

The primary requirement of the above model is that the inventory cost must be differentiable with respect to the maximum quantity on hand.

## 2.9 Summary

The literature review reveals that research applicable to the warehouse stock location model has been extensive in the last 20 years, starting with Heskett's cube-per-order index rule. Models have been developed to analyze various aspects of the stock location problem utilizing the cube-per-order index rule, linear programming and computer simulations. Many different models are in use today, yet researchers are still striving to formulate a model that will provide more accurate answers to the real life warehouse stock location problem.

The location process involves several dependent considerations. The order picking method depends on warehouse design, item bulk, and distribution of other items. The cost of transporting an item is dependent upon the quantity of all items stored in the warehouse. A

complete location analysis should include an assessment of all the above considerations providing a total minimum cost (order picking cost plus inventory costs).

Stock location models have tended to focus on one consideration or another by making simplifying assumptions about the other dependent facets of the model. One of the major simplifications has been the assumption that the item space requirements, derived from the simple economic order quantity inventory model, are given. Wilson [2] states that the quantity and location problems must be considered simultaneously in order to achieve a minimum total cost. Another major assumption made frequently is the use of an out-and-back order picking method. This ignores the potential higher throughput obtained by using an interleaving order picking method.

A new warehouse stock location model that will incorporate an interleaving order picking method, while considering the quantity and location problems simultaneously, is desired. Just as previously developed stock models have done, the new model will generate individual item locations and their reorder quantities while continuing to meet demand requirements at all times.

## CHAPTER III

### THE MODELING APPROACH

#### 3.1 Introduction

In this chapter the Interleaving Warehouse Layout Model is developed. The assumptions made to formulate the model are summarized and discussed. This chapter deals also with how these assumptions may be relaxed to encompass a wide variety of warehouse layout problems. The chapter concludes with a summary of the modeling approach.

#### 3.2 The Model Assumptions

The warehouse layout is assumed to be rectangular in shape, where the number of columns (x-axis) is one greater than the number of rows (y-axis), or a square shape where the number of columns equals the number of rows. The area of the warehouse is assumed to be known. The warehouse is to have only one input/output point (dock), located in the lower left corner of the warehouse. A coordinate system is established so that origin (0,0), represents the input/output point. An orthogonal network of aisles running parallel to the x and y axes allows the approximation of the rectilinear travel distance between subareas (bays) and the input/output point. All aisles are of the same size throughout the warehouse and all travel is by the shortest route possible. The backtracking is not allowed.

Due to the above assumptions the cost of travel of the order picking vehicle is proportional to the rectilinear travel distance. Using the interleaving order picking method, cost of travel for all

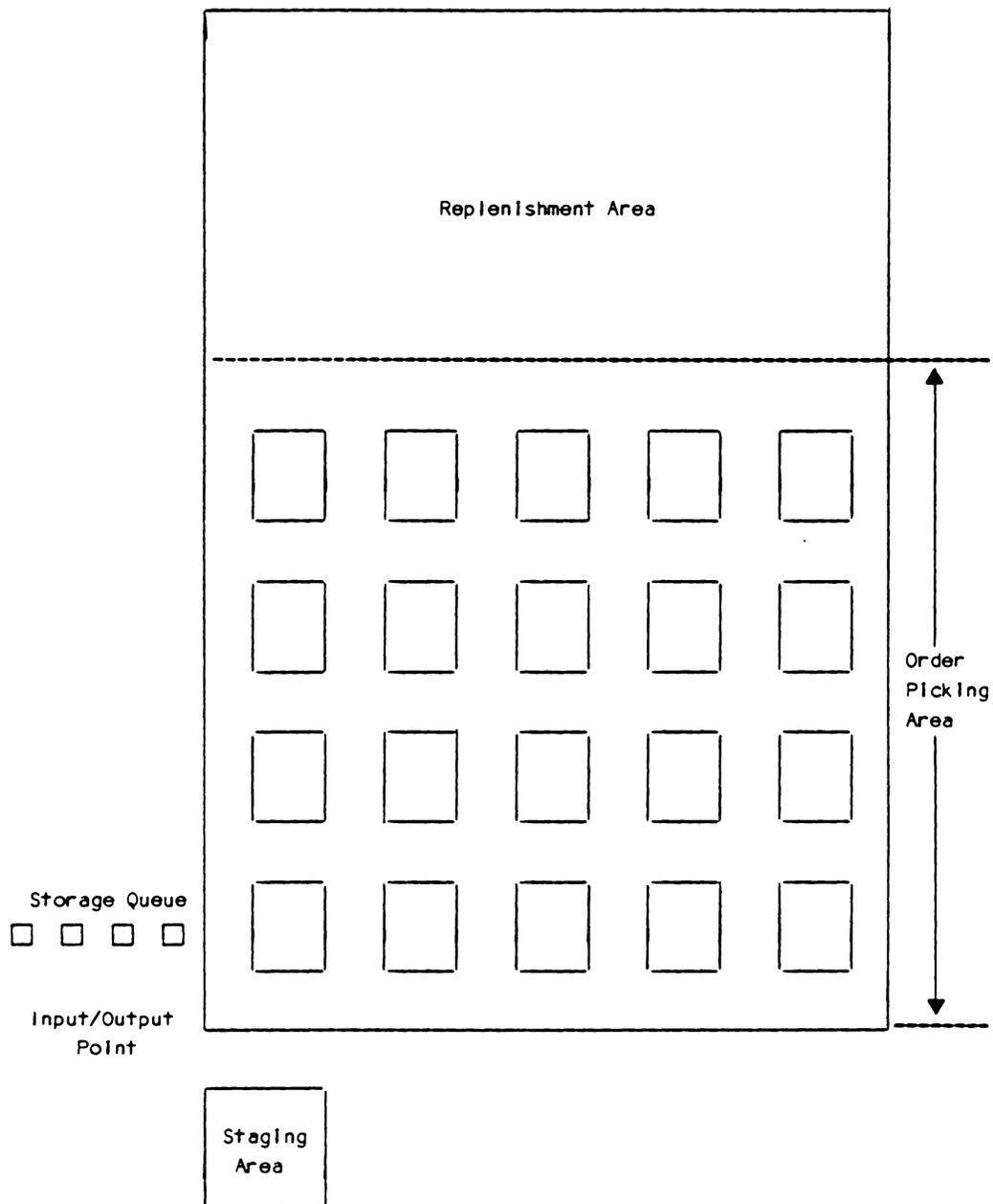


Figure 3.1. Warehouse Layout.

interleave combinations must be generated.

The system is bound at the point where the order picking vehicle transfers pallets to the input/output conveyor. Incoming and outgoing pallets are transferred at the same point (the input/output point), which has a fixed and known location at one corner of the warehouse.

Individual subareas (bays) will be given two types of address classifications. First a Unique Address Classification (U.A.C., Figure 3.2), which is a unique address for each subarea in the form of a column, row configuration. Secondly, Preference Classification (P.C., Figure 3.3), which is based on rectilinear travel distance from the input/output point to the subarea in question. The most preferred space is numbered 1, the next most preferred is numbered 2 and so on. It is realized that there will be ties, and all ties are given the same classification.

The Preference Classification for each space may be calculated by the relationship  $PC = c+r-1$ , where  $c$  and  $r$  are the column and row numbers respectively of the Unique Address Classification. Therefore, the distance from location  $c, r$  (U.A.C.) is equivalent to the distance to Preference Class  $c+r-1$ . This Preference Classification relationship will be used in the Modified Location ABC analysis.

The dedicated cube-per-order (COI) index storage assignment rule will be used. Dedicated storage assigns each item to a fixed location. In order to maximize throughput while using dedicated storage, items will be assigned to storage locations based on their COI value. The COI value of an item is simply the ratio of space to be allocated for

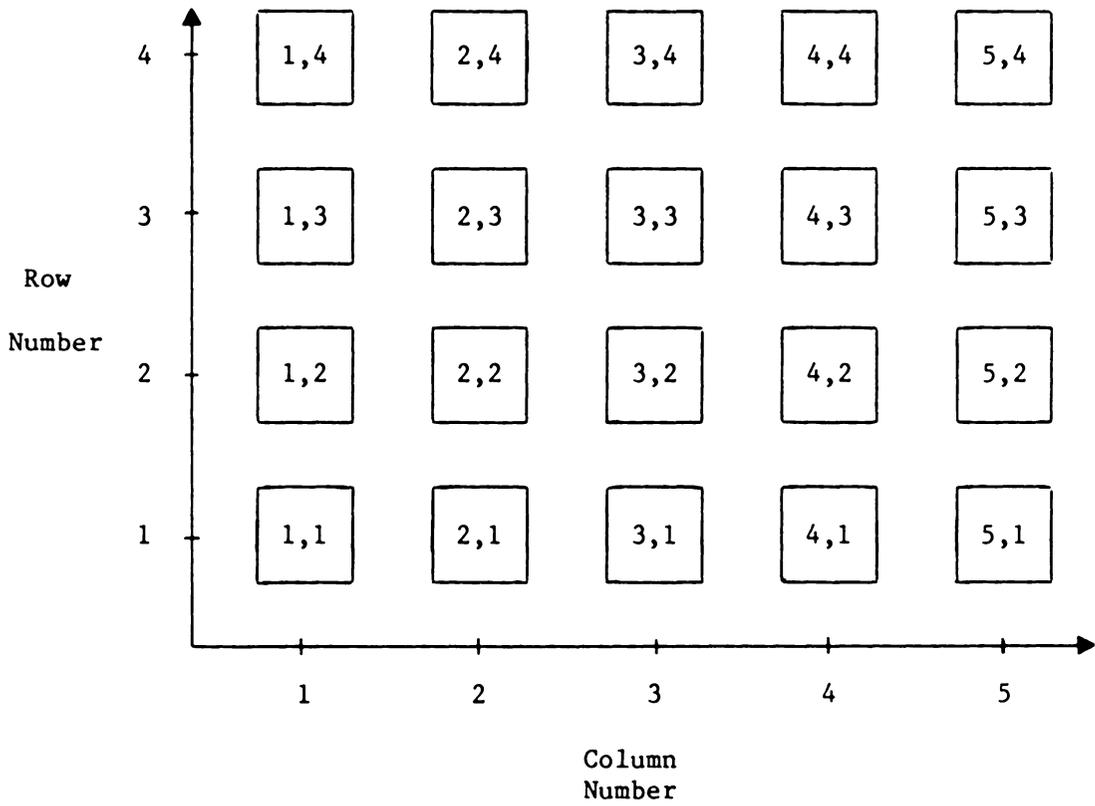


Figure 3.2. Unique Address Classification.

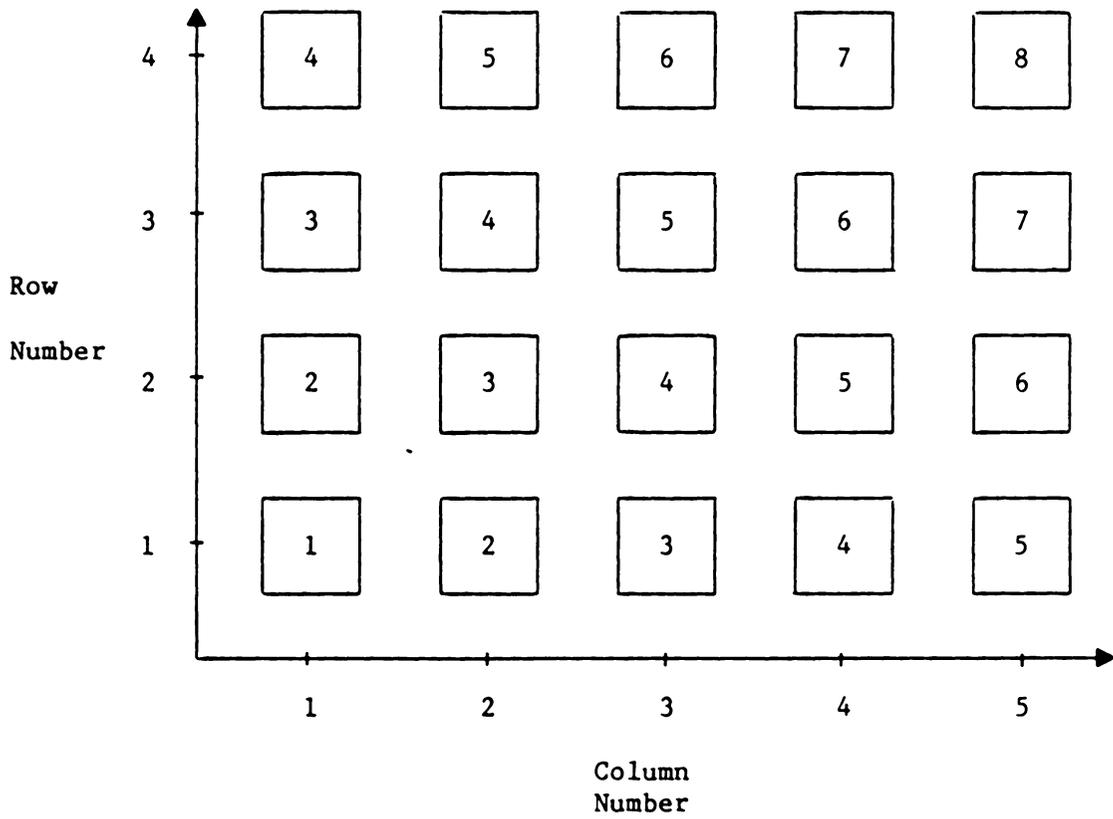


Figure 3.3. Preference Classification.

that item, to the number of orders for that item.

The item having the lowest COI value is assigned to the most preferred opening, the next lowest COI value to the next most preferred opening, and so on. Each location is filled to capacity. When the capacity of a particular space is reached, the next most preferred space is used. This procedure is carried out until all items are allocated. Since "compact fast movers" are up front and "bulky slow movers" are in the back, throughput is maximized for a given set of order quantities.

All storage locations in the warehouse are of the same size, as are the pallets themselves. Also, there are no compatibility constraints prohibiting close proximity of item pairs. Therefore, all storage locations are candidates for storing a pallet load of any item. The assignment of multiple items to the same pallet is valid. For example, more than one item may be found in a particular location and one item may be split among several locations.

It is assumed that a mandatory interleaving (MIL) order picking method is used. The order picking vehicle visits two locations in the warehouse, a storage location and a retrieval location (which may be the same), between successive returns to the input/output point. If the storage location and the retrieval location are the same, the interleave travel cost will be zero and the cost of travel to the storage location will equal the cost of travel to the retrieval location. The statistical independence between and among storages and retrievals is also assumed. A typical round trip consists of a storage

request (a pallet load) followed by a retrieval. The order picking vehicle operator will retrieve a pallet and transport it to the staging area where a picker will pick the desired item. The pallet is then placed in the storage queue as shown in Figure 3.1.

The warehouse is divided into two distinct and separate areas. The order picking area, which is the area being studied, and the replenishment area, which is separate and is not included in the analysis. The replenishment area's sole purpose is to restock the order picking area during an off-shift.

The cost of moving an order of any item from a particular location to the input/output point is constant, depending only on the location. Actual time to load or unload a pallet (transfer time) at the input/output point, storage location, or retrieval location, is ignored. The number of order picking transactions per period of each item is known and constant.

Finally, short-run dynamic considerations are ignored and the long run average behavior of the system (steady state) is investigated.

### 3.3 Profile of Related Problems

The assumptions stated in the previous section restrict the warehouse layout problem addressed so that a model could be solved with a minimum of effort. This section describes the assumptions that can be easily manipulated or relaxed to encompass a wider variety of problems.

If the actual warehouse capacity is not large enough to handle the EOQ reorder quantities, dummy locations may be added to increase this

capacity. Obviously, the actual warehouse capacities must be large enough to accommodate the final reduced reorder quantities.

The algorithm works well for a distribution warehouse, using a dedicated COI assignment rule, where item assignment is based on the item popularity and the item size. The algorithm will not work well for a manufacturing warehouse, where the dedicated COI assignment rule is discarded, and item assignment is based on group component assembly (where items are stored on the basis of assembly order).

If mandatory interleaving (MIL) rule is not strictly enforced, actual travel times will be weighted average of non interleaving (NIL) and MIL performances. This allows for a particular queue, of storage or retrieval, to become empty; but both queues should not be empty simultaneously.

The warehouse layout need not be restricted to a square or the one additional column rectangular shape discussed earlier. The Preference Classification relationship and the rectilinear travel distance relationship remain true for all possible rectangular shaped layouts. Only the item allocation routine must be adapted to match each individual layout.

### 3.4 Model Formulation

The last chapter described how an ABC Curve is plotted to provide the cumulative fraction of the total demand as a function of the cumulative fraction of the total items. In this section, a Modified Location ABC Curve is developed. This curve gives the cumulative fraction of the total demand for cumulative fraction of the total

number of Preference Classes used. The Modified Location ABC Curve permits approximation of the probability of a particular space classification being accessed.

The following notations and relationships are used in the development of the model in subsequent sections.

The  $n$  items stored in  $C \times R$  locations, where  $C$  and  $R$  are the column and row number respectively, are characterized by following:

$DD_j$  = The time period for which the demand for item  $j$  should be satisfied (time).

$VU_j$  = The volume of space required per unit of item  $j$  (length<sup>3</sup>/unit).

$AOS_j$  = The average order size of item  $j$  (units/order).

$OP_j$  = The number of orders per period for item  $j$  (orders/time).

$X_{crj}$  = The amount of item  $j$  in location  $(c,r)$  (units).

$D_j$  = Demand rate of item  $j$  (units/time).

$$D_j = OP_j * AOS_j$$

$$DD_j = Q_j / D_j$$

$$Q_j = \sum_{c=1}^C \sum_{r=1}^R X_{crj}$$

The last relation specifies that the maximum quantity on hand of any product is its reorder quantity and it is based on the simplistic assumptions of the EOQ model.

#### 3.4.1 The Modified Location ABC Curve

In the model developed the demand as measured by the number of orders per period for each item is assumed to be known. Once the items are allocated to various locations within the warehouse, each location

satisfies a certain fraction of the total demand for the inventory cycle period. This fraction of demand satisfied by a particular location determines the frequency of visits to that location. To obtain the total cost of interleave and retrieval, a measure of operational probabilities for accessing any location within the warehouse is needed. These operational probabilities are determined by the fraction of the total demand stored within various locations.

When a pallet load is retrieved and moved to the input/output point, the order for the item requested is removed and the pallet awaits storage till the next demand is made. Since dedicated storage assignment is employed, visit to a storage location depends only on the allocation of the pallet to the particular location from which it was obtained. Visit to a retrieval location depends only on the demand distribution of that location (percent of the total demand stored within the location). As successive demands are assumed to be independent, the storages and retrievals are also statistically independent. The probability of visiting a retrieval location  $(z,y)$  will not depend on the storage location visited before but will depend only on the demand distribution of the retrieval location.

The percentage of total demand met by a particular location is obtained from allocation of items within that location, thus a different allocation of items will change the percentage demand (and hence the operational probability) for that location. The Modified Location ABC Curve described next gives the cumulative percentage of total demand for preference classifications. This curve is utilized to compute the operational probabilities of each location in the

warehouse.

The following procedure describes how the Modified Location ABC Curve is obtained.

1) Using the Economic Order quantity analysis, quantity  $Q_j = EOQ_j$  is computed for each of the  $n$  items.

2) The cube-per-order index  $COI_j$  for each item  $j$  is calculated from the expression

$$\begin{aligned}
 COI_j &= VU_j * AOS_j * OP_j * DD_j / OP_j \\
 &= VU_j * AOS_j * DD_j \\
 &= VU_j * AOS_j * Q_j / D_j \\
 &= VU_j * AOS_j * Q_j / (OP_j * AOS_j) \\
 &= VU_j * Q_j / OP_j.
 \end{aligned}$$

All items are ranked in the order of their COI values such that

$$COI_1 < COI_2 < \dots < COI_i < COI_{i+1} < \dots < COI_n.$$

3) Space requirement  $VU_j * Q_j$  for each item is calculated. Spaces in the warehouse are numbered using the Preference Classification. Since all locations have the same volume of space, this volume is constant for all locations. Hence volume for each Preference Classification is determined by multiplying the number of locations in each Preference Classification by the above constant.

4) The items are then allocated to locations, where the lowest COI value occupies the most preferred space (closest distance to the input/output point), the next lowest COI value to the next most preferred space, and so on. Item allocation in the case where ties

occur because of Preference Classifications of spaces being equal (distances from the input/output point for several locations are equal) is done arbitrarily.

5) Starting with the lowest numbered Preference Classification, the cumulative percentage of demand stored in each Preference Classification is computed. The Modified Location ABC Curve can be represented by the function

$$\text{MLC}(i) = i^s \quad \text{for } 0 < s < 1$$

where  $s$  is the fit parameter for the Modified Location ABC Curve. The  $s$  corresponding to a Preference Classification value  $k$  may be found by solving

$$\begin{aligned} i^s &= \text{OP}_k / \sum_j \text{OP}_j \\ &= (\text{VU}_k * Q_k / \sum_j \text{VU}_j * Q_j)^s, \end{aligned}$$

which gives:

$$\text{OP}_k / \sum_j \text{OP}_j = (\text{VU}_k * Q_k / \sum_j \text{VU}_j * Q_j)^s \quad (1)$$

where

$\text{OP}_k$  = Cumulative demand stored in Preference Classification up to and including Classification  $k$ ,

$\text{VU}_k * Q_k$  = Cumulative volume of space of all items stored in Preference Classifications up to and including Preference Classification  $k$ .

$\sum_j \text{OP}_j$  and  $\sum_j \text{VU}_j * Q_j$  represent the total demand and total volume of space of all items  $j = 1, \dots, n$  stored in all Preference Classifications.

Following equation is obtained by taking logarithms of both sides of equation (1).

$$\ln(OP_k / \sum_j OP_j) = s * \ln(VU_k * Q_k / \sum_j VU_j * Q_j)$$

which gives

$$s = \ln(OP_k / \sum_j OP_j) / \ln(VU_k * Q_k / \sum_j VU_j * Q_j)$$

For each individual point (corresponding to each Preference Classification value) on the Modified Location Curve, an individual  $s$  parameter is obtained. The representative value  $s^*$  for the entire curve is obtained by averaging the nontrivial  $s$  values for the Preference Classifications.

The probability of Preference Classification  $Z$  being accessed can now be represented by:

$$P(Z) = i' s^* - i'' s^*$$

where  $i' = Z / (\text{total number of preference classes used})$  and  $i'' = (Z-1) / (\text{total number of preference classes used})$ .

The Preference Classification number for Unique Address  $(z,y)$  is  $z + y - 1$ . If  $NUM_{z,y}$  is the number of Unique Address Classifications in the Preference Classification number  $z + y - 1$ , then  $P(z,y)$ , the probability of Unique Address Classification  $(z,y)$  being accessed can be represented by

$$P(z,y) = (i' s^* - i'' s^*) / NUM_{z,y} \quad (2)$$

The term  $P(z,y)$  represents the probability of visiting the retrieval location  $(z,y)$ . This probability depends upon the external demands for the items when the warehouse is in actual operation. However, when the mathematical model is formulated to solve the

quantity and location problems simultaneously for the warehouse, the distribution of external demand for items is not known. Therefore, the probability of visiting the retrieval location  $(z,y)$  is computed utilizing the distribution of items to different locations within the warehouse as per the layout in the current iteration.

### 3.4.2 Formulation of Travel Distances

Each subarea of the warehouse has length of each side equal to  $LS$ . The orthogonal network of aisles running parallel to the  $x$  and  $y$  axes allows the calculation of the rectilinear distance traveled between a subarea and the input/output point or the distance traveled between one subarea and another. The aisle width  $AW$  is assumed to be constant.

#### 3.4.2.1 Travel to the Storage Location

The rectilinear distance that any item stored in the warehouse travels between its location at point  $(c,r)$  and the input/output point  $(0,0)$  is given by

$$\begin{aligned} & \{(c-1) + (r-1) + 1\} * LS + (c+r) * AW \\ & = (c+r-1) * LS + (c+r) * AW. \end{aligned} \quad (3)$$

#### 3.4.2.2 Interleave Travel Distance

If  $(c,r)$  is the storage location and  $(z,y)$  is the retrieval location, the rectilinear distance traveled while performing the interleave may be represented by

$$(|c-z| + |r-y|) * LS + (|c-z| + |r-y|) * AW. \quad (4)$$

Since the cost of moving any item to any location is proportional to the rectilinear distance traveled, multiplying the rectilinear distance traveled by the cost of moving an order through unit distance,  $DOL$ ,

will give the cost of a particular move.

### 3.4.2.3 Retrieval Travel Distance

The distance traveled in moving an order of an item from a retrieval location to the input/output point is same as the distance traveled in moving an order of an item from the input/output point to the same location. Hence the distance traveled in moving an order of an item from retrieval location  $(z,y)$  to the input/output point is given by the following expression

$$(z+y-1) * LS + (z+y) * AW.$$

Multiplying the above expression by DOL, the cost of moving an order of an item through a unit distance gives  $CM_{zy}$ , the cost of moving an order of an item from location  $(z,y)$  to the input/output point. Thus,

$$CM_{zy} = \{(z+y-1) * LS + (z+y) * AW\} * DOL.$$

Similarly, the cost of moving an order of an item from the input/output point to the location  $(c,r)$  is

$$CM_{cr} = \{(c+r-1) * LS + (c+r) * AW\} * DOL.$$

### 3.4.3 Order Picking Cost Expressions

The order picking cost expression has components for following three travels:

(i) travel from the input/output point to the storage location  $(c,r)$ ,

(ii) travel from the storage location  $(c,r)$  to the retrieval location  $(z,y)$ , and

(iii) travel from the retrieval location  $(z,y)$  to the input/output point.

### 3.4.3.1 Storage Travel Costs

Let  $X_{crj}$  and  $OP_j$  be defined as follows:

$X_{crj}$  = number of units of item  $j$  stored in location  $(c,r)$ ,

$OP_j$  = number of orders per period of item  $j$ .

Then,  $X_{crj} / \sum_C \sum_R X_{crj}$ , is the proportion of item  $j$  stored in location  $(c,r)$ . This proportion multiplied by the number of orders per period,  $OP_j$ , gives the number of orders for item  $j$  stored in location  $(c,r)$ .

Let  $CM_{cr}$  be the cost of moving an order of an item from the input/output point to location  $(c,r)$ . Then the travel cost per period of all items stored in location  $(c,r)$  is given by the expression:

$$\sum_{j=1}^n (X_{crj} / \sum_C \sum_R X_{crj}) * OP_j * CM_{cr}.$$

The following total travel cost expression for all storage operations is obtained by summing up the above cost over all storage locations.

$$\sum_C \sum_R [ \sum_{j=1}^n (X_{crj} / \sum_C \sum_R X_{crj}) * OP_j * CM_{cr} ]. \quad (5)$$

### 3.4.3.2 Interleaving Travel Costs

The interleave travel from location  $(c,r)$  to location  $(z,y)$  occurs to perform a retrieval from the location  $(z,y)$ . As shown in equation (4), the interleave travel distance from location  $(c,r)$  to location  $(z,y)$  is given by

$$(|c-z| + |r-y|) * (LS + AW) \quad (6)$$

where

LS = length of the side of a storage subarea within the warehouse.

AW = Aisle width.

The cost per order of interleave travel from storage location  $(c,r)$  to

retrieval location  $(z,y)$  is obtained by multiplying the above travel distance by DOL (the travel cost per unit distance per order) as in expression (7).

$$(|c-z| + |r-y|) * (AW + LS) * DOL. \quad (7)$$

The expected cost per order of interleave travel from storage location  $(c,r)$  to all retrieval locations  $(z,y)$  is obtained by multiplying expression (7) by  $P(z,y)$ , the probability of accessing unique address location  $(z,y)$  and summing up over all locations  $Z, Y$  as in expression (8)

$$\sum_Z \sum_Y (|c-z| + |r-y|) * (AW + LS) * DOL * P(z,y). \quad (8)$$

The total expected interleaving travel cost per period for interleave travel from all storage locations  $(c,r)$  is obtained by multiplying expression (8) by proportion of orders for all items stored in location  $(c,r)$  and summing over all locations  $C, R$ , as in expression (9)

$$\sum_C \sum_R \left[ \left( \frac{\sum_{j=1}^n X_{crj}}{\sum_C \sum_R \sum_{j=1}^n X_{crj}} \right) * OP_j \right] * \left\{ \sum_Z \sum_Y \{ (|c-z| + |r-y|) * (AW + LS) * DOL * P(z,y) \} \right\}. \quad (9)$$

### 3.4.3.3 Retrieval Travel Costs

Each interleave travel to a location  $(z,y)$  from a given location  $(c,r)$  is followed by the retrieval operation performed at the location  $(z,y)$ . The cost per order of retrieval travel to the input/output point from the location  $(z,y)$  is  $CM_{zy}$ . Hence, the expected cost per order of retrieval travel from all location  $(z,y)$  to the input/output point following interleave travel to the same locations from location  $(c,r)$  is given by;

$$\sum_Z \sum_Y CM_{zy} * P(z,y).$$

The total expected retrieval cost per period following interleave travel from all location (c,r) is obtained by multiplying the above expression by proportion of orders for all items stored in location (c,r) and summing over all storage locations as given by equation (10).

$$\sum_C \sum_R \left[ \sum_{j=1}^n \left( \frac{X_{crj}}{\sum_C \sum_R X_{crj}} \right) * OP_j * \left\{ \sum_Z \sum_Y CM_{z,y} * P(z,y) \right\} \right] \quad (10)$$

#### 3.4.4 Inventory Costs

From the simple Economic Order Quantity model, the total inventory cost for all n items is

$$\sum_{j=1}^n \left[ Cp * OP_j * AOS_j / \left( \sum_C \sum_R X_{crj} \right) + Ch * V_j * \left( \sum_C \sum_R X_{crj} / 2 \right) \right]. \quad (11)$$

The first component is the ordering cost given by the product of cost of reorder  $Cp$  times the demand  $D_j = AOS_j * OP_j$  divided by the reorder quantity  $\sum_C \sum_R X_{crj}$ . The second component is the holding cost obtained by the product of holding cost per unit per dollar  $Ch$  and the dollar value  $V_j$  of the item  $j$  times the average inventory level

$$\sum_C \sum_R X_{crj} / 2.$$

#### 3.4.5 The Total Cost Equation

Adding the order picking costs given in equations (5), (9), and (10) and inventory costs given in equation (11), the total cost expression becomes:

$$\begin{aligned}
& \sum_C \sum_R \left[ \left( \sum_{j=1}^n (X_{crj} / \sum_C \sum_R X_{crj}) * OP_j \right) * CM_{cr} \right] \\
& + \sum_C \sum_R \left[ \left( \sum_{j=1}^n (X_{crj} / \sum_C \sum_R X_{crj}) * OP_j \right) \right. \\
& \quad \left. * \sum_Z \sum_Y \{ (|c-z| + |r-y|) * (AW + LS) * DOL * P(z,y) \} \right] \\
& + \sum_C \sum_R \left[ \left( \sum_{j=1}^n (X_{crj} / \sum_C \sum_R X_{crj}) * OP_j * \left\{ \sum_Z \sum_Y CM_{zy} * P(z,y) \right\} \right) \right] \\
& + \sum_{j=1}^n \left[ Cp * OP_j * AOS_j / \left( \sum_C \sum_R X_{crj} \right) + Ch * V_j * \sum_C \sum_R X_{crj} / 2 \right] \quad (12)
\end{aligned}$$

The above total cost equation includes the interaction between inventory costs and order picking cost. The total cost expression also suggests that the optimal reorder quantities ought to be lower than those obtained by the simple Economic Order Quantity considerations based on the inventory costs alone.

#### 3.4.6 Constraints

The function represented by the total cost expression (12) is to be minimized subject to the following set of constraints:

$$\sum_{j=1}^n VU_j * X_{crj} < CAP_{cr}.$$

The above states that the volume capacity of location (c,r) is not exceeded for all (c,r) in the warehouse. The second set of constraints is

$$X_{crj} > 0 \quad j = 1, 2, \dots, n.$$

This requires that the number of units of item j, assigned to location (c,r), for all locations and all items in the warehouse, is nonnegative. The constraints

$$\sum_C \sum_R X_{crj} = AOS_j * OP_j * DD_j \quad j = 1, 2, \dots, n$$

that shortages are not allowed, have been incorporated into the

objective function, and hence can be omitted.

### 3.5 Summary and Conclusions

The foundation of the interleaving warehouse layout model has been established in this chapter. The modeling assumptions encompass as many real life warehouse layout problems as possible. It must be realized that each warehouse layout problem is unique in some way.

Some of the modeling assumptions simply cannot be altered, dedicated storage being one of them. It is realized that the use of dedicated storage takes more space, but knowing where a particular item is stored can be of great advantage in many cases. By knowing a particular item location and item demand characteristics, we are able to calculate the probability of a Unique Address Classification being accessed. Statistical independence between storages and retrievals allows us to calculate the probability of all possible interleaves. The travel cost being proportional to rectilinear travel distance is a practical constraint that allows us to calculate very good estimate of travel costs.

Turnover frequency of each item being known and constant is a simplifying assumption. If it were not to be accepted, a probabilistic calculation of turnover frequency could be implemented.

## CHAPTER IV

### PROPOSED SOLUTION METHODOLOGY

#### 4.1 Introduction

This chapter presents the implementation of the Interleaving Warehouse Layout Model. A description of the computational logic used in the computer program is provided. The computer code GRG2.5 [4] is described briefly. The chapter ends with a brief discussion on how the original solution generated from the model must be adjusted for actual implementation.

#### 4.2 Description of the Computational Logic

The computer code employed to optimize the Interleaving Warehouse Layout consists of four major modules. Figure 4.1 illustrates the general logic for one iteration of the optimization process, showing important steps of the main calling program and the three major subroutines. Figures 4.2, 4.3, and 4.4 give the flowcharts of the logic used in the three major subroutines.

##### 4.2.1 Main Calling Program

The main program computes the warehouse and product information that remains unchanged throughout the optimization process. Thus cost of travel to all locations within the warehouse and volume of space in each Preference Classification are determined in this segment of the computer code. Economic Order Quantities are used as the initial or starting point for the optimization process. Hence, initial values of variables corresponding to the items stored within the warehouse, capacity constraints for each storage location and the total cost

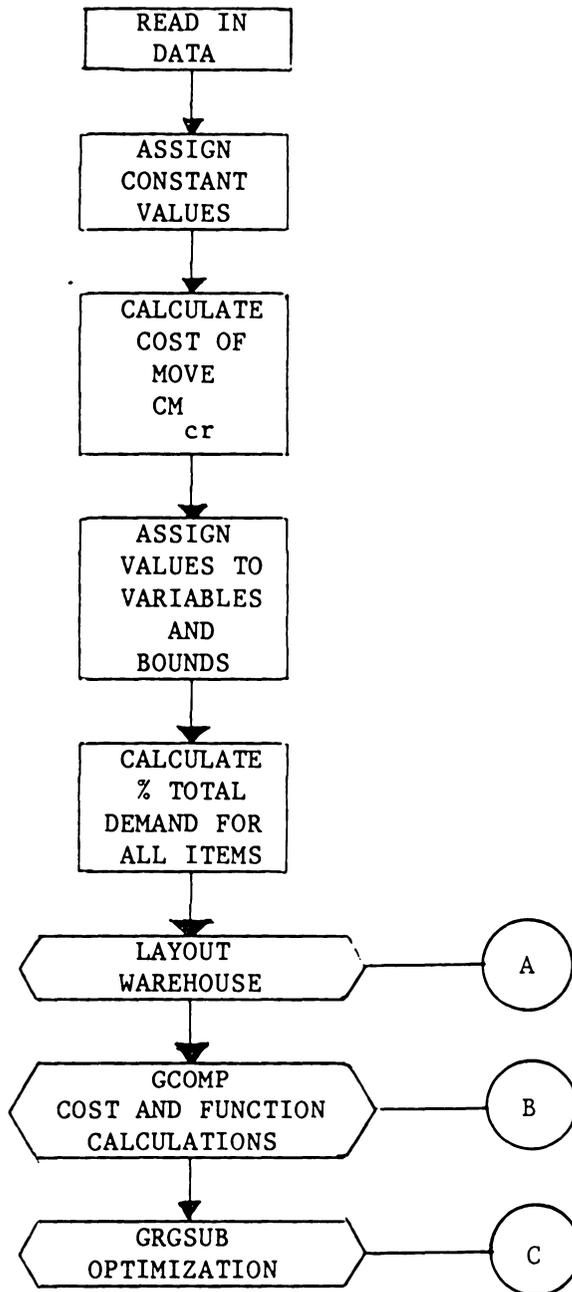


Figure 4.1. Flowchart of General Logic for One Cycle of Optimization of the Interleaving Warehouse Layout Model.

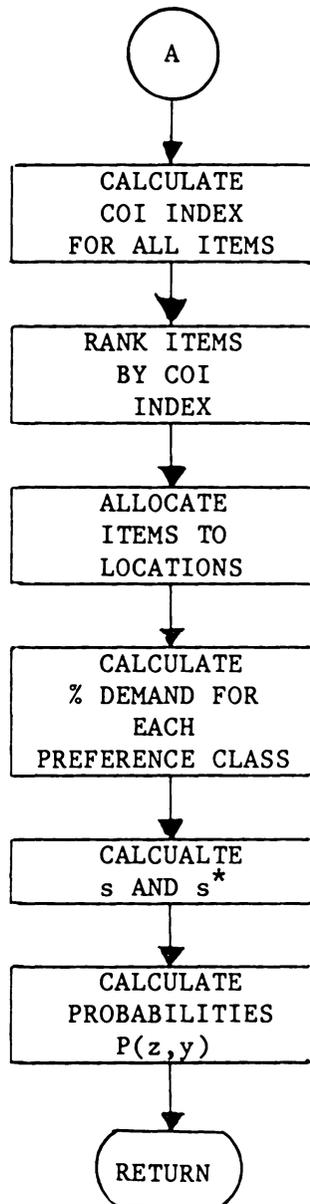


Figure 4.2. Flowchart of General Logic for the LAYOUT Routine.

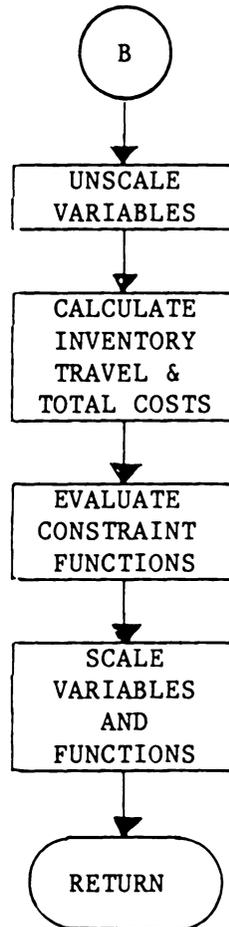


Figure 4.3. Flowchart for General Logic for the GCOMP Routine.

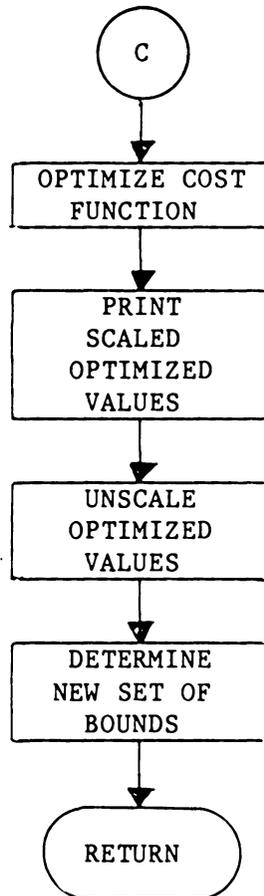


Figure 4.4. Flowchart for General Logic of One Optimization Cycle of GRGSUB Routine.

function are obtained from these Economic Order Quantities. Next, the percentage of total demand for each item, used in the Modified Location ABC Curve calculations, is computed for each item stored.

#### 4.2.2 Layout Subroutine

This subroutine allocates all items to various Unique Address Classifications and then computes the probability of accessing those Unique address Classifications for retrieval of items stored. The COI value for each item is calculated and the items are ranked in the increasing order of their COI values. The subroutine then allocates items to locations such that the item with the lowest COI value occupies the most preferred location, the item with the next lowest COI value occupies the next most preferred location, and so on. Once the allocation procedure is completed, the total cumulative percentage demand for each Preference Classification is calculated. The fit parameter  $s$  corresponding to each Preference Classification value, and the representative value  $s^*$  for the entire Modified Location Curve are computed next. The fit parameter  $s^*$  for the Modified Location Curve is used to compute the probability of accessing different Unique Address Classifications.

The next two major modules of the program are related to the GRG2.5 optimization code, and are described in the following section.

#### 4.3 GRG2.5 Computer Code

GRG2.5 is a Fortran program which solves nonlinear problems of the following form:

$$\begin{aligned}
 &\text{Minimize or maximize } g_k(x) \\
 &\text{subject to } lb(n+i) \leq g_i(x) \leq ub(n+i) \\
 &\qquad\qquad i = 1, 2, \dots, m, \quad i \neq k \\
 &\qquad\qquad lb(i) \leq x_i \leq ub(i) \\
 &\qquad\qquad i = 1, 2, \dots, n.
 \end{aligned}$$

In the above problem  $x$  is a vector of  $n$  real-valued variables. The  $g_i$  are real-valued functions of  $x$  and may be linear or nonlinear. There are  $m$  such functions, one of which is the objective and the others, if present, are equality constraints and/or inequality constraints. Upper and lower bounds on the variables are optional and if present are not treated as additional constraint functions but are handled separately.

The program attempts to solve problems of this form by the Generalized Reduced Gradient Method [15,16,17,18]. A subroutine GCOMP is prepared to compute the values of  $g_1, g_2, \dots, g_m$  for any given vector  $x$ . Data specifying the upper and lower bounds as well as other parameters are also provided.

GRG2.5 uses first partial derivatives of each function  $g_i$  with respect to each  $x_i$  variable. These are automatically computed by finite difference approximation (either forward or central differences) unless a subroutine PARSH, that evaluates them analytically is provided.

In general, after the initial data entry segment, the program operates in two phases. If the initial point supplied does not satisfy all  $g_i$  constraints, a phase I optimization is started. The phase I objective function is the sum of the constraint violations plus,

optimally, a fraction of the true objective. This optimization terminates either with a message that the problem is infeasible or with a feasible solution.

Phase II begins with a feasible solution either obtained by phase I or by a starting point that is feasible, and attempts to optimize the objective function provided.

At the conclusion of Phase II, a complete optimization iteration has been completed and summary output is provided. The optimization process can be restarted, if desired, with the final values obtained or with partial or full modification of the final values obtained.

When GRG2.5 code is used as a stand-alone system, after solving the problem it writes solution data on a printer or on an output data file. The situations like optimization of the Interleaving Warehouse Layout require setting up and solving a sequence of optimization problems, where the solution of each problem is processed to determine what the next problem will be. In such cases, it is convenient to use GRG2.5 as a subroutine GRGSUB which is called from within the user's program. In this mode, all input and output data are communicated to and from GRG2.5 through an argument list of GRGSUB.

Proper scaling of both variables and problem functions is very important for successful operation of GRG2.5. A subroutine SETSC is used to determine variable, constraint and objective function scale factors. These scale factors are determined such that the scaled variable values are from zero to one at the starting point and the scaled objective and constraints values at the starting point are equal

to 1.0.

The variable  $X_{crj}$  used in the main calling program and in the constraints and cost function is a triple subscripted variable, whereas the variable  $XX_i$  used in the subroutine GRGSUB is a single subscripted variable. Hence proper transfers of the values between the arrays of the two types of variables are made to maintain one-to-one correspondence between them all the time.

#### 4.3.1 Subroutine GCOMP

Subroutine GCOMP computes the values of the total cost function and all the constraints. The total cost expression is given by:

$$\begin{aligned} & \sum_C \sum_R \left[ \left( \sum_{j=1}^n (X_{crj} / \sum_C \sum_R X_{crj}) * OP_j \right) * CM_{cr} \right] \\ & + \sum_C \sum_R \left[ \left( \sum_{j=1}^n (X_{crj} / \sum_C \sum_R X_{crj}) * OP_j \right) \right. \\ & \quad \left. * \sum_Z \sum_Y \{ (|c-z| + |r-y|) * (AW + LS) * DOL * P(z,y) \} \right] \\ & + \sum_C \sum_R \left[ \left( \sum_{j=1}^n (X_{crj} / \sum_C \sum_R X_{crj}) * OP_j * \{ \sum_Z \sum_Y CM_{zy} * P(z,y) \} \right) \right] \\ & + \sum_{j=1}^n [ Cp * OP_j * AOS_j / (\sum_C \sum_R X_{crj}) + Ch * V_j * \sum_C \sum_R X_{crj} / 2 ]. \end{aligned}$$

The objective function given above is minimized by GRG2.5 subject to following C \* R constraints:

$$\sum_{j=1}^n VU_j * X_{crj} < CAP_{cr}.$$

Each of the above constraint states that the volume capacity of location (c,r) is not exceeded for any (c,r) in the warehouse. The set of constraints

$$X_{crj} > 0 \quad j = 1, 2, \dots, n$$

are incorporated in the lower bounds for all variable being equal to

zero.

Additional  $n$  constraints for reorder quantities given by,

$$\sum_C \sum_R X_{crj} = Q_j \quad j = 1, 2, \dots, n$$

are introduced. The starting  $Q$  values are Economic Order Quantities for the  $n$  items. These are reduced successively using GRG2.5 until the lowest value for total cost function is obtained.

The subroutine GCOMP is called internally by optimizing routine GRGSUB, which has values of arguments for variables, constraints, objective function and bounds for these which have been scaled. Hence the variables are unscaled before the functions are evaluated and the resulting values are scaled before being passed on to GRGSUB. Figure 4.3 describes the general logic for subroutine GCOMP described in this section.

#### 4.3.2 GRGSUB Optimization Routine

The main program gives the values for variables, objective function and constraints and the upper and lower bounds for the same. These values are used by the GRGSUB subroutine through its argument list to arrive at an optimized solution. The main program also gives default or specified values for GRGSUB arguments that determine nature of input/output information printed, memory requirement etc.

Economic Order Quantities for various items provide the initial or starting point for the optimizing process. These form the upper bounds for the constraint functions corresponding to the reorder quantities. Lower bounds for these functions are chosen initially so that at the end of an optimization cycle, the reduced reorder quantities are not significantly lower than the EOQ values. This is done to allow fairly

close approximation of the probabilities  $P(z,y)$  for different retrieval locations.

The initial values of the variables, and functions and their lower and upper bounds are scaled using subroutine SETSC. GRGSUB optimizes the objective function using the scaled values described above and gives scaled output for the optimized results. These are unscaled and new reduced reorder quantities are used as the upper bounds for functions corresponding to reorder quantities and new lower values for the lower bounds are selected. For those items which did not have reorder quantities reduced, the lower bounds are left unchanged. This completes one cycle of optimization and resulting values of variables, constraints, objective function and their bounds are used as the starting point for the next cycle. Figure 4.4 illustrates the general logic for one optimization cycle described above.

These optimization cycles are repeated until no further reduction in the total cost is achieved in successive cycles, or reduction in the total cost for successive cycles is insignificant or until a predetermined number of iterations have been completed.

#### 4.4 Summary and Conclusions

The solution methodology proposed in this chapter for optimizing Interleaving Warehouse Layout generates a more cost effective reorder quantity and item location for each item stored within the warehouse. For each optimization cycle, total reduced cost and its components, the travel cost and inventory cost are obtained.

The COI and EOQ models assume continuous variables. The solution

methodology used in the optimization of the Interleaving Warehouse Layout model also assumes continuous variables. The solution obtained by the optimization of the model must be rounded to integer values. This increases the total cost somewhat over that obtained by the optimization process, but the increase would be negligible unless the reorder quantities are very small.

## CHAPTER V

### RESULTS, CONCLUSIONS AND RECOMMENDATIONS

#### 5.1 Introduction

This chapter discusses the results of optimization of the Interleaving Warehouse Layout Model. Additional information on the input parameters of the test problem is given. Numerical values for the GRG algorithm parameters are discussed and explained. The optimal solution derived is given next with the conclusions drawn from the computational experience. The chapter closes with recommendation for further research.

#### 5.2 Profile of Input Parameters

Two hypothetical problems were generated using realistic item and warehouse parameters. These problems were optimized using the Interleaving Warehouse Layout Model developed in this research to test the optimizing algorithm employing GRG2.5 discussed in Chapter IV.

Corresponding to any Unique Address location  $(c,r)$ , there is a variable  $X_{crj}$  for an item  $j$ . Hence, the number of unknown variables equals the number of Unique Address locations in the warehouse times the number of items stored within the warehouse. A smaller representative problem of 5 items stored in a warehouse with 12 storage locations was chosen to yield a problem with 60 variables. A second larger problem of 25 items stored in a warehouse with 72 storage locations was chosen to yield a problem of 1800 variables.

From previous studies conducted by Wilson [2], item parameter ranges were adopted to offer a wide variety of product size, product value, order size, and order frequency. Individual item data was

generated using a uniform distribution within these ranges:

Item volume	$VU_j$	(ft <sup>3</sup> /unit);	1.00-11.00
Average order size	$AOS_j$	(items/order):	1-30
Order frequency	$OP_j$	(orders/week):	7-80
Item value	$V_j$	(dollars/unit):	1.00-20.00.

The warehouse layout was chosen to permit the warehouse storage capacity to exceed the storage capacity needed to store Economic Reorder Quantities of all items. For the 5 item problem, warehouse storage capacity was approximately 9187 ft<sup>3</sup> and for the 25 item problem, warehouse capacity was approximately 55123 ft<sup>3</sup>. Storage volume for all Unique Address storage locations of the sides 27.67' long, was chosen to be 765.6 ft<sup>3</sup>. This is a realistic representation of a warehouse in the fact that approximately fifty 48" x 46" x 24" pallets may be stored in one location. Aisle width of 10' is large enough to allow most types of fork lift trucks to turn around.

Inventory carrying costs and reorder costs were chosen to be realistic and similar to the values reported by Wilson [2]. The inventory carrying costs and reorder costs are \$0.006 per week and \$5.00 per order respectively for all items. Kallina and Lynn [12] note that order picking costs typically dominate stock setting costs. A dollar value for cost of travel per foot (DOL) of \$0.003 was used to be consistent with this relationship.

Tables 5.2.1 and 5.2.2 give pertinent data for items and warehouse for the five items and the 25 items problems respectively.

Table 5.2.1. Input Data for 5 Item Problem.

ITEM	VU	AOS	OP	V
1	5.12	14	77	7.57
2	3.05	24	70	6.22
3	2.45	27	51	8.73
4	3.46	4	78	18.71
5	9.28	10	29	10.69

No. of Columns: 4, No. of Rows: 3

No. of U.A.C. Storage Locations: 12

Warehouse Storage Capacity: 9187 ft<sup>3</sup>

Table 5.2.2. Input Data for 25 Item Problems.

ITEM	VU	AOS	OP	V
1	5.12	14	77	7.57
2	3.05	24	70	6.22
3	2.45	27	51	8.73
4	3.46	4	78	18.71
5	9.28	10	29	10.69
6	7.20	12	63	1.24
7	5.34	22	10	13.32
8	6.15	27	67	2.99
9	10.18	21	72	4.48
10	6.34	18	23	16.40
11	3.96	1	72	7.06
12	1.68	20	57	17.63
13	5.78	1	77	6.63
14	4.46	4	32	8.60
15	4.65	17	71	6.47
16	5.08	20	8	3.05
17	5.05	19	56	10.77
18	3.63	27	7	10.37
19	4.81	15	36	17.04
20	5.00	11	44	19.30
21	4.15	15	63	7.59
22	5.19	28	62	19.80
23	3.56	6	70	12.41
24	8.07	8	17	3.56
25	10.11	22	11	8.51

No. of Columns: 9, No. of Rows: 8

No. of U.A.C. Storage Locations: 72

Warehouse Storage Capacity: 55123 ft<sup>3</sup>

### 5.3 GRGSUB Optimization Routine

There are 60 unknown variables in the 5 item problem with 12 storage locations and 1800 unknown variables in the 25 items problem with 72 storage locations. Although a large number of these variables have a value of zero at the beginning of the optimization process, different numbers of these variables may assume non-zero values at later stages of optimization. The approach described next identifies a large number of these variables that remain zero in value throughout the optimization process, i.e., their upper and lower bounds are both equal to zero. Sorting out these fixed variables considerably decreased the CPU time needed for the optimization process.

Items are allocated in the warehouse using the COI rule, i.e., the item with lowest COI value is located in the most preferred location and so on. Also, the optimization process uses EOQ values as the base case and attempts to reduce the reorder quantities in successive iterations to values far below EOQ values. Hence, in the base case, once an item is located in a location belonging to a specific Preference Classification, variables for that item corresponding to all locations belonging to higher Preference Classifications will always have values equal to zero. For example, in the 25 items problem, item 11 has the lowest COI value and is placed in Preference Classification 1, which has only one Unique Address Location: (1,1). Thus, for item 11, 71 other variables for the 71 remaining locations in the higher Preference Classifications 2 through 16, have fixed values of zero throughout the optimization process. For the 25 items problem, 1110

such fixed variables were obtained.

Also, a hypothetical optimized layout is considered, where the final reorder quantities have highly optimistic values of 0.4 times the Economic Order Quantities for the individual items. Here, the approach described earlier is applied in reverse. Thus, once an item is in a location belonging to a particular Preference Classification, variables for that item corresponding all locations belonging to lower Preference Classifications values will always have values equal to zero. If during actual optimization of the model, an item is allocated to a location in the preference Classification same as that considered in the hypothetical layout, the next lower Preference Classification is adopted as the new lower limit for allocating that item.

With help of this approach, a total of 1399 variables out of the 1800 variables for the 25 items problem were determined to have fixed values of zero throughout the optimization process. For the 5 item problem, 30 out of the 60 variables were determined to have fixed values of zero.

#### 5.3.1 Bounds for Variables and Functions

For the variables with fixed values of zero, the upper and lower bounds were both equal to zero. For the other variables, lower bounds were equal to zero and upper bounds were equal to the storage capacity of the location ( $765.6 \text{ ft}^3$ ) divided by the unit volume of the corresponding item. For the 72 functions (for the 25 items problem) corresponding to the capacity constraints for the 72 Unique Address storage locations, upper bounds were  $765.6 \text{ ft}^3$ , which is the storage

capacity of each location. To promote rapid convergence to the optimal solution, functions corresponding to the first 28 most preferred locations had lower bounds equal to or very close to the upper bounds. This made sure that these storage locations, which are closer to the input/output point than the remaining 44 storage locations in the warehouse, remained filled to the capacity to keep travel costs at minimum. If the final optimal solution had the reduced warehouse utilization approaching 40%, the lower bounds of some of the functions would have been reduced to zero. For the five items problem, functions corresponding to first 6 of the most preferred locations had lower bounds equal to or very close to the upper bound.

For the 25 functions corresponding to reorder quantities of the 25 item problem, the base case upper bounds were the EOQ values. The lower bounds for these were selected at 0.915 times the EOQ values. In the next iteration, the new reduced reorder quantities were used as the upper bound and lower bounds were selected at 0.915 times the reduced quantities. For the items that did not have reduced reorder quantities, the bounds were left unaltered. These procedure was followed iteratively, to insure that the fit parameter  $s^*$  used to evaluate the probabilities continued to provide good approximation of the probabilities for accessing a retrieval location, until it was reevaluated in the next iteration.

### 5.3.2 Argument Values for GRGSUB Routine

The default values for the arguments of the optimizing subroutine GRGSUB yielded optimal solution for most iterations. Lasdon and Warren

[4] describe these arguments thoroughly, giving their default and optional values and their applications. For the Interleaving Warehouse Layout Model, values of three of these arguments may need to be altered. Argument FPNEWT had the default value of  $1.0E-04$ , which meant that a constraint was assumed to be binding if it was within this value of one of its bounds. This was changed to  $1.0E-06$ . The other two arguments FPSTOP and NNSTOP determined when the optimization would stop. If the fractional change in the objective was less than FPSTOP (default value  $1.0E-04$ ) for NNSTOP (default value 3) consecutive iterations, the optimization program was stopped. For FPSTOP and NNSTOP, optional values of  $1.0E-06$  and 6 respectively were adopted. Use of the optional values for these three arguments helped continuation of the optimization process when the algorithm did not make any progress.

#### 5.4 Results for the 5 Items Problem

Table 5.4.1 and the Table 5.4.2 summarize the results for the warehouse problem with 12 storage locations and 5 items. The base case inventory costs were 29.95% of the total cost and the base case travel costs were 70.05% of the total cost. 90.85% of the warehouse capacity was used up to store the Economic Order Quantities for the 5 items. The items characteristics were such that the base case inventory costs were a high 29.95% of the total costs as compared to approximately 20% or less found typically in larger warehouses. Consequently, the travel costs were proportionately lower than those typical of larger problems, as a result a smaller reduction in the base case total cost was

Table 5.4.1. Results for the 5 Item Problem - I.

Iteration No.	Total Cost,\$	Inventory Cost,\$	Travel Cost,\$
0	355.15	106.38	248.77
1	353.26	106.54	246.72
2	351.53	107.11	244.42
3	349.78	107.97	241.81
4	347.57	108.60	238.97
5	333.28	109.55	223.73
6	324.90	112.24	212.66
7	322.89	112.37	210.52
8	322.65	112.76	209.89
9	322.29	113.54	208.75
10	321.95	114.13	207.82
11	321.66	114.13	207.53
12	321.59	114.26	207.33
13	321.54	114.31	207.23
14	321.51	114.31	207.20

Table 5.4.2. Results for the 5 Item Problem - II.

	Total Cost	Inventory Cost	Travel Cost
Base Case	355.15	106.3	248.78
Final Solution	321.51	114.31	207.20
% Change	-9.47	+7.93	-16.71
Base Case % of Total Cost		29.92	70.05
Final Solution % of Total Cost		35.55	64.45

Base Case Warehouse Utilization: 90.85%

Final Solution Warehouse Utilization: 54.64%

was achieved.

The optimal solution reduced the reorder quantities to obtain a warehouse utilization of 54.64%. The optimal total cost was 9.47% less than the base case total. The optimal inventory costs and travel costs were 35.55% and 64.45% respectively of the optimal total cost. The COI order for the 5 items in the final solution was the same as that for the base case. Figure 5.4 gives a clustered bar chart depicting variations of the total cost, travel cost and inventory cost with the optimization iterations.

### 5.5 Results for the 25 Items Problem

Table 5.5.1 and Table 5.5.2 give the results of optimizing the total cost function for the 25 item warehouse with 72 storage locations. The base case inventory costs were 17.80% of the total cost, and the base case travel costs were 82.20% of the total cost. The base case utilization of the warehouse was 94.40%.

With larger number of items of more varied characteristics, the 25 item problem produced greater percentage reduction in the total cost than that produced by the 5 item problem. The final solution total cost was 20.77% less than the base case total cost. The final solution inventory costs had risen to 30.32% of the final total cost and the optimized travel costs had been reduced to 69.68% of the final total cost. The final solution warehouse utilization was 51.48%. Figure 5.5 gives a clustered bar chart depicting variations of the total cost, travel cost, and inventory cost with the optimization iterations.

## Cost vs. Iterations

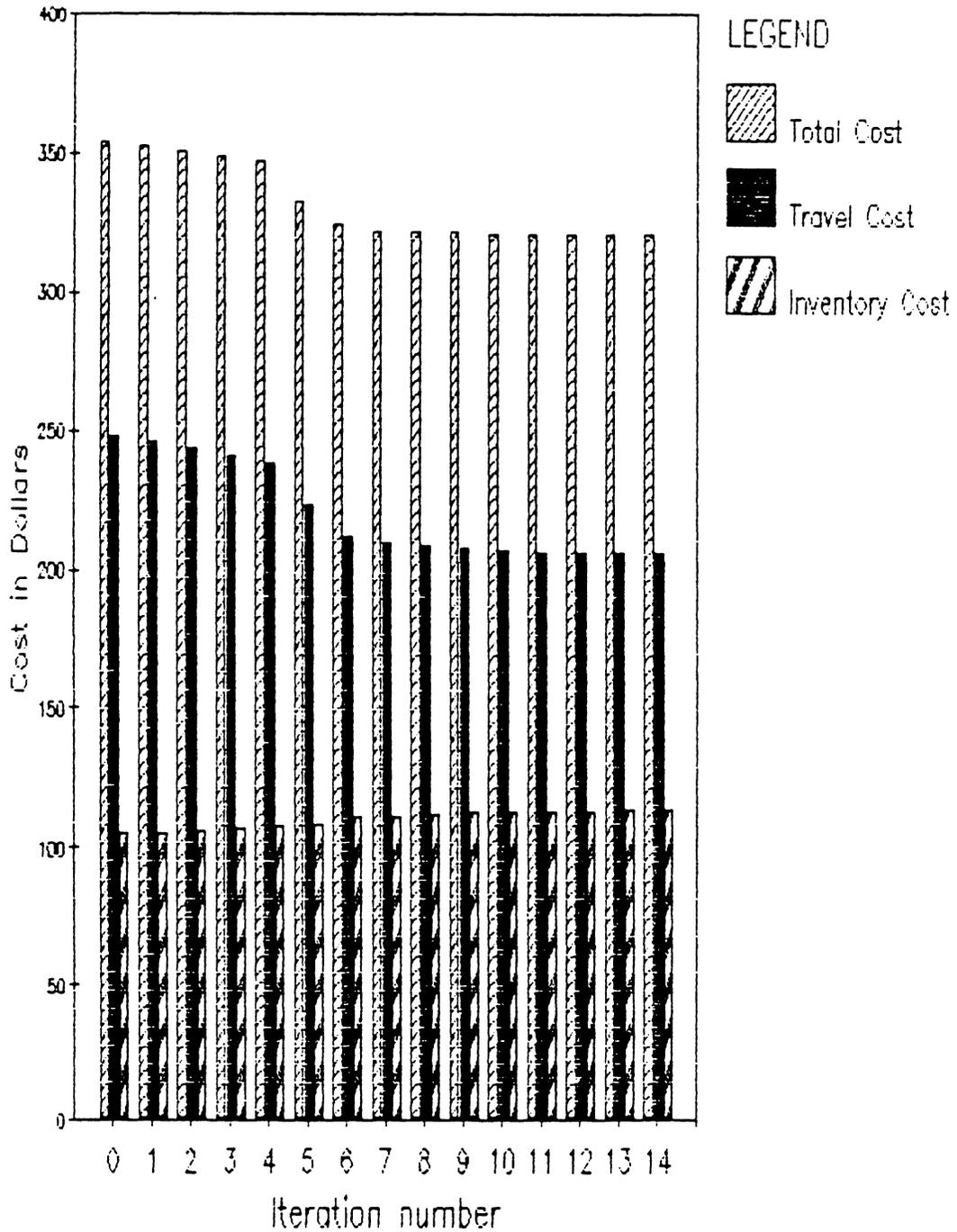


figure 5.4 Cost Chart for 5 Item Problem.

Table 5.5.1. Results for the 25 Item Problem - I.

Iteration No.	Total Cost,\$	Inventory Cost,\$	Travel Cost,\$
0	2533.98	451.05	2082.93
1	2501.62	455.81	2045.81
2	2448.89	463.53	1985.33
3	2377.12	471.33	1905.79
4	2318.64	479.12	1839.52
5	2294.78	481.68	1813.10
6	2289.36	482.07	1809.29
7	2286.83	482.96	1803.87
8	2238.55	488.51	1750.04
9	2182.42	500.17	1682.25
10	2121.47	512.55	1608.92
11	2088.56	518.16	1570.40
12	2075.31	521.02	1554.29
13	2073.88	521.54	1552.34
14	2041.12	525.99	1515.13
15	2036.88	538.81	1498.07
16	2029.72	559.37	1470.35
17	2018.96	573.41	1445.55
18	2011.32	588.58	1422.74
19	2009.12	601.92	1407.20
20	2008.86	607.88	1400.98
21	2007.66	608.53	1399.13
22	2007.56	608.76	1398.80

Table 5.5.2. Results for the 25 Item Problem - II.

	Total Cost	Inventory Cost	Travel Cost
Base Case	2533.98	451.05	2082.93
Final Solution	2007.56	608.76	1398.80
% Change	-20.77	+34.96	-32.84
Base Case % of Total Cost		17.80	82.20
Final Solution % of Total Cost		30.32	69.68

Base Case Warehouse Utilization: 94.40%

Final Solution Warehouse Utilization: 51.48%

## Cost vs. Iterations

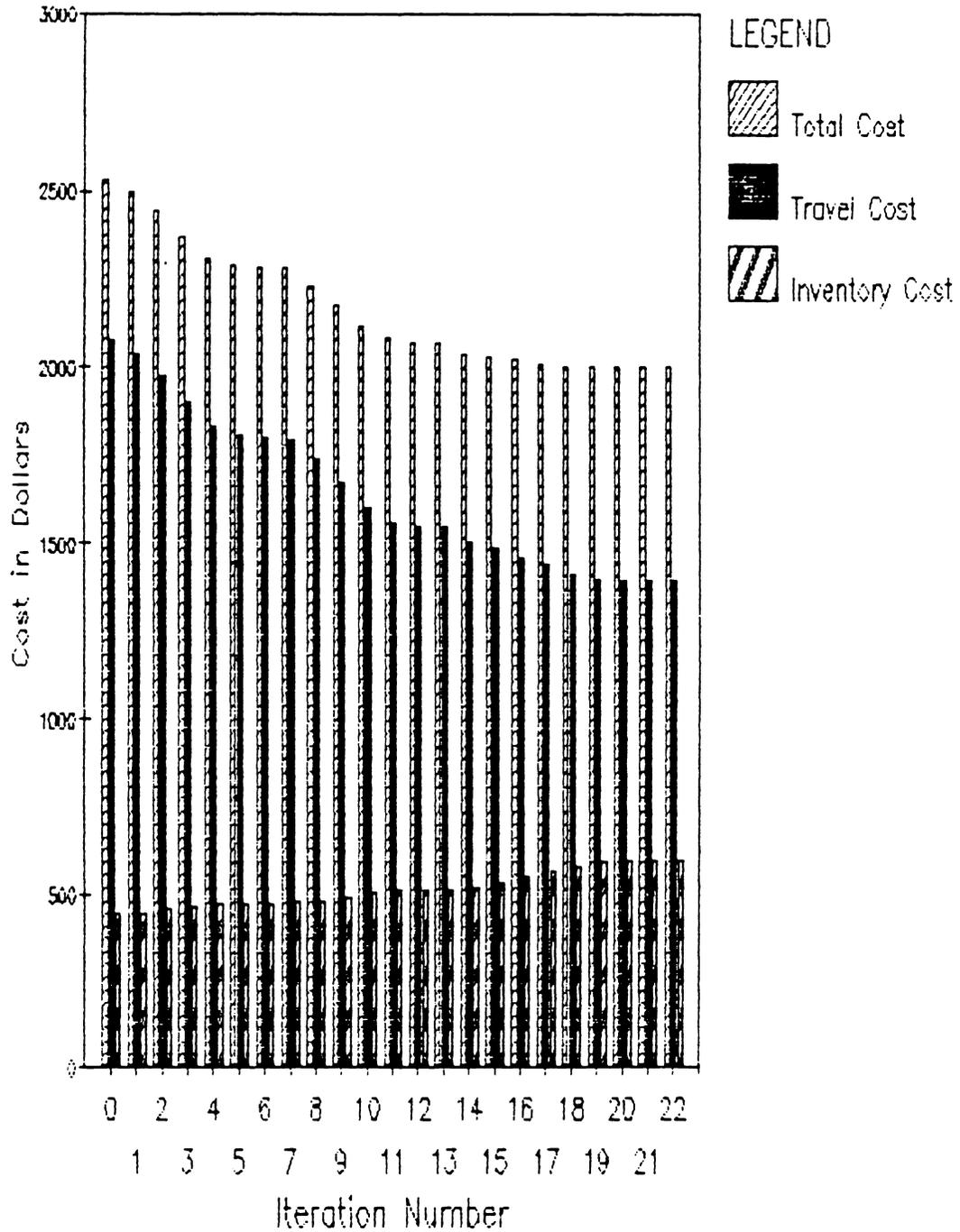


figure 5.5 Cost Chart for 25 Item Problem.

### 5.5.1 Comparison with Solutions from Literature

Kyle [5] used a heuristic algorithm to reduce the total cost for the 25 items. Results obtained using this heuristic algorithm are not directly comparable to results obtained by the optimization used in this research as different formulas for travel cost computations were used in the two studies.

Kyle [5] assumed that when the order picking vehicle travels to a storage location, the distance traveled is the rectilinear distance from the input/output point to the corner of that location nearest to the input/output point. The model used in this research assumes a more realistic point, the center of the storage location, as the point up to which the order picking vehicle has to travel to store or retrieve an order. This allows for a more accurate measure of the average distance traveled to reach a storage location. This assumption requires an additional rectilinear travel distance of half the length of the side of a storage location along x and y axes. Thus each travel of the order picking vehicle to any location in the warehouse requires additional travel equal to one length of the side of a storage location (27.67'). Hence, total travel cost for the base case in this research is 2082.93 dollars, compared to 1767.99 dollars in the model used by Kyle [5].

Final result obtained by Kyle reduces the base case total cost by 26.81% and produces a warehouse utilization of 55.25%. This research reduces the base case total cost by 20.77% and yields a warehouse utilization of 51.48%.

## 5.6 Conclusions

This research developed a mathematical model that considered quantity and location problems for an Interleaving Warehouse simultaneously. Starting from a base case obtained by the traditional Economic Order Quantity model and an initial layout, the model developed was optimized by reducing the reorder quantities. As the reorder quantities were reduced, the items stored within the warehouse were moved closer to the input/output point. The travel costs constitute a very substantial portion of the total cost and hence, reducing the reorder quantities produced reduction in travel costs that were only partly offset by the increase in the inventory costs. The increase in the inventory costs resulted from using reorder quantities lower than the Economic Order Quantities. Optimization of the model used in this research thus involved tradeoff between increasing the inventory costs and reducing the travel costs.

A Generalized Reduced Gradient optimization method as implemented in the computer code GRG2.5 was used to minimize the total cost function subject to the operational constraints for the warehouse. For the 25 item problem, the total cost was reduced by 20.77% and the warehouse utilization was decreased from the base case utilization of 94.40% to the final optimal utilization of 51.48%. This formulation of the Interleaving Warehouse Layout design provides a very useful optimizing tool to the warehouse management to reduce costs. Also as warehouse utilization is reduced by the reduction in the reorder quantities, it is possible to carry additional items in the warehouse.

This helps in maximizing throughput of the warehouse.

### 5.7 Recommendations for Further Research

The model developed in this research provides analysis of item allocations and reorder quantities to achieve a minimum total cost for an Interleaving Warehouse. The methodology developed in this research for optimizing warehouse location and quantity problems may be improved by incorporating the following recommendations for further research.

1) The mandatory interleaving policy considered in this research may be relaxed. In such a case, a queue of pallets waiting to be stored must be considered in the analysis. The pallets in the queue may be selected for storage and routed utilizing a sub-optimal procedure within the model. Thus, the routing problem may be combined with the simultaneous analysis of layout and inventory problems.

2) The model can be refined if the probability distributions of the external demand for the items is available. The interleave probabilities could then be computed utilizing the above probability distributions.

3) The location of the input/output point may be changed from the lower left corner to some other point of the warehouse. Also, more than one input/output points may be considered.

4) A three dimensional extension of the model developed in this research may be considered for further research. Such a model would be useful for designing and developing automated storage and retrieval systems for tape units, microfilms and books for large libraries dealing with such items.

5) A simulation approach for optimizing the model may be investigated. This approach will be capable of dealing with stochastic model parameters. Azadivar [19] shows that when warehouse systems become stochastic and more complex, better results could be obtained by applying optimization to responses obtained from the simulation model of the system, than by approximating it with a mathematical model.

6) A more comprehensive model for the problem considered in this research would include routing of multiple storages and/or retrievals.

7) The optimization process investigated in the research has been established to be feasible. But, this procedure has been found to be tedious. An approach that would take advantage of decomposition of the model into smaller manageable subproblems is worth investigating. This would be a substantial contribution to the development of user friendly software for the warehouse personnel on the floor.

8) Easily implementable heuristic algorithms for optimally managing a realistic warehouse may be developed with help of the insights obtained about the model from this and further research.

## REFERENCES

- [1] Ballou, R. H., "Improving the Physical Layout of Merchandise Warehouse", Journal of Marketing, Vol. 31, July, 1967, pp. 60-64.
- [2] Wilson, H. G., "Order Quantity, Product Popularity, and the Location of Stock in a Warehouse", AIIE Transactions, Vol. 9, 1977, No. 3, pp. 230-237.
- [3] Graves, S. C., Hausman, W. H., and Schwarz, L. B., "Storage Retrieval Interleaving in Automatic Warehousing Systems", Management Science, Vol. 23, May 1977, pp. 935-945.
- [4] Lasdon, L. S., and Waren, A. D., GRG2.5 User's Guide, Department of Computer and Information Science, College of Business Administration, Cleveland State University, May 1982.
- [5] Kyle, D. M., "An Interleaving Warehouse Model", Department of Industrial Engineering and Operations Research, V.P.I. & S.U., Blacksburg, 1985.
- [6] Zimmerman, G. W., "The ABC's of Vilfredo Pareto", Production and Inventory Management, Vol. 16, 1975, pp. 1-9.
- [7] Lee, L., Jr., and Dobler, D. W., Purchasing and Material Management Texts and Cases, 3rd edition, McGraw-Hill Book, Co., 1977.
- [8] Petterson, R., and Silver, E. A., Decision Systems for Inventory Management and Production Planning, John Wiley and Sons, 1979.
- [9] Harmatuk, D. J., "A Comparison of Two Approaches to Stock Location", The Logistics and Transportation Review, Vol. 12, 1976, pp. 282-284.
- [10] Hausman, W. H., Schwarz, L. B., and Graves, S. C., "Optimal Storage Assignment in Automatic Warehousing Systems", Management Science, Vol. 22, no. 6, February 1976, pp. 629-638.
- [11] Francis, R. L., and White, J. A., Facility Layout and Location an Analytical Approach, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.
- [12] Kallina, C., and Lynn, J., "Application of the Cube-Per-Order Index Rule for Stock Location in a Distribution Warehouse", Interfaces, Vol. 7, No. 1, November, 1976, pp. 37-45.
- [13] Francis, R. L., "On Some Problems of Rectangular Warehouse Design and Layout", Journal of Industrial Engineering, Vol. 12, October, 1967, pp. 595-604.

- [14] Mallette, A. J., and Francis, R. L., "A Generalized Assignment Approach to Optimal Facility Layout", AIIE Transactions, Vol. 4, No. 2, June, 1972, pp. 144-147.
- [15] Lasdon, L. S., Waren, A. D., Jain, A., and Ratner, M., "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming", ACM Transactions on Mathematical Software, Vol. 4, No. 1, March 1978, pp. 34-50.
- [16] Lasdon, L. S. and Waren, A. D., "Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Problems", in "Design and Implementation of Optimization Software", H. Greenberg, ed., Sijthoff and Noordhoff, pubs., 1979.
- [17] Abadie, J. and Carpentier, J. "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints", in Optimization, R. Fletcher (ed.), Academic Press, London; 1969, pp. 37-47.
- [18] Murtagh, B. A. and Saunders, M. A. "Large-scale Linearly Constrained Optimization", Mathematical Programming, Vol. 14, No. 1, January 1978, pp. 41-72.
- [19] Azadivar, F., "A Simulation Optimization Approach to Optimum Storage and Retrieval Policies in an Automated Warehousing System", Proceedings of the 1984 Winter Simulation Conference, pp. 207-214.

The vita has been removed  
from the scanned document