AN INTERLEAVING WAREHOUSE LAYOUT MODEL

by

Daniel McDowell Kyle

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:

C. J. Malmborg, Co-Chairman

S. Balachandran, Co-Chairman

M. H. Agee

March, 1985

Blacksburg, Virginia

AN INTERLEAVING WAREHOUSE LAYOUT MODEL

by

Daniel McDowell Kyle

(ABSTRACT)

This thesis describes the development and implementation of an Interleaving Warehouse Layout Model. Traditionally, the space allocated to items in a warehouse is determined on the basis of inventory cost considerations. With space requirements taken as given, the actual assignment of items to locations in the warehouse is carried out independently. Assuming an interleaving ("dual command") order picking method and the simple economic order quantity inventory model, it is demonstrated that the quantity and location problems must be considered simultaneously in order to achieve a minimum total cost (order picking cost plus inventory cost). A heuristic optimization technique is developed and applied to a set of realistic, hypothetical problems. This model allows warehouse management to assess the tradeoffs in handling costs among various stock arrangements and reorder quantities to achieve a minimum total cost.

ACKNOWLEDGEMENTS

Many people have contributed to the completion of this thesis and I would like to formally recognize their assistance.

I am particularly grateful to the chairman of my committee, Dr. Charles J. Malmborg, who is held in the highest esteem for his unfaltering guidance and support.

I also thank the remaining members of my committee, Dr. S. Balachandran and Dr. Marvin H. Agee for their positive guidance and suggestions.

Gratitude is extended to , who formatted and typed the many drafts cheerfully and efficiently.

Special thanks go to my parents, , whose encouragement and support sustained me throughout.

Finally, I wish to thank my wife, , for her patience, understanding, and encouragement throughout my graduate career.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

The cost of handling goods in a warehouse can amount to over half of the total operating expenses, and represent a significant portion of total operating revenues (1). There are two significant contributing factors to the materials handling expense in most warehouses; the order picking method and the stock or merchandise location. The question dealt with in this research involves the determination of a method for improving the physical layout of the merchandise so that materials handling expenses are decreased.

Every item stored in the warehouse has a handling cost associated with it that may depend on its particular location. This cost is the expense incurred from routing the item through the warehouse, that is, the cost of moving the item from a receiving dock to a storage location(s) and then to a shipping dock (input/output point). Obviously this cost is uniquely dependent on the picking discipline employed. Depending on the warehouse design, item bulk, and distribution of orders, a number of picking disciplines may be employed. Two of the most popular methods being the out and back order picking method and the interleaving ("dual command") order picking method. The out and back method visits one location (a storage or retrieve) between returns to the input/output point. While the interleaving order picking method visits two locations (a storage and a retrieve) between returns to the input/output point.

Reorder quantities must also be considered since they define the space requirements for all items. Reducing reorder quantities, while still meeting demand, will allow items to be moved closer to the input/output point, thereby reducing the order picking costs. The tradeoff is between the increase in inventory costs, resulting from the deviation from the EOQ reorder quantities, and the corresponding decrease in travel cost, allowed by moving all items closer to the input/output point.

Today's warehouse manager needs a tool that will aid him in assessing the tradeoffs in handling costs among various stock arrangements, order picking disciplines, and reorder quantities to achieve a minimum total cost.

1.1 STATEMENT OF THE PROBLEM

There are two basic problems with warehouse layout models intended to serve as planning and analysis tools for warehouse managers. First, space requirements for items are obtained from the simple economic order quantity inventory model. Models have tended to take item space requirements as given while the actual assignment of items to storage locations within the warehouse is carried out independently. Under this assumption, reduction in space requirements is impossible. Wilson (18) states that the quantity and location problems must be considered simultaneously in order to achieve a minimum total cost (order picking costs plus inventory costs).

The second problem pertains to the order picking method used. Most warehouse models assume an out and back order picking method ignoring the potentially higher throughput that interleaving systems permit (6). Interleaving systems sometimes called "dual address" systems, are capable of visiting up to two storage locations between returns to the input/output point. A typical round trip would include a visit to a storage location, where the order picking vehicle would perform a storage operation, interleave travel to a retrieve location, where the order picking vehicle would perform a retrieve operation and then return travel to the input/output point.

1.2 OBJECTIVE

The primary objective of this research is to develop a warehousing model, using an interleaving order picking method, that will generate reorder quantities and stock locations to minimize total cost (order picking costs and inventory costs), while continuing to meet demand requirements.

A base case solution will be obtained using the EOQ reorder quantities. A methodology is developed to proceed from one set of improved reorder quantities to the next in such a way that the total cost is reduced.

The operations impact of the model is to provide opportunities for a more efficient use of warehouse space. The systematic allocation of items not only results in a lower total cost of the warehouse, but also in an increase in throughput of the warehouse.

In summary, the objective is to show that the optimal reorder quantities should be less than the EOQ reorder quantities to reduce the total cost of the warehouse.

1.3 OVERVIEW OF THE RESEARCH

The following chapters are included in this research document: Literature Review, Modeling Approach, Model Implementation, Computational Experience and Conclusions and Recommendations. After a brief introduction, the literature review discusses the base case economic order quantity inventory model. The ABC principle expands on the practical necessity of subdividing the inventory for more effective management control. Different types of stock allocation rules are also examined. Travel times for different storage assignment rules are then derived. The warehouse layout problem is formulated using two different methodologies. Finally, a solution procedure is described to solve the quantity and location problems simultaneously.

Chapter III, the discussion of the modeling approach, opens with an introduction. The assumptions used in the research model are then expanded upon. A profile of different types of warehouse layout problems that the model will be applicable to solving is then introduced. A new approach using the ABC principle, called the Modified Location ABC Curve, is discussed. This allows approximation of the probability of a particular location access. A formulation of travel distance between storage locations is presented. The total cost equation (objective function) is then derived and expanded upon. Chapter III closes with a summary of the interleaving warehouse model.

Chapter IV, the discussion of model implementation, begins with a profile of test problem input parameters. An in depth description of the computational logic of the program is reviewed. A discussion of the solution algorithm procedure is introduced. The chapter closes with a summary of the Interleaving Warehouse Layout Model implementation.

Chapter V, the discussion of the computational experience gained from the model, explains the generation of all test problems. Model comparisons are carried out and the results from all test problems are discussed in depth. Interpretation and explanation of the results generated from all test problems are elaborated within the chapter. The chapter closes with a presentation of the conclusions drawn from this research.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

For many years, the assignment of materials to storage locations has received considerable attention from warehouse planners. Wilson (18) states that in the assignment of warehouse space to inventory items, the amount of space allocated to each item is traditionally determined on the basis of inventory or production cost considerations. The inventory cost is typically extracted from the simple economic order quantity inventory model. The actual assignment of items to storage locations within the warehouse is then carried out independently, with the space requirements taken as given.

One criterion for assigning inventory items to locations within the warehouse is the cube-per-order index rule (COI), 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. The objective being to minimize order picking costs. The justification for focusing on order picking costs has been discussed by Kallina and Lynn (10).

The analytical results of Francis (4), and Mallette and Francis (14), support the use of dedicated storage to minimize travel time of the order picking vehicle based on single command cycles. Graves, Hausman, and Schwarz (6), 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 (5) 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 (18), by assuming an out and back order picking method and the simple economic order quantity inventory model, 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 will proceed from the simple economic order quantity inventory model, where inventory costs and reorder quantities are obtained, to consideration of the quantity and location problems simultaneously producing a minimum total cost solution to the warehouse layout problem.

2.2 ECONOMIC ORDER QUANTITY

Deciding the procurement or production lot size for an item is one of the most common and unresolved questions of inventory management. The problem arises because of the need to purchase or produce quantities greater than will be used or sold at the moment. The practice of replenishing stocks in sizable quantities compared with the typical usage quantity necessitates the carrying of inventory. If the replenishment orders become too large, the resulting inventory carrying cost will dominate reordering cost. A balance must be found between ordering costs and carrying cost to minimize the total cost of inventory, while making sure that demand is met. If it is assumed the demand for an item is uniform throughout the time period, the purchase lead time is zero, and no shortages are allowed; the resulting inventory level over time may be represented graphically as shown in Figure 2.1.

The total cost (TC) for the time period is the sum of the item cost (IC) for the period, the purchase cost (PC) for the period, and the holding cost (HC) for the period. That is;

$$TC = IC + PC + HC$$

where

$$IC = \Sigma IC_j$$

$$PC = \Sigma PC_j$$

$$HC = \Sigma HC_j$$

$$TC = \Sigma TC_j.$$

The item cost for the period will be the cost per unit (V_j) times the yearly demand in units (D_j), or

$$IC_j = V_j * D_j.$$

The purchase cost for the period will be the cost per purchase (Co_j) times the number of purchases per year (N), or

$$PC_j = Co_j * N.$$

But since N is equal to the yearly demand (D_j) divided by the purchase quantity (Q_j), or

$$PC_j = (Co_j * D_j)/Q_j.$$

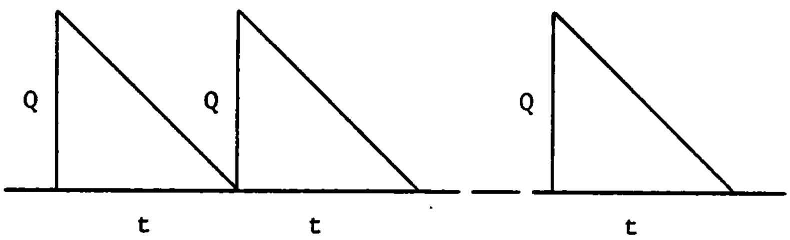

Figure 2.1. Inventory Level Over Time.

Since the interval, t, begins with Q_j units in stock and ends with none, (assuming instant replenishment, lead time = 0, and uniform demand), the average inventory during the cycle is approximated by $Q_j/2$. The holding cost (Cm) for the time period will be the holding cost per unit times the average number of units in stock for the period $(Q_j/2)$, or

$$HC_j = Cm * V_j * Q_j/2.$$

The total cost for the period of providing the required item is thus;

$$TC_j = (V_j * D_j) + (Co_j * D_j)/Q_j + (Cm * Q_j)/2.$$

The economic order quantity is the size of the order that minimizes the total cost (TC_j). The determination of the order quantity involves two opposing variable cost curves. The first curve is the order preparation cost curve. Let PC be the total cost for the time period of placing orders for the inventory item. This cost must decrease as the lot size is increased as shown in Figure 2.2.

The second variable cost curve involves the cost of holding the average inventory for a given period of time.

$$HC_j = Cm * V_j * Q_j/2.$$

These costs increase as lot size increases as shown in Figure 2.3.

The two curves added together give a total variable cost (VC_j) as shown in Figure 2.4, where $VC = \Sigma VC_j$.

Since it is apparent the total cost curve is convex, we can find the minimum economic lot size by taking the first derivative of VC_j

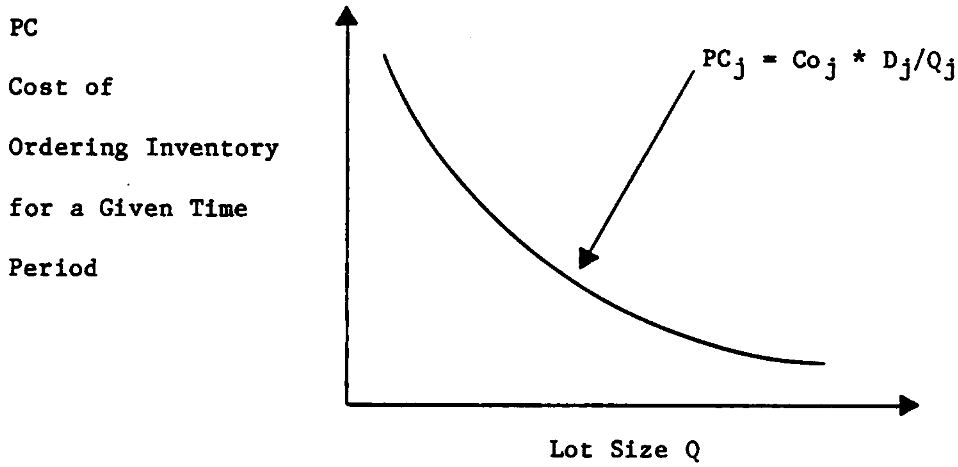

Figure 2.2. Order Preparation Cost Curve.

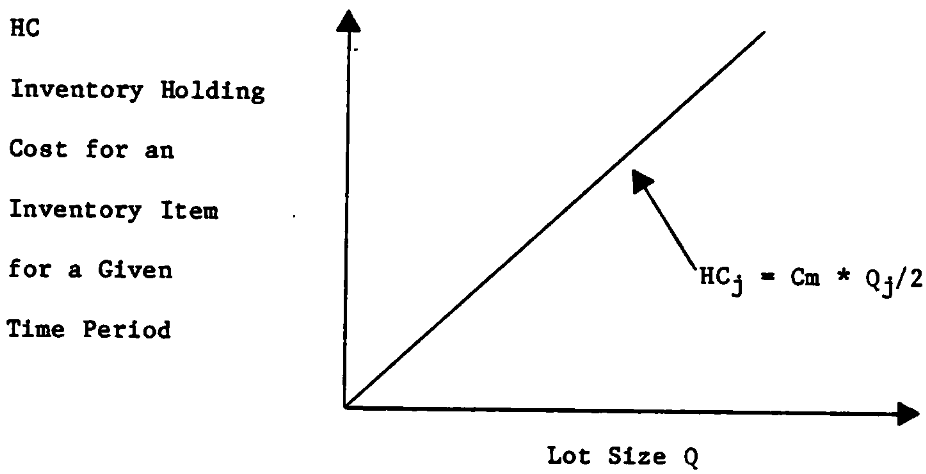

Figure 2.3. Holding Cost Curve.

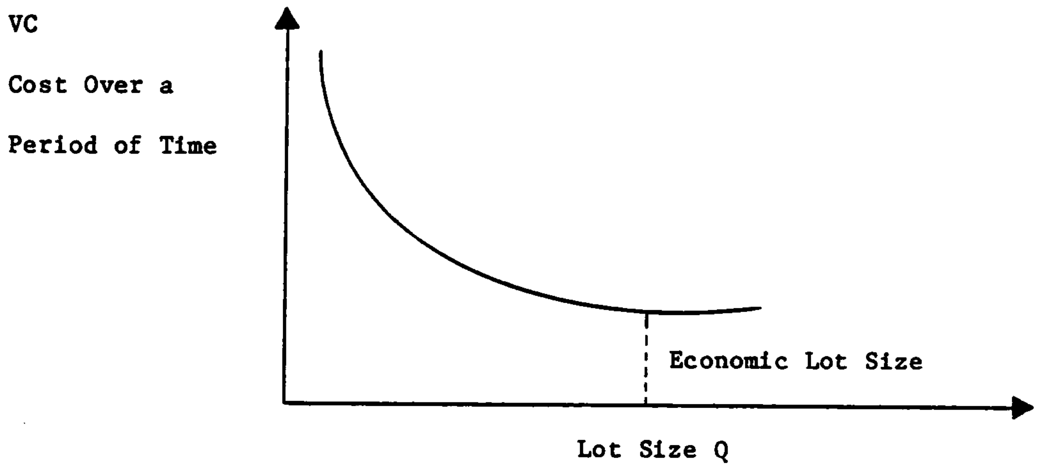

Figure 2.4. Total Variable Cost Curve.

with respect to Q_j, setting this equal to zero and solving for Q_j.

$$VC_j = (Co_j * D_j)/Q_j + (Cm * Q_j)/2$$

$$\frac{\partial VC_j}{\partial Q_j} = -(Co_j * D_j/Q_j^2) + Cm/2$$

$$Q_j = ((2Co_j * D_j)/Cm)^{1/2}.$$

Substituting Q_j into the total variable cost equation and simplifying leads to;

$$TC_j = (V_j * D_j) + [(Co_j * D_j * Cm/2)]^{1/2} + [(Cm * Co_j * D_j/2)]^{1/2}$$

Above is the total cost per period, assuming the simple economic order quantity is used.

After total inventory cost per period for each individual item has been calculated, the items are then grouped together as an inventory system. A practical model is desired by management to control this system.

2.3 THE ABC CURVE

The "ABC Principle" is recognized as an important management tool that effects and influences management's control of production and inventory systems. In the field of inventory control, the ABC principle is vitally important because it recognizes all of the individual items that make up the total inventory are not of the same relative importance. It suggests there is a practical necessity of sub-dividing an inventory for the purpose of more effective management control.

Zimmerman (20) points out that efficient management occurs when the amount of management effort is directly proportional to the importance of the item being managed, where importance is measured by inventory investment or dollar usage. A uniform control system that provides adequate controls for the high value items, over controls the low value items. Likewise, at the opposite end of the spectrum, a uniform system that is economically justifiable for the low value items, doesn't provide adequate control of the high value items. Effective management requires that the segregation of the "vital few" from the "trivial many" is obtained before a management control system can be developed that is economically justifiable for both groups.

Assuming demand characteristics are known for every item of inventory, an ABC analysis can be easily conducted. Lee and Dobler (12) define the procedure for this analysis. All items are ranked in the order of their demand (measured in average inventory investment or dollar usage). The total of these values for all inventory items is then computed. The demand for each individual item is then expressed as a percentage of the total. By going down the list and successively cumulating the individual percentages for each item and plotting these values against the percentage of items inventoried, an ABC curve can be developed as shown in Figure 2.5. It is possible to determine which items make up the first 75 percent of inventory investment, the first 90 percent, and so on.

This curve can then be divided into three parts. The first is class A items. These should receive the most attention from management, since a small percentage of items accounts for a large portion of demand. The second is class B items, which are of secondary

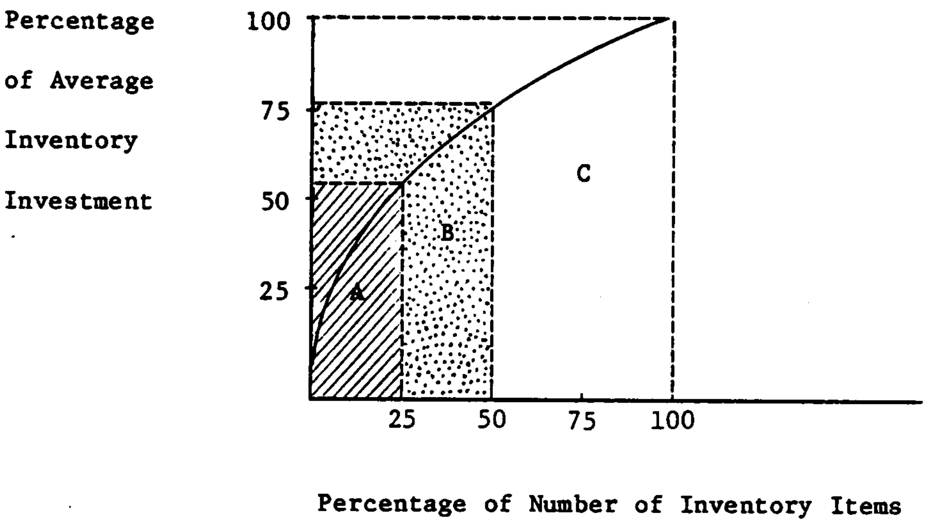

Figure 2.5. The ABC Curve.

importance in relation to class A. For class B items, a moderate percentage of items account for a moderate portion of demand. Class C items, where a large percentage of items account for a small portion of demand, comprises the third class of items.

Peterson and Silver (15) point out other basis for ABC types of classifications are sometimes used. Some large volume distribution centers plan the allocation of warehousing space (the warehouse layout problem) on the basis of cubic feet per unit using an ABC curve. Items with a low ratio of required cubic footage to the number of daily orders are placed nearest the input/output point, guaranteeing the greatest stock-volume moves the shortest distance.

2.4 STOCK LOCATION

The warehouse layout problem is to find the optimal location of items in a warehouse, where optimization corresponds to the minimization of order picking costs (handling costs) associated with assembling orders and transporting them to the input/output point of the warehouse.

Ballou (1) writes that one of the most common layout methods is one based on product popularity. The underlying assumption of this method is the total handling cost is directly related to the total distance traveled in the warehouse. The location of fast moving items near the input/output point, and the slower moving items to the rear of the items with more rapid turnover, will minimize the total travel distance and therefore, the total handling costs. This ensures items requiring a large number of trips to satisfy demand are located near the input/output point, so the distance traveled per trip is low.

A second layout method commonly used is based on the volume of the items stored. The main idea of this rule is that more items of smaller size can be located near the input/output point, which will satisfy a greater percentage of demand, reducing the total distance traveled. Therefore, items are located so that smaller items are nearest to the input/output point and the larger items are at the more distant points in the warehouse.

A third method of stock location combines both turnover and size of an item. This method is called the cube-per-order index rule. Items with a low ratio of required cubic footage (or floor space) to the number of daily orders, in which the items are requested, are placed nearest the input/output point. This guarantees the greatest stock-volume moves the shortest possible distance.

2.4.1 THE CONTINUOUS CASE

A linear-programming model, presented by Harmatuck (7), is composed of an objective function and any number of restrictions expressed in the form of inequalities.

The objective is to minimize the sum of all handling costs for all merchandise in the warehouse, assuming an out-and-back order picking method. (Given a storage request, the order picking vehicle travels from the input/output point to the assigned rack location, stores the pallet and returns empty to the input/output point. Similarly, given a retrieve request, the order picking vehicle travels empty from the input/output point to the appropriate rack location, removes the pallet, and returns to the input/output point. The order picking vehicle is only able to pick one item during one trip out and back to the input/output point.) One practical restriction is the amount of products allocated to any one subarea (bay), cannot exceed the capacity of the subarea. This practical constraint can be expressed mathematically, and there will be as many constraints as there are subareas in the warehouse. An additional restriction, that can also be written mathematically, is needed to assure enough space is allocated to each product in order to meet demand requirements for that product. Items are characterized by their popularity, order characteristics, and cubic dimensions. Subareas are characterized by their capacities and the cost of movement to the input/output point.

Predetermined quantities of items are to be assigned to subareas of a warehouse. A locational pattern for items which minimizes total cost of movement is sought.

2.4.2 MODEL FORMULATION

The n items to be located are characterized by:

DD_j = number of periods demand for item j for which space must be allocated (time).

CU_j = volume of space required by a unit of item j ($length^3$/unit).

AOS_j = average order size of an item j (units).

OD_j = the number of orders per period for item j ($time^{-1}$).

The m subareas are characterized by:

CAP_i = the volume capacity of location i ($length^3$).

$COST_i$ = the cost of moving an order of any item from location i to the input/output point (dollars).

The locations are ordered so that $COST_i < COST_{i+1}, \ldots$ where

$i = 1,2,\ldots,m-1$.

The problem is to determine the number of units of item j assigned to location i, X_{ij}, for each i and j.

Heskett proposes a cube-per-order index rule for deciding the subarea (locations) of each item. The index for item j is defined as;

$$COI_j = CU_j * AOS_j * DD_j.$$

Items with lower COIs are located closer to the input/output point.

The linear programming approach to minimize material handling cost solves the following problem.

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

subject to:

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

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

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

Here, the objective function is just the order picking costs incurred per unit time.

2.4.3 COMPARISON OF THE COI AND LP METHOD

Harmatuck (7) 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 (given) quantities of space required for the n items may

be allocated among m locations in any manner. More than one item may be found in a particular location and an item may be split among several locations. There are no compatibility constraints prohibiting close proximity of item pairs.

3) The cost of moving an order (any quantity) of an item from a particular location to the shipping area is a constant, depending only on location.

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

To demonstrate the equivalence of the two methods it is only necessary to show a violation of the COI rule will never lower costs. Assume item 1 is located in subarea A and item 2 in a more remote subarea B. It follows, by the definition of the cube-per-order index rule that:

$$COI_1 < COI_2$$

$$COST_A < COST_B.$$

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

$$(COST_B - COST_A)/COI_1$$

and the change in cost for item 2 will be

$$(COST_A - COST_B)/COI_2.$$

Summing and simplifying these expressions we obtain

$$(COST_B - COST_A) * (1/COI_1 - 1/COI_2)$$

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

Due to the use of computer controlled inventory systems, randomized storage assignment is one of the most common storage assignment rules used today. Incoming items are assigned to a pallet. The contents of the pallet are then communicated to a minicomputer, which assigns the pallet to a location in the storage area (warehouse), and records the assigned location in the computer. Upon receipt of a request for an item, the computer obtains the pallet location from computer storage and directs an order picking vehicle to retrieve the pallet. With randomized storage, material can be stored anywhere in the system, reducing space requirements and resulting in a very efficient throughput.

The random storage assignment rule states any pallet is equally likely to be stored in any of the m locations of a warehouse. Consider a warehouse with an out-and-back order picking method, where stores alternate with retrieves, and there are m storage locations. The closest open location will be the location where the preceding retrieve has taken place, or a location of equal distance from the input/output point. Therefore, random storage approximates the closest open location rule (i.e., a pallet is stored in the closest location to the input/output point regardless of the pallets turnover rate). Under the assumption of no interleaving (the system is given one command; a store or a retrieve), Hausman, Schwarz, and Graves (8), show the expected one way travel time for random storage assignment (T_r) is;

$$T_r = \bar{y} = 1/m \sum_{i=1}^{m} y_i$$

where y_i = ranked one way travel time to travel from the input/output point to location i, i = 1,2,...,m; ranked so that $y_i < y_{i+1}$, all $i < m$.

Using the turnover-based assignment rule, it can be shown the expected one way travel time is minimized by placing the highest turnover pallet closest to the input/output point. Under this scheme, they have derived expected one way travel time for turnover based assignment to be;

$$T_r = \sum_{i=1}^{m} y_i * \lambda_i / \sum_{i=1}^{m} \lambda_i$$

where λ_i = turnover rate of location i.

The percentage reduction in expected one way travel time reached with turnover based assignment can be represented by;

$$I_T = ((T_r - T_T)/T_r) * 100.$$

The above equation can be evaluated for any specific y_i and λ_i, i,j = 1,...,m. However, before proceeding, we will introduce continuous representations of the distance and turnover functions. These representations considerably reduce the difficulty of analysis and yield results sufficiently close to the discrete analysis to merit our attention.

2.5.1 THE CONTINUOUS REPRESENTATION

In the continuous representation the discrete functions y_i and λ_j are approximated by continuous functions y(i) and λ(j). Furthermore, the indices i and j themselves become continuous on the interval (0,1). The storage rack is assumed to be square in time, where x_1 and x_2 co-ordinates of each point are the horizontal and vertical travel times respectively, for the crane to reach a given point. Without loss of generality, we will select a unit of travel time measurement to yield a rack with unit sides, as shown in Figure 2.6.

Assuming the following points:

1) Each pallet holds only one item type.

2) All storage locations are the same size, as are the pallets themselves. Therefore, 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 and the input/output conveyor transfer pallets. The input/output point 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 is capable of moving both vertically and horizontally simultaneously, and its vertical and horizontal speeds are such that the time to reach the row most distant from the input/output point equals the time to reach the most distant column (i.e., the system is square in time). The crane travel will be time rather than distance.

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

7) Actual time for the crane to load or unload a pallet at the input/output point or storage location (called transfer time) is ignored.

8) The turnover frequency of each item is known and constant

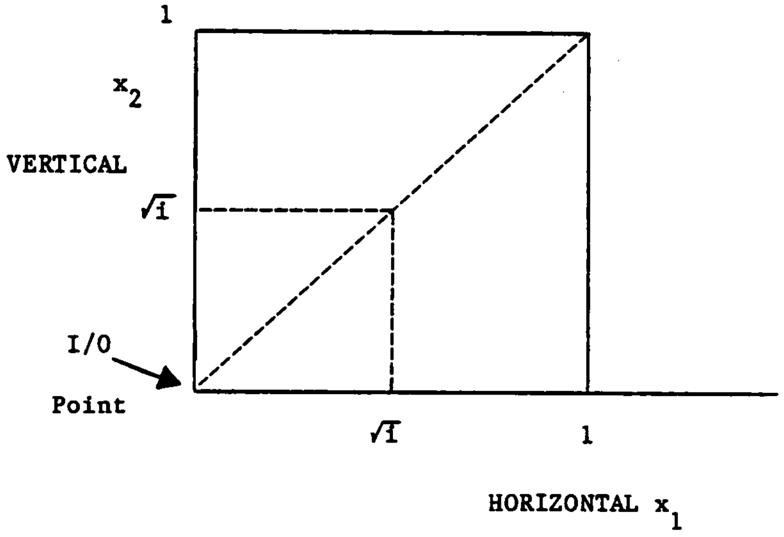

Figure 2.6. Continuous Representation of Storage Rack.

throughout time. Turnover frequency is the number of times a given item requires storage and retrieval in some time period. Alternatively, it is the reciprocal of the average length of storage time for the item. This assumption is relaxed when class-based turnover assignment is considered.

9) Short-run dynamic considerations are ignored. We are interested here only in the long-run average behavior of the system.

Hausman et al. (8) derive the continuous distance functions y(i) (the travel time to the ith percentile) as;

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

They use the ABC phenomenon for inventories and the basic EOQ model to estimate the distribution of pallet turnover. The ABC curve is represented by the function;

$$G(i) = i^{s} \qquad 0 < s < 1$$

where demand is measured in full pallet loads.

Now let D(i) = demand rate (pallets per unit time) of item i and Q(i) = the economic order quantity of item i. By definition $G(i) = i^s = \int_0^i D(j)dj / \int_0^1 D(j)dj$.

Letting $\int_0^1 D(j)dj = 1$;

$$i^s = \int_0^i D(j)dj$$

which has the solution;

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

then $Q(i) = (2K * D(i))^{1/2}$ where K is the ratio of ordering costs to holding cost, and is assumed to be constant for all items. 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;

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

Using the demand rate (D(i), pallets per unit time) of item i, they calculate the total number of rack locations required for the average inventory of all items. They then proceed to determine the time index i, of the jth pallet i(j). Substituting this into the demand rate D(i), the turnover rate for the jth pallet, λ(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.

The expected one way travel time under random storage assignment is;

$$T_r' = E[y(i)] = \int_0^1 i^{1/2} d_i = 2/3.$$

For full turnover based assignment the expected one way travel time is;

$$T_T' = \int_{j=0}^{1} \lambda(j)y(j)dj / \int_{j=0}^{1} \lambda(j)dj$$

which, evaluated becomes;

$$T_T' = 4s/(5s+1).$$

The percentage improvement over random storage assignment is;

$$I'_T = (100)((2/3) - (4s/(5s+1))/(2/3))$$

$$= (100)(1-s)/(5s+1).$$

Hausman et al. (8) using different ABC curves and their corresponding s values have shown improvement I'_T increases considerably as the skewness of inventory distribution increases.

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

The symbol R represents the partitioning of the unit square into two classes (a two class system), where class 1 is used for higher turnover pallets and class 2 for lower turnover pallets as shown in Figure 2.7. Hausman et al. (8) have derived the one way travel time under this two class system as a function of R, $T'_2(R)$, to be;

$$T'_2(R) = (2/3)(R^{(5s+1)/(s+1)} + (1-R^3)(1-R^{4s/(s+1)})/(1-R^2)).$$

Testing a wide range of inventory distributions they found the two class system yields approximately 70% of the potential gain of a fully turnover based system. The most highly skewed inventory distribution yielding the largest improvement.

For the three-class turnover-based system a potential gain of approximately 85% over the full turnover-based system was found.

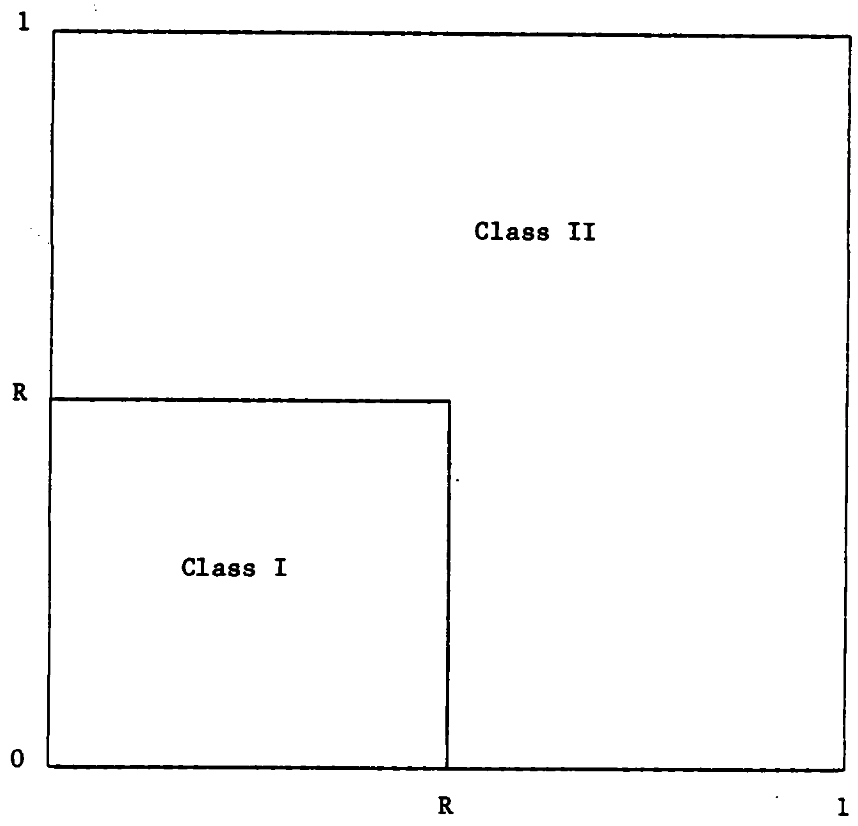

Figure 2.7. Graphical Illustration of Two-Class Storage Assignment.

Hausman et al. (8) have written a computer program to determine the relationship between the continuous model and the discrete model. The results indicate, as the measure of skewness of the inventory distribution increases, the error involved in the continuous approximation increases significantly. In particular, the true potential improvement is less than that indicated by the continuous approximation. The reason for this is the increased importance given to the first few pallets and their discreteness relative to the continuous approximation.

Given an out-and-back order picking method (single command), significant potential reductions of crane travel times in automatic warehousing systems are possible using a class-based turnover assignment policy, rather than the closest open location (essentially random) policy.

2.6 STORAGE ASSIGNMENT USING THE INTERLEAVING ORDER PICKING METHOD

This section extends the work previously reported on storage assignment rules to include interleaving. Interleaving systems (dual address systems), are capable of visiting up to two rack locations between successive returns to the input/output point. After completion of a given storage request, the order picking vehicle can move directly to the location of the next retrieve before returning to the input/output point. Obviously, interleaving systems permit potentially higher throughput than out and back order picking methods. However, maximum throughput of dual-address systems may only be obtained from an optimal combined storage assignment/interleaving policy.

Operating performance is measured in terms of expected round trip travel times which is, the expected time for the system to complete one store and one retrieve. For mandatory interleaving policies (MIL), the expected round trip travel time is twice the expected one way 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 corresponding interleaves). For the non-interleaving policy (NIL), the expected round trip travel time is four times the expected one way travel time.

The following storage assignment and interleaving rules are examined by Graves, Hausman, and Schwarz (6):

1) Random Storage Assignment (RAN).

2) Class-Based Storage Assignment (C2 or C3).

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

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

The highest turnover pallet is assigned to the location closest to the input/output point. This is the limiting case of 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 class-based storage assignment rules. An attempt was made to match the above storage assignment rule 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 Retrieves (MIL/FCFS).

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

3) Mandatory Interleaving with Selection Queue of K Retrieves (MIL/Q=K).

Retrieves need not be chosen FCFS within the first K retrieves.

Graves et al. (6) 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

These results are summarized below:

RAN/MIL/FCFS. In the continuous model the expected one way travel time given a random storage assignment has been determined as 2/3. They have found an expected interleave time of 7/15 for the case in question. Thus, the expected round trip time, denoted RT(RAN/MIL/FCFS) is

$$RT(RAN/MIL/FCFS) = 2(2/3) + 7/15 = 27/15$$

regardless of the inventory turnover distribution.

FULL/MIL/FCFS. The full turnover based storage assignment rule minimizes expected one way travel time. Graves et al. (6) point out this might not minimize expected round trip travel time. They have derived the expected interleave time (L) for this policy to be

$$L(FULL/MIL/FCFS) = \frac{2z^2}{3}\left\{\frac{[48z^3+36z^2+42z-48+96(2^{-2z-2})]}{(4z+1)(2z+2)(2z+1)(2z)(2z-1)}\right\}$$

where $z = 2s/(s+1)$. It has been shown previously the expected one way travel time for full turnover based storage is $4s/(5s+1)$. It follows that the expected round trip travel time for the case in question is;

$$RT(FULL/MIL/FCFS) = 2(4s/(5s+1)) + L(FULL/MIL/FCFS).$$

The best classed-based storage assignment policies use classes whose outer boundaries are square in time. These are called "square-L" boundaries. This is due to the fact that crane velocities are square in time and all travel in NIL policies is to and from the input/output point, (crane travel time between rack location (x_1,x_2) and the input/output point is $\max(x_1,x_2)$).

In interleaving systems travel also occurs between storage locations. Thus, the best class boundary shape in terms of expected round trip travel time need not be - and is probably not - of the "square-L" type. The exact shape of the optimal boundaries are quite difficult to specify.

Graves et al. (6) investigated the sensitivity of 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 "square-L" boundaries.

They derive the respective round trip times for the following policies;

C2/MIL/FCFS

C3/MIL/FCFS

C2/MIL/Q=K

C3/MIL/Q=K

The storage classifications R_1 and R_2 (as the case may be) are chosen numerically to minimize the expected round trip travel time for different values of s. Where R_1 and R_2 indicate the partioning 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 (high turnover) and their "discreteness" relative to the continuous representation.

They state in general, class-based storage reduces round trip travel time over random storage. More rack locations are needed for class-based than for 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 qf 4% to 5% is needed for the three class system.

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

By adopting the selection queue (no longer taking retrieves 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 in the retrieve queue. By considering only the first few requests, degredation in customer service is kept to a minimum.

2.7 THE WAREHOUSE LAYOUT PROBLEM

As previously stated the warehouse layout problem is to find the optimal location of items in a warehouse, minimizing the order picking costs (handling costs) associated with assembling orders and transporting them to the input/output point of the warehouse. Francis and White (5) describe two formulations of the problem, using the general assignment model and the factoring assumption model.

Define region L, as part of the floor in a warehouse where items are stored. Assume it is divided into m grid squares of equal size, numbered from 1 to m. Suppose n items are to be stored in the warehouse. Let A_j be the total number of grid squares item j will occupy. Let the warehouse have p input/output points (docks) at fixed locations. Define d_{ki} as the distance between dock k and the center of grid square i. Let S_j be the number of grid squares taken up by item j; for example if item j takes up four grid squares ($A_j = 4$), which have the numbers 7,14,15, and 19, then $S_j = (7,14,15,19)$. It is 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 item j travels between dock k and its storage region is given by;

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

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 that 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 its 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 the 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;

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

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;

$$F(S_1,S_2,\ldots,S_n) = \sum_{j=1}^{n} \sum_{k=1}^{p} w_{jk}(\sum_{i \varepsilon S_j} (1/A_j) * d_{ki}).$$

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 stated above. Since sets $S_1,S_2,\ldots,S_n$ determine the storage regions of the items, the collection of all sets $(S_1,S_2,\ldots,S_n)$ represents a warehouse layout. The problem is to find a layout to minimize 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 $i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$ 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 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, it follows that;

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

Since each grid square is also taken up by one item, it follows that;

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

S_j is the collection of all grid squares i for which $x_{ij} = 1$; that is, $S_j = (i\colon\ x_{ij} = 1)$, it follows that;

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

The total cost expression may 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;

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

The total cost expression can then be rewritten as;

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

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

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

subject to: $\sum_{i=1}^{m} x_{ij} = A_j \qquad j = 1,\ldots,n$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,\ldots,m$$

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

Taking each A_j to be 1 gives the assignment problem. When some A_j are positive integers greater than 1 a generalization of the assignment problem is obtained. The generalized assignment problem is itself a special case of the transportation problem. The common way to solve the generalized assignment problem is as a 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;

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

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

$$x_{ij} = 0 \quad \text{for} \quad j = 1,\ldots,n+1 \text{ and } i = 1,\ldots,m.$$

The objective function remains the same. In a general linear-programming context the inclusion of a dummy item is equivalent to the inclusion of slack variables 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;

$$w_{n+1,1},\ldots,w_{m+1,p}$$

to be zero and use the equation;

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

This has some intuitive appeal since it is based on the fact that the dummy item has no exchange with any dock. That is;

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

2.7.2 THE DISCRETE WAREHOUSE LAYOUT PROBLEM USING THE FACTORING ASSUMPTION

In this section, an assumption will be made permitting a least cost layout to be found by a simpler procedure than the generalized assignment approach.

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 exists numbers $u_1,u_2,\ldots,u_m$ and $v_1,v_2,\ldots,v_p$ such that;

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

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

Given the factoring assumption the total cost equation can be rewritten as;

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

where

$$f_i = \sum_{k=1}^{p} v_k * d_{ki} \quad \text{for} \quad i = 1,2,\ldots,m.$$

An equivalent condition for the matrix W to factor is;

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

where

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

and

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

This condition is equivalent to the original factoring condition. If $u_j = c_j$ and $v_k = w_k$ and the matrix W factors, it then follows that;

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

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

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

Solving for u_j and v_k, and then using the above expression gives;

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

This is an important intuitive result allowing us to state the factoring assumption is satisfied whenever the total number of pallets of item j traveling in and out of storage 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 BY USE OF THE FACTORING ASSUMPTION

To motivate the procedure to be developed for finding a least cost layout, consider the following two sequences 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 multiplied together, and so on, the maximum scalar product will be obtained;

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

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 the warehouse layout formulation, the first of the sequences represent the average distance between a grid square and the docks and letting the second sequence represent the numbers;

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

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

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

The problem of interest is to find the layout $(S_1^*, S_2^*, \ldots, S_n^*)$ such that $F(S_1^*, S_2^*, \ldots, S_n^*) < F(S_1, S_2, \ldots, S_n)$ for all $(S_1, S_2, \ldots, S_n)$. It will be useful to denote that class of all layouts by $H_n(L:A)$. The procedure for finding the least cost layout may be stated. For $i = 1, 2, \ldots, m$ compute;

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

and let $i_1, i_2, \ldots, i_m$ be a permutation of $1, 2, \ldots, m$ such that $f_{i_1} < f_{i_2} <, \ldots, < f_{i_m}$. A least cost layout $(S_1^*, S_2^*, \ldots, S_n^*)$ is then given, on defining $B_j = \sum_{h=1}^{j} A_h$ for $j = 1, 2, \ldots, n$ by;

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

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

$$S_n^* = (i_{B_{m-1}}+1, \ldots, i_{B_m}).$$

Consider the case of two different items. Item #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 item #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. Item #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 #1 to minimize the order picking costs.

2.8 QUANTITY AND LOCATION PROBLEMS CONSIDERED SIMULTANEOUSLY

In the previous sections we have only considered the stock location problem to achieve a minimum order picking cost. We will now expand this consideration to include the quantity and location problem simultaneously to achieve a minimum total cost, which includes order picking costs and inventory costs.

Wilson (18) considers 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. With the space requirements taken as given, 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 plus inventory cost). His analysis assumes an out and back order picking method and the simple economic order quantity inventory 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 to minimize the cost of assembling orders. The objective is generally to minimize the man hours required to pick the orders and transport them to the input/output point. It is obvious 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. First we will discuss the results for a simple out-and-back order picking method.

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

Harmatuck (7) observed that order frequency (OD_j) does not appear in the COI formula therefore, optimal location is independent of item popularity. This statement is only valid if the number of periods demand for which space is allocated (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

We will now expand the scope of analysis to embrace the inventory costs plus order picking costs. The total cost objective function, retaining the assumption of an out-and-back order picking method and the simple EOQ inventory model with fixed and known demand and lead times becomes;

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} (COST_i/AOS_j * DD_j) * X_{ij} + (Co * D_j/Q_j) + (Cm * V_j * (Q_j/2)).$$

where

- Co = cost of reordering any item (dollars).
- Cm = inventory carrying cost ($time^{-1}$).
- D_j = demand rate for item j (units/time).
- Q_j = reorder quantity for item j (units).
- V_j = value of item j (dollars/unit).

A few helpful relationships will be introduced here.

$$D_j = OD_j * AOS_j$$

$$DD_j = Q_j/D_j$$

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

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

The total cost (objective) function may now be written;

$$Z=\sum_{j=1}^{n} OD_j(\sum_{i=1}^{m} COST_i*X_{ij})/\sum_{i=1}^{m} X_{ij}+\sum_{j=1}^{n}(Co*D_j/(\sum_{i=1}^{m} X_{ij})+(Cm*(V_j/2))\sum_{i=1}^{m} X_{ij}).$$

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

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

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

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

This requires the number of units of item j assigned to location i, for each i and j, is non-negative. Note the constraint;

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

which states no shortages are allowed, has been incorporated into the objective function by substituting for DD_j. Note also 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 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 moved 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. What is needed, is a way to proceed from one set of reorder quantities to the next in such a way that the total cost, Z, is reduced.

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

$$\partial Z/\partial X_{ij} = OD_j(COST_i*Q_j - \sum_{h=1}^{m} COST_h*X_{hj})/Q_j^2 + (Cm*V_j/2) - (Co*D_j/Q_j^2).$$

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 accomodated in location i. If the change in Q_j is positive, the actual cost change will be greater (algebraically). 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 kth iteration, $Q_1^k, Q_2^k, \ldots, Q_n^k$, which have been allocated by the COI method in quantities;

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

Say that item p has the highest COI of all items allocated to i (item p was the last item allocated to location i). The increased picking cost due to moving a unit of item p from location i to location i+1 (assuming that space is available in location i+1) is;

$$\Delta i' = (OD_p/Q_p^k)(COST_{i+1})(X_{i+1,p}+1) + COST_i(X_{i,p}-1)$$

$$- (OD_p/Q_p^k)(COST_{i+1} * X_{i+1,p} + COST_i * X_{i,p})$$

$$\Delta i' = (OD_p/Q_p^k) * (COST_{i+1} - COST_i).$$

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

$$\Delta i = \Delta i'/CU_p = (OD_p/(Q_p^k * CU_p))(COST_{i+1} - COST_i).$$

The increased order picking cost attributable to the relocation results 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;

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

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

Define $Z*(Q_1^k, Q_2^k, \ldots, Q_n^k)$ 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;

$$\partial Z*/\partial Q_j = G_j = CU_j \sum_{i=i_j}^{m-1} \Delta i + OD_j(COST_{i_j} * Q_j - \sum_{i=1}^{m} COST * X_{ij})/(Q_j^k)^2$$

$$+ Cm * V_j/2 - Co * D_j/(Q_j^k)^2.$$

This is the jth element of the gradient vector. Small movements in the direction of the negative gradient will decrease 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 at the EOQ solution, and the procedure is not likely to approach zero quantities, for practical purposes, the non-negativity constraint can be 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 ΔQmax. The step size is calculated by;

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

from which the new reorder quantities are calculated;

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

The step parameter ΔQmax is left unchanged until an increase in total cost is encountered, whereupon ΔQmax 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 has been completed. The flow chart shown in Figure 2.8 depicts the logic of the procedure. It is difficult establishing the local minimum found by this iterative procedure is also a global minimum. The solution obtained is the local minimum in the vicinity of the EOQ solution. It appears unlikely any other local minima would

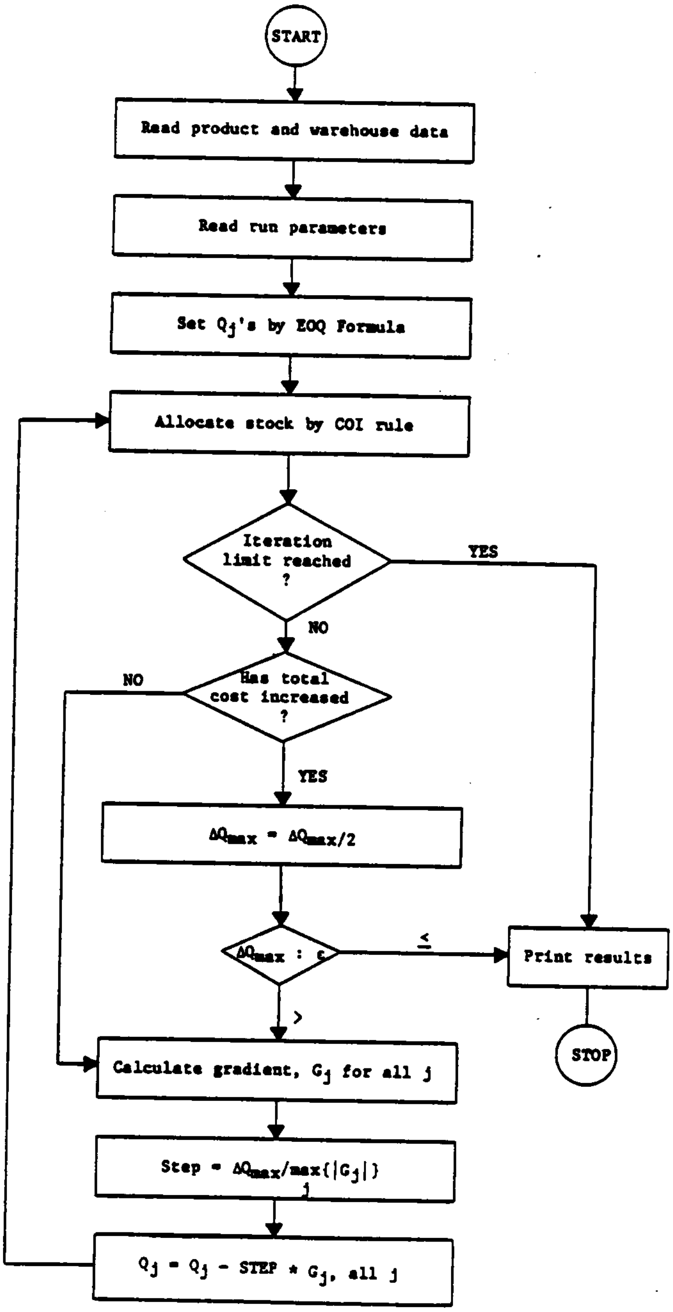

Figure 2.8. Wilson's Flow Chart.

represent a substantial improvement in terms of cost savings. In any event, the algorithm always gives a lower cost solution than the naive use of the EOQ and COI rule independently.

The algorithm described here appears to represent 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; or

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

The out-and-back order picking 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 which particular product is being retrieved, and of which other products may happen to be contained in the same order.

The primary requirement of the more involved inventory model is the first derivative of inventory cost with respect to the maximum quantity on hand must be calculable.

2.9 SUMMARY

The literature has revealed 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, optimize, or simulate one or more phases of the warehouse 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 item orders. The cost of transporting an item is dependent upon the location of the item. The location of an item is dependent upon the quantity of all items stored in the warehouse. A complete location analysis includes an assessment of all the above considerations providing a total minimum cost (order picking costs 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 made has been the assumption of the item space requirements, derived from the simple economic order quantity inventory model, are given. Wilson (18) states the quantity and location problems must be considered simultaneously in order to achieve a minimum total cost. Another major assumption frequently made 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 will incorporate an interleaving order picking method, while considering the quantity and location problems simultaneously, is desired. Just as previously developed stock location models have done, the "new" model will generate individual item locations. The "new" model will also generate individual item reorder quantities while continuing to meet demand requirements of all items.

CHAPTER III

THE MODELING APPROACH

3.1 INTRODUCTION

This chapter will present information providing insight into the concepts to be applied during the development of the Interleaving Warehouse Layout Model. Model assumptions will be stated and expanded upon. Additional information will describe why these concepts were chosen and how they will be applied. A possible relaxation of some model assumptions to encompass a wider variety of unique warehouse layout problems is also elaborated within the chapter. The chapter will close with a summary of the modeling approach.

3.2 MODEL ASSUMPTIONS

A base case solution will be obtained using the EOQ reorder quantities. A methodology is developed to proceed from one set of reorder quantities to the next in such a way that the total cost (order picking costs and inventory costs) is reduced.

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, we will assign items to storage locations based on their COI value. The COI value of an item is simply the ratio of space to be allocated for 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 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 any pallet load or any item. The assignment of multiple items to the same pallet, pallet assignment 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 a mandatory interleaving (MIL) order picking method is used. The order picking vehicle visits two locations in the warehouse, a storage location and a retrieve location, (which may be the same), between successive returns to the input/output point. If the storage location and the retrieve location are the same, the travel cost of the interleave will equal zero and the cost of travel to the storage location will equal the cost of travel to the retrieve location. We also assume statistical independence between and among stores and retrieves. By making this assumption we include the potentially higher throughput of an interleaving order picking method (6). A typical round trip consists of a storage request (a pallet load) followed by a retrieve. 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.

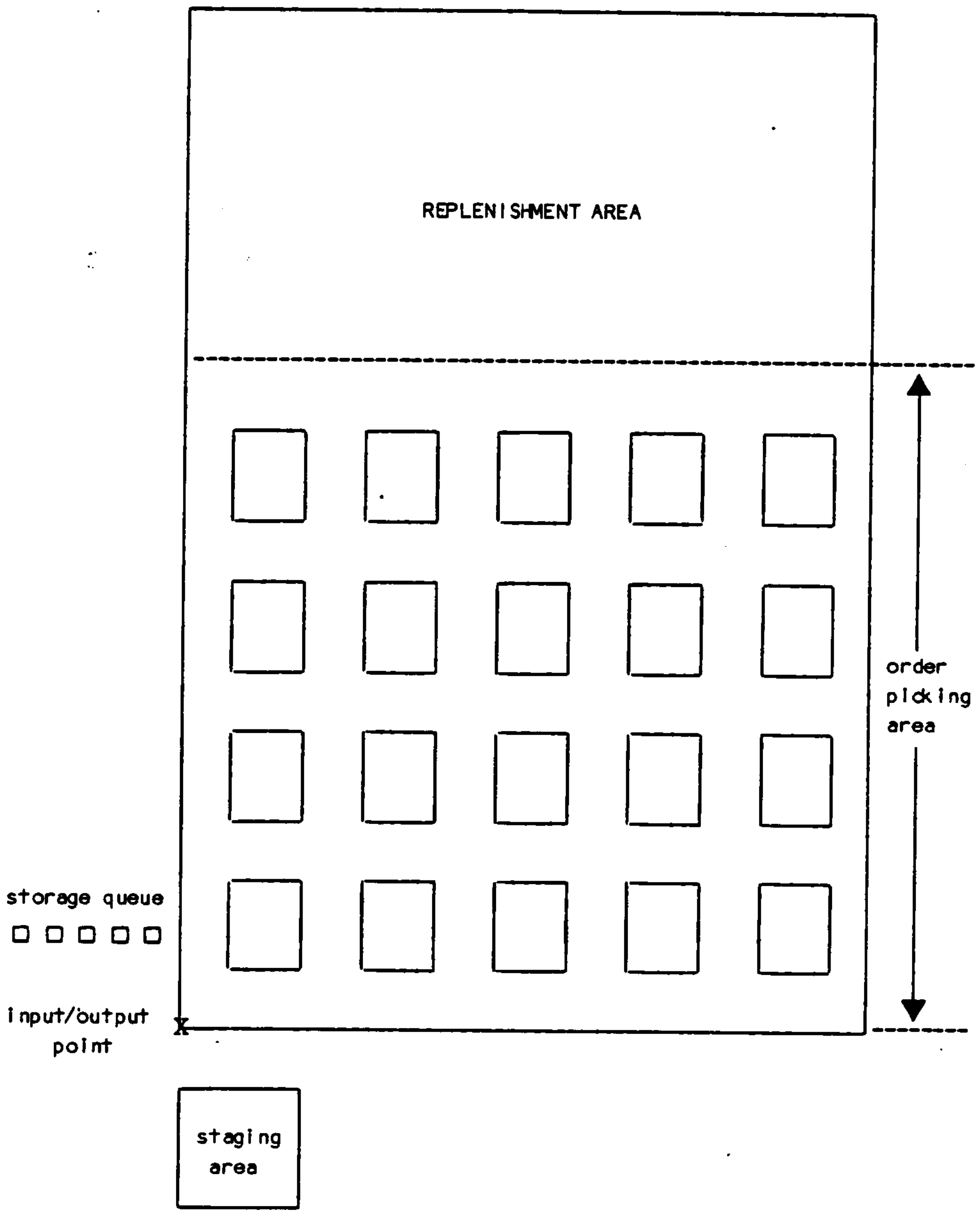

Figure 3.1. Warehouse Layout.

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 retrieve location, is ignored.

The number of order picking transactions per period of each item is known and constant.

The warehouse storage area is known. 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 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 the origin (0,0), represents the location of the input/output point. An orthoganal network of aisles running parallel to the x and y axes allows for the approximation of rectilinear travel distance between subareas (bays) and the input/output point. All aisles are the same size throughout the warehouse and all travel is by the shortest route possible. No backtracking is allowed.

This assumption allows for the cost of travel of the order picking vehicle to be approximately proportional to the rectilinear travel distance. Using the interleaving order picking method, cost of travel for all interleave combinations must be generated.

The system is bounded at the point where the order picking vehicle and the input/output conveyor transfer pallets. 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, a 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 there will be ties, and all ties are given the same classification. For example, there may be three Unique Addresses in a particular Preference class.

It has been found 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.

Finally, short-run dynamic considerations are ignored. We are interested here only in the long run average behavior of the system (steady state).

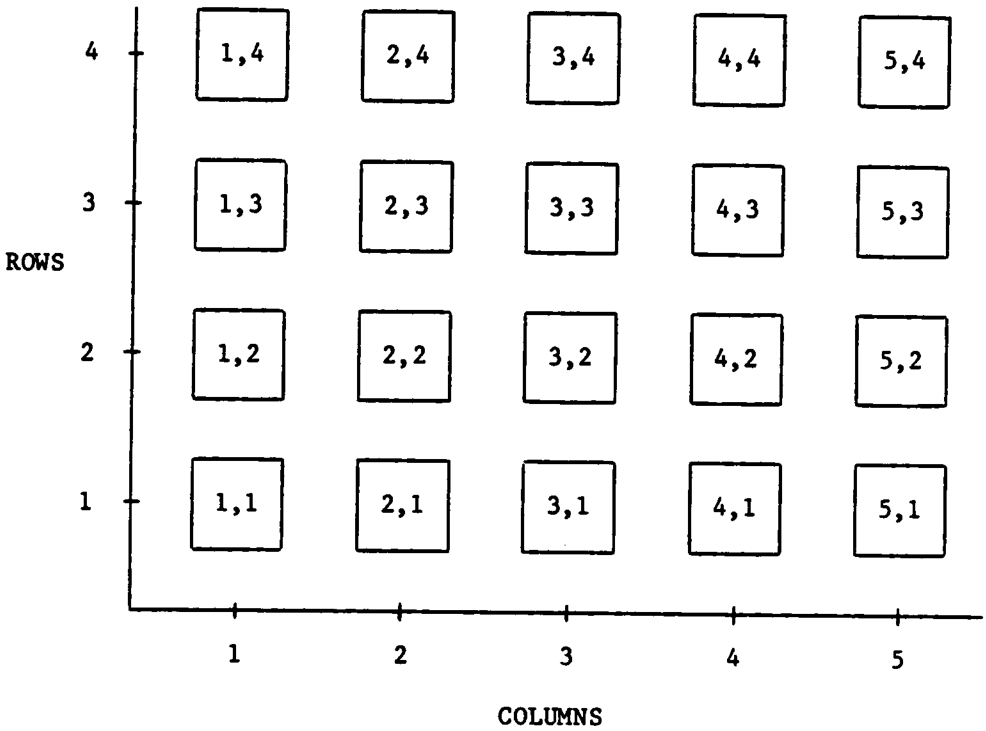

Figure 3.2. Unique Address Classification, U.A.C.

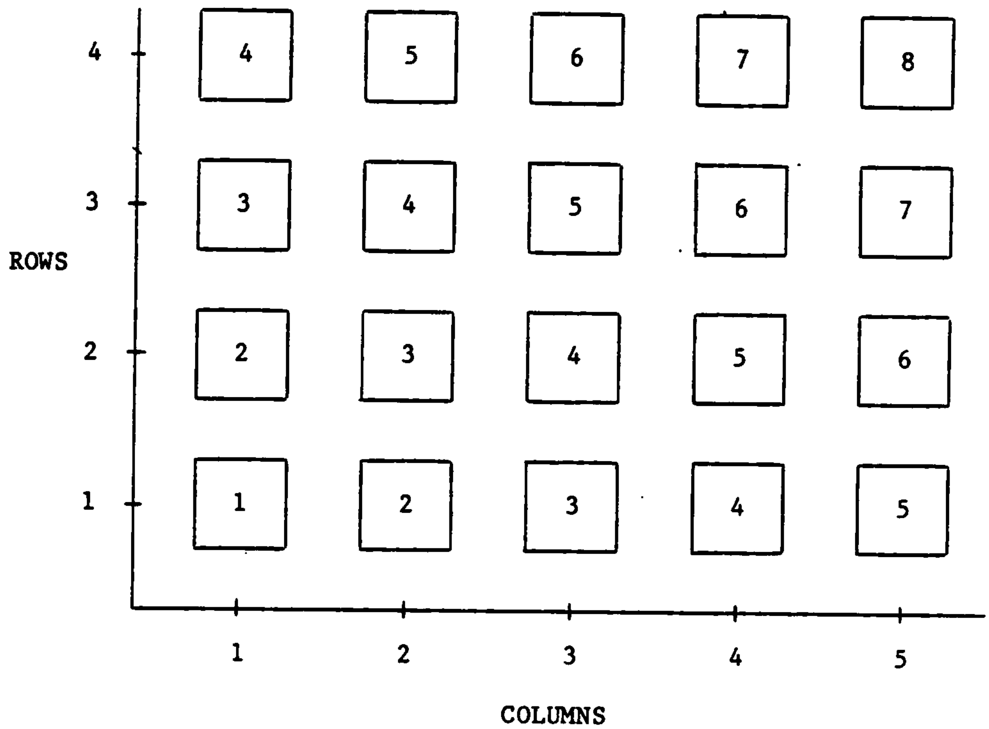

Figure 3.3. Preferrence Classification, P.C.

3.3 PROFILE OF POTENTIAL CASE PROBLEMS

The assumptions stated in the previous section restrict the warehouse layout problem addressed so different examples could be solved with a minimum effort. This section describes which assumptions can be easily manipulated or relaxed to encompass a wider variety of unique 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 capacity must be large enough to accommodate the final reorder quantities.

The algorithm works well for a distribution warehouse, using a dedicated COI assignment rule, where item assignment is based on item popularity and 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 the mandatory interleaving (MIL) rule is not strictly enforced actual travel times will be the weighted average of noninterleaving (NIL) and MIL performance. This allows for a particular queue, 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 rectangle shape discussed previously. The Preference Classification relationship and the rectilinear travel distance relationship remain true for all possible rectangular shaped warehouse layouts. Rather it is the item allocation routine that must be adapted to each individual layout.

3.4 MODEL FORMULATION

The most important facet of the modeling approach is the development of the Modified Location ABC Curve. Given this curve we are able to approximat the probability of a particular Preference Classification being accessed.

The Modified Location ABC Curve is based on the "ABC Principle", that is, all individual items making up the total inventory are not of the same relative importance. It suggests there is a practical necessity of sub-dividing an inventory for the purpose of more effective management control [20].

The n items and C x R locations, where C and R are the column and row number respectively, are characterized by:

DD_j = the number of periods demand for item j for which space must be allocated (time).

CU_j = the volume of space required by a unit of item j (length^3/unit).

AOS_j = the average order size of item j (units).

OD_j = the number of orders per period for item j (time).

X(c,r,j) = the amount of item j in location (c,r) (units).

D_j = demand rate for item j (units/time).

A few helpful relationships to be used in the future are:

$$D_j = OD_j * AOS_j$$

$$DD_j = Q_j/D_j$$

$$Q_j = \sum_{1}^{C} \sum_{1}^{R} X(c,r,j).$$

The last relationship relies on the simplistic assumptions of the EOQ model, that is maximum quantity on hand of any product is its reorder quantity.

The solution procedure starts with a given quantity ($Q_j = EOQ_j$; base case measured in units) for each item.

Assuming the demand characteristics are known for every item of inventory, the Modified Location ABC analysis can be implemented. All items are ranked in the order of their COI values

$$COI_i, COI_{i+1}, \ldots, COI_n.$$

$$COI_j = CU_j * AOS_j * OD_j * DD_j/OD_j$$

$$COI_j = CU_j * AOS_j * DD_j$$

$$COI_j = CU_j * AOS_j * Q_j/D_j$$

$$COI_j = CU_j * AOS_j * Q_j/OD_j * AOS_j$$

$$COI_j = CU_j * Q_j/OD_j.$$

The total demand for all inventoried items is computed by $\sum_{j=1}^{n} OD_j$. The demand for each individual item is then expressed as a percentage of the total demand by $OD_i / \sum_{j=1}^{n} OD_j$ where i represents the individual item number, $i = 1,2,\ldots,n$, and j represents the sum of all items, $j = 1,2,\ldots,n$. Starting with the item having the lowest COI value, we calculate the cumulative percentage of total demand for each item. To calculate the space requirement for each item we use $CU_j * Q_j$. The cumulative space requirement for each item is then obtained.

Spaces (locations) are numbered using the Preference Classification previously stated, in increasing order with the lowest value being the most preferred space. We then calculate the volume of space for each Preferred Classification value. Recalling all locations have the same volume of space (a constant), this can be done by counting the number of locations in a particular Preference Classification and multiplying by that constant.

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 due to Preference Classifications of spaces being equal (distance from the input/output point for several locations are equal) is done arbitrarily.

Starting with the lowest numbered Preferred Classification, we compute the cumulative percentage of demand stored in each Preferred Classification.

3.4.1 THE MODIFIED LOCATION ABC CURVE

To develop the Modified Location ABC Curve, the cumulative percentage demand versus the Preference Classification number for each class is plotted.

We can represent the Modified Location ABC Curve by the function

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

In this expression, s is the fit parameter for the Modified Location ABC Curve. The corresponding s value may be found by solving

$$i^s = (OD_i / \sum_j OD_j) = (CU_i * Q_i / \sum_j CU_j * Q_j)^s$$

$$\ln(OD_i / \sum_j OD_j) = s * \ln(CU_i * Q_i / \sum_j CU_j * Q_j)$$

$$s = \ln(OD_i/\sum_j OD_j)/\ln(CU_i * Q_i/\sum_j CU_j * Q_j)$$

where i represents the individual item number, i = 1,2,...,n, and j corresponds to the sum of all items, j = 1,2,...,n.

For each individual point (corresponding to each Preference Classification value) on the curve, we are able to calculate an individual s parameter. Disregarding the s value that has a cumulative percentage demand of one, sum the s values and divide by one less than the number of Preference classes used, to obtain a representative value of s* for the entire curve.

The probability of space classification Z being accessed may now be represented by:

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

where i' = Z/(total number of preference classes used), i" = (Z-1)/(total number of preference classes used).

The Modified Location ABC Curve, being a plot of cumulative percentage demand versus Preference Classification, will be unique for every different item allocation of the warehouse. Being that the curve is based on space requirements and the allocation order of all items, a change in reorder quantity of any item will effect this curve.

3.4.2 FORMULATION OF TRAVEL DISTANCE

It has been previously stated the warehouse design is rectangular in shape with a known area. The orthogonal network of aisles running parallel to the x and y axes allows for the rectilinear calculation of distance traveled between a subarea (bay) and the dock (input/output point), or, the distance traveled between one subarea and another.

The aisle width is assumed to be constant. Using the U.A.C., if (c,r) is any subarea in the warehouse layout then

$$(|c-1| + |r-1|) * LS + (|c-1| + |r-1|) * AW$$

will be considered to be the rectilinear distance that any item stored in the warehouse travels between its location at the point (c,r) and the dock at (0,0). Where LS is the length of a side of a storage area and AW is the aisle width, each value being a constant.

There is one exception to this rule, when (c,r) = (1,1) the rectilinear travel distance of storage is computed by 2(AW).

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

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

Since the cost of moving any item to any location is proportional to the rectilinear distance traveled. Multiplying the rectilinear distance traveled by a cost constant (DOL) will result in the cost of a particular move.

3.4.3 THE TOTAL COST EQUATION

The total cost equation may be broken down into two parts. The first being the order picking cost, which must capture the cost of travel to a storage location, the interleaving cost (the cost of travel from a storage location to a retrieve location), and the cost of travel from a retrieve location to the input/output point. The second part being the inventory cost of all items which includes, reordering costs and holding costs.

The order picking cost may be represented by;

$$\sum_{C}\sum_{R}[(\sum_{j=1}^{n}(X_{crj}/\sum_{C}\sum_{R}X_{crj})*OD_j)*COST_{cr}$$

$$+[\sum_{Z}\sum_{Y}\{|c-z|+|r-y|)*(AW+LS)*DOL+COST_{zy}\}*(\frac{i'^{s*}_{zy}-i''^{s*}_{zy}}{NUM_{zy}})$$

$$*(\sum_{j=1}^{n}(X_{crj}/\sum_{C}\sum_{R}X_{crj})*OD_j)]]$$

where

$X_{crj}/\sum_{C}\sum_{R}X_{crj}$ = proportion of item j in location (c,r).

$COST_{cr}$ = the cost of moving an order of any item from location (c,r) to the input/output point (dollars/order).

$COST_{zy}$ = the cost of moving an order of any item from retrieve location (z,y) to the input/output point (dollars/order).

i'^{s*}_{zy} = proportion of orders filled from the class (P.C.) of locations including (z,y) and those classes of locations closer in travel time to the input/output point.

i''^{s*}_{zy} = proportion of orders filled from the next lower class (P.C.) of locations from (z,y) and those classes of locations closer in travel time to the input/output point.

NUM_{zy} = the number of Unique Address Classifications in the Preference Classification that includes address (z,y).

$\frac{i'^{s*}_{zy}-i''^{s*}_{zy}}{NUM_{zy}}$ = probability of Unique Address Classification (z,y) being accessed.

The following relationships are helpful in linking the Preference Classification to the Unique Address Classification:

i'_{zy} = (z+y-1)/(total number of Preference classes used).

i''_{zy} = (z+y-2)/(total number of Preference classes used).

Multiplying the number of orders of all items j = 1,2,...,n at address (c,r) per period by the cost of travel per order from the input/output point to location (c,r) results in the travel cost per period of all items stored at location (c,r). Add to this, the expected interleave cost per order and the expected retrieval cost (from (z,y) to the input/output point given that (z,y) is the retrieval address corresponding to a storage operation at address (c,r)). We then multiply by the number of orders at address (c,r) per period and this results in the order picking cost of one dual command for all items. Repeating this calculation for all locations in the warehouse results in the total order picking cost for the warehouse.

The inventory cost for all items may be represented by the EOQ model as;

$$\sum_{j=1}^{n} [Co * OD_j * AOS_j/(\sum_C \sum_R X_{crj}) + Cm * V_j * \sum_C \sum_R X_{crj}/2)].$$

The inventory cost includes two opposing variable-costs. The replenishment cost expressed as cost of reorder (Co) times the demand (which may be expressed by number of orders per period (OD_j) times the average order size (AOS_j)) divided by the reorder quantity ($\sum_C \sum_R X_{crj}$). Add to this the stock setting cost expressed as carrying cost (Cm) times the value of an item (V_j) times the average inventory level of that item ($\sum_C \sum_R X_{crj}/2$). It is seen, the replenishment cost per unit time decreases as reorder quantities increase, (fewer replenishments) whereas the stock setting costs increase as reorder quantities increase (larger average inventory).

The total cost equation considering interleaves may now be written as;

$$\sum_C \sum_R [(\sum_{j=1}^{n} (X_{crj}/\sum_C \sum_R X_{crj}) * OD_j) * [COST_{cr}$$

$$+ [\sum_Z \sum_Y \{(|c-z|+|r-y|) * (AW+LS) * DOL + COST_{zy}\} \cdot (\frac{i'^{s*}_{zy} - i''^{s*}_{zy}}{NUM_{zy}})]$$

$$* (\sum_{j=1}^{n} (X_{crj}/\sum_C \sum_R X_{crj}) * OD_j)]]$$

$$+ \sum_{j=1}^{n} [Co * OD_j * AOS_j/(\sum_C \sum_R X_{crj}) + Cm * V_j * \sum_C \sum_R X_{crj}/2]$$

Inspection of the total cost equation reveals the interaction between inventory costs and order picking costs. Since the reorder quantity $\sum_C \sum_R X_{crj}$ appears in the first term, it is evident that 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 moved closer to the input/output point, thereby reducing order picking costs.

3.5 SUMMARY AND CONCLUSIONS

The foundation of the interleaving warehouse layout model has been established in this chapter. The modeling assumptions were constructed to encompass as many real life warehousing layout problems as possible. It must be realized each warehouse layout problem is unique in some way (i.e., a different layout, order picking method etc.).

Some of the modeling assumptions simply cannot be altered, dedicated storage being one of them. It is realized the use of dedicated storage takes more space but, knowing where a particular item is stored can be of greater 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 stores and retrieves 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 a very good estimate of travel costs.

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

Using the modeling assumptions stated in this chapter as a foundation for the interleaving warehouse model, a discussion of the model implementation and a description of the computational logic will follow in the next chapter.

CHAPTER IV

MODEL IMPLEMENTATION

4.1 INTRODUCTION

This chapter will discuss the implementation of the Interleaving Warehouse Layout Model. A description of the computational logic used in the computer program will be expanded upon. Emphasis will be given to the heuristic solution procedure of the model and its computational logic. A variation of the original algorithm to reduce computational time will be discussed. The chapter will close 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 is designed to include four major modules, the main program and three major subroutines. Flowcharts of the general logic used in the program and be found in Figures 4.1, 4.2, 4.3, and 4.4. We will describe three of the major modules in this section.

The main program calculates all the warehouse and product information that remains constant throughout the program. The product information includes the EOQ reorder quantities for each item (base case) and the percentage of total demand for each item, used in the Modified Location ABC Curve calculations. The warehouse information includes the cost of travel to all storage and retrieve locations, used in the total cost calculations, and the volume of space in each Preference Classification, used in the Modified Location ABC Curve analysis.

The first major step in the algorithm is to perform the item

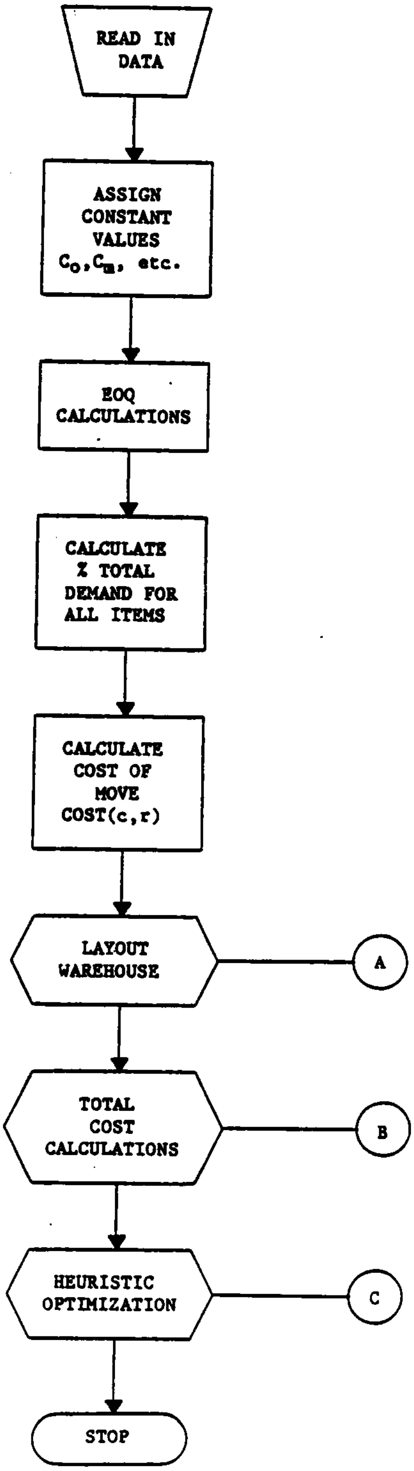

Figure 4.1. Flowchart of General Logic for the Interleaving Warehouse Layout Model.

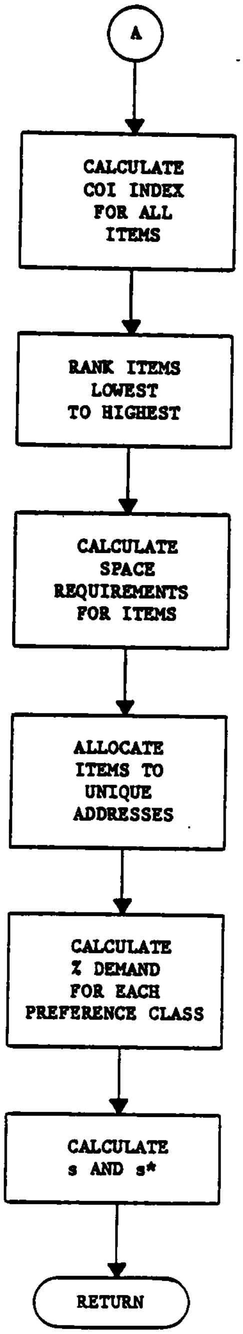

Figure 4.2. Flowchart of General Logic for the LAYOUT Routine.

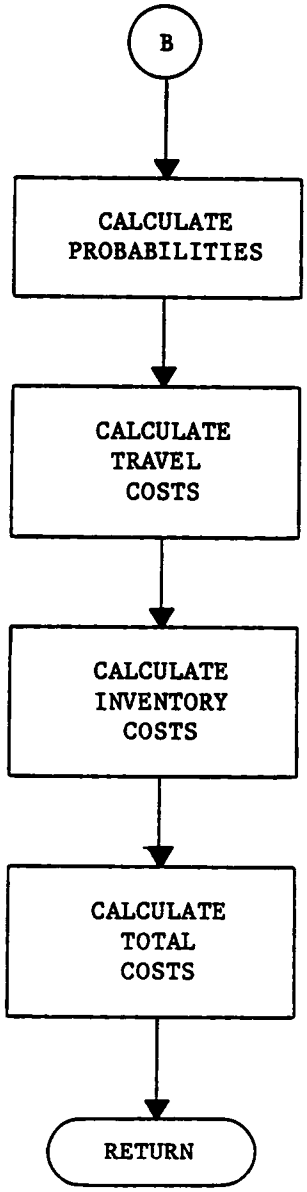

Figure 4.3. Flowchart of General Logic for the TCOST Routine.

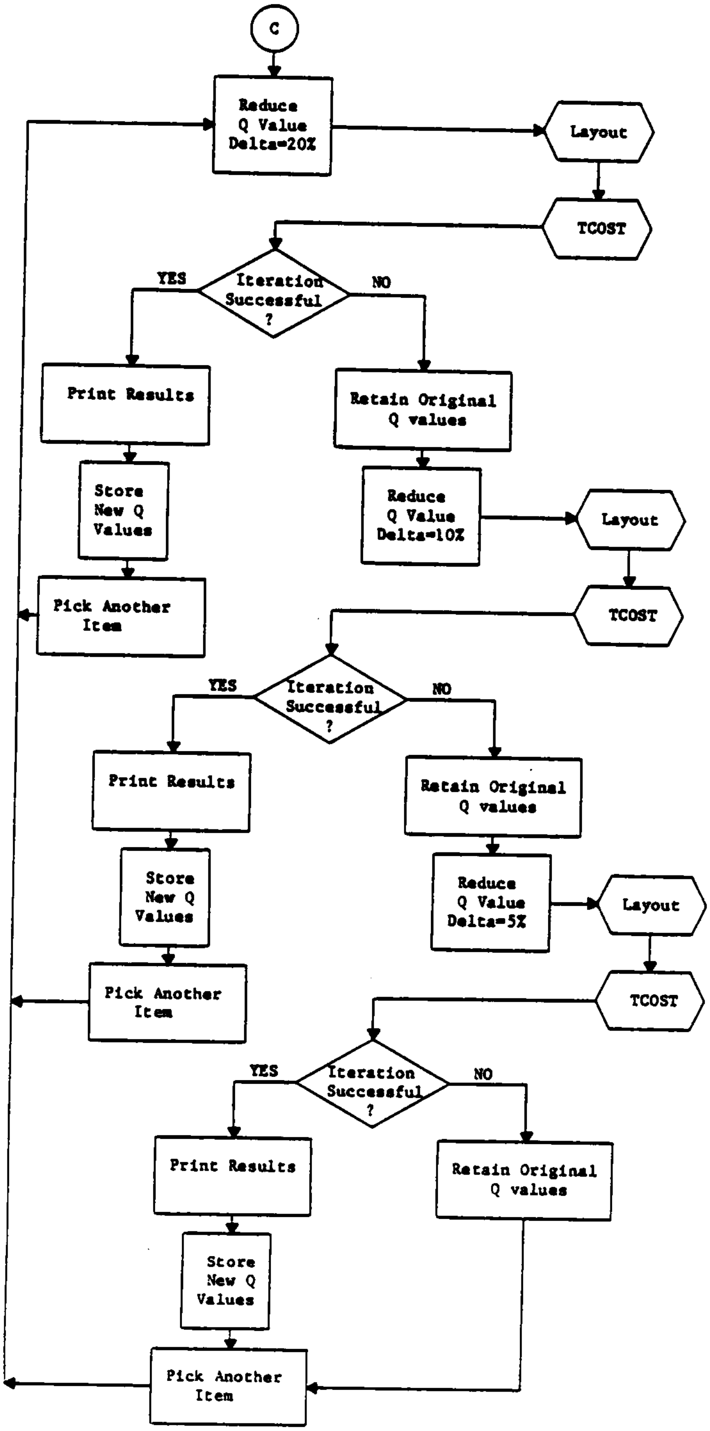

Figure 4.4. Flowchart of General Logic for the OPT Routine.

allocation procedure of the EOQ reorder quantities. This is performed in the subroutine LAYOUT.

The COI values for each item are calculated and items are ranked ordered from lowest to highest. The cumulative percentage of total demand is then calculated, as are the space requirements for each item. Both are used in the construction of the Modified Location ABC Curve. The program then allocates all items to Unique Address Classifications, starting with the most preferred address and the item having the lowest COI value, to insure maximum throughput. Once the allocation procedure has been completed, the total cumulative percentage demand for each Preference Classification may be calculated. This is the last piece of information needed to construct the Modified Location ABC Curve. We are now able to calculate the s* value, which is the fit parameter of the Modified Location ABC Curve, used in the probability calculation of a particular retrieve location being accessed for the present fixed layout.

The second major step in the algorithm is the total cost calculation performed in the subroutine TCOST. The probability of a Unique Address being the retrieve location is then calculated for all Unique Addresses in the warehouse. These probabilities are normalized to insure their summation equals one. This is needed due to the fact we are using a MIL order picking method. Travel costs including storage, interleave and retrieve travel, are calculated for individual items and the total warehouse. The inventory costs of individual items and the entire warehouse are calculated and added to the travel costs to complete the total cost calculations.

We are now ready to apply the optimization procedure discussed in the following section.

The objective function, along with its associated constraints, represents a nonconvex nonlinear programming problem appearing to defy closed form solution. Further, we would expect the problem to be of high dimensionality and sparse for most realistic problems. This is due to the fact that the variable of interest, X_{crj}, is a very high dimensional variable, where (c,r) refers to a unique address location and j refers to the item number. For example, a 25 item problem with 100 unique address locations results in a 2500 variable optimization problem. Most of these values are initially zero, but during the optimization process a particular value of zero could increase to some positive number. This is due to the fact that a reduction in reorder quantity of a particular item could lead to a change in the COI ordering. When a change in COI order occurs, a particular X_{crj} value, (which had a value of zero), increases to a positive number by having the lower ranked item move back in order and possibly to a less preferred location.

It is known, for any fixed set of reorder quantities Q_j, the COI rule allocates stock optimally. What is needed, is a way to proceed from one set of reorder quantities to the next in such a way that the total cost is reduced.

A gradient search technique is another possible way to solve the problem. The primary requirement, in order to use a gradient search technique, is that the first derivative of the objective function, with respect to the maximum quantity on hand, be calculable. We are unable to obtain this derivative due to the fact that this is a dynamic objective function changing with each allocation of the warehouse. Thus, we chose a heuristic solution procedure to solve the interleaving warehouse layout model.

4.3 SOLVING THE MODEL

The total cost (objective) function, as previously introduced, may be written as;

$$\sum_C \sum_R [(\sum_{j=1}^{n} (X_{crj}/\sum_C \sum_R X_{crj}) * OD_j) * COST_{cr}$$

$$+ [\sum_Z \sum_Y \{(|c-z|+|r-y|) * (AW + LS) * DOL + COST_{zy}\} * (\frac{i'^{s*}_{zy} - i''^{s*}_{zy}}{NUM_{zy}})$$

$$* (\sum_{j=1}^{n} (X_{crj}/\sum_C \sum_R X_{crj}) * OD_j)]]$$

$$+ \sum_{j=1}^{n} [Co * OD_j * AOS_j/(\sum_C \sum_R X_{crj}) + Cm * V_j * \sum_C \sum_R X_{crj}/2].$$

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

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

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

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

This requires the number of units of item j, assigned to location (c,r), for all locations and all items in the warehouse, is non-negative. Note the constraint;

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

which states no shortages are allowed, has been incorporated into the objective function. As a result, this constraint can be deleted from the problem.

4.3.1 The Heuristic Solution Procedure

A heuristic solution procedure is used to solve the interleaving warehouse layout problem. Starting with the item having the lowest COI value, a three step reduction procedure is initiated. First a 20% reduction in Q_j is attempted. If the total cost of the resulting warehouse layout is not reduced by 1/10 of 1% of the previous total cost, a reduction of 10% in Q_j is attempted, and then a reduction of 5% in Q_j. If all reductions for the first item fail, the item having the next higher COI value is tried. This procedure continues until all items have been tried. Program termination occurs when all items have failed to reduce the total cost in the three step reduction procedure (i.e., 3 * number of items, consecutive failures).

After each reduction, a new COI value for all items is calculated, and the warehouse layout procedure is reinitiated. This is due to the fact a reduction in reorder quantity of an item could possibly change its COI value and, as previously stated, the COI rule allocates stock optimally. Stock location also effects the fit parameter (s*) obtained from the Modified Location ABC Curve, which is used in the approximation of the probability of a retrieve location being accessed. By reallocating all items after each successful reduction we are able to obtain an accurate estimate of the cost for a given warehouse layout.

The variable of interst is X_{crj} where (c,r) refers to a unique address location and j refers to the item number. Thus the number of variables will equal the number of unique addresses times the number of items. Since the optimization procedure is a heuristic, sequential search procedure utilizing reductions of lot sizes the following relationship will always be true $\sum_C \sum_R X_{crj} < EOQ_j$. Further, this relationship holds as a strict equality for the base case. It is obvious some X_{crj} variables will equal zero. However, it is also possible that a X_{crj} value might increase, due to the fact a reduction in reorder quantities could effect an item's COI value, and its proximity to the input/output point. For example if item #1 is closer to the input/output point than item #2 ($COI_1 < COI_2$) a reduction in item #2 reorder quantity could possibly reduce its COI value enough to move it closer to the input/output point than item #1 ($COI_2 < COI_1$). Thus, the X_{crj} value which was zero could increase to some positive value.

If a particular unique address (c,r) has a zero value for all j ($\sum_j X_{crj} = 0$) it will always remain zero. This is due to the fact a reduction in reorder quantities is taking place and items are being moved closer to the input/output point.

4.3.2 Description of the Computational Logic Used in the Heuristic Solution Procedure

The heuristic solution procedure is performed in the subroutine OPT. Reorder quantities of all items are stored in computer memory. The reorder quantity of the item having the lowest COI value is chosen to be reduced by DELTA (20%). A new COI value for all items is calculated and ranked lowest to highest. The allocation procedure is repeated, starting with the lowest COI valued occupying the most preferred space, and continuing until all items are allocated. A new s* value is calculated from each new warehouse layout, as well as a new total cost value. If the new total cost value is at least 1/10 of 1% less than the previous total cost value, the new reorder quantities are accepted and the reorder quantity of the item having the next lowest COI value is reduced by DELTA = 20%. If the total cost is not sufficiently reduced, DELTA is reduced to 10% and the optimization process is repeated. If the third step (DELTA = 5%) in the optimization process does not reduce total cost sufficiently, the reorder quantity of that item remains unchanged and the reorder quantity of the item having the next lowest COI value is reduced by DELTA = 20%. Program completion is obtained when all items fail the three step optimization procedure.

A variation of the algorithm aimed at reducing the high computational overhead associated with the reestimation of the s* parameter called the Modified Warehouse Procedure will be introduced here. Instead of reducing reorder quantities one item at a time the Modified Warehouse Procedure will reduce a group of item reorder quantities simultaneously. This reduces the number of times the warehouse is reallocated and the number of s* calculations. All other facets of the original algorithm remain the same. The group size may be easily specified by the user and is analyzed in the following chapter with sizes of five and ten in a group.

The solutions generated by the Modified Warehouse Procedure may not be as desirable as the original algorithm due to the fact some individual items will be forced to be reduced when their reduction actually contributes to an increase in total cost. This individual increase in total cost will be compensated for by the contributions of the other members of the group to a reduction in total cost.

4.4 SUMMARY AND CONCLUSIONS

This section concludes a detailed description of the warehouse layout model implementation. The model generates a more cost effective reorder quantity and item location for all items. Total cost is broken down into two major components, travel costs and inventory costs, for each individual item.

The COI and EOQ models both assume continuous variables, as does the marginal analysis employed in deriving solutions. Representation of discrete quantities with continuous variables is taken only in the decision variables; values of all other parameters are exact and feasible. The original solution must be rounded to integer values, which increases the actual cost somewhat over the output of the model. The total cost of the solution should not be very sensitive to rounding unless reorder quantities are quite small.

The computational experience gained by the implementation of the Interleaving Warehouse Layout Model on a selected set of example problems will be discussed in detail in the following chapter.

CHAPTER V

COMPUTATIONAL EXPERIENCE

5.1 INTRODUCTION

This chapter discusses the results of the Interleaving Warehouse Layout Model and the Modified Warehouse Procedure. Additional information on input parameters and how they were selected are discussed, and a complete review of test problems is given. Comparisons between various test problems and model types are made. Emphasis is given to the analysis and interpretation of the results obtained from the models. The chapter closes with a discussion and summary of the conclusions drawn from the computational.studies.

5.2 PROFILE OF INPUT PARAMETERS

Realistic, hypothetical problems were generated and employed to test the algorithm and to study the nature of the solutions.

Recall the variable of interest is X_{crj}, where (c,r) refers to a unique address location and j refers to the item number. Thus, the number of unknown variables will equal the number of unique addresses times the number of items. Two sizes of test problems were chosen, 25 items with 72 locations representing a problem with 1800 unknown variables, and 50 items with 144 locations representing a problem with 7200 unknown variables.

Item parameter ranges were chosen to offer a wide variety of product size, product value, order size, and order frequency. Individual product data was generated randomly using a uniform distribution within the following ranges. These ranges were extracted from previous studies reported by Wilson (18).

Product size:	CU_j	(ft^3/unit);	1.0-11.0
Average order size:	AOS_j	(items/order);	1-30
Order frequency:	OD_j	(orders/week);	7-80
Product value:	V_j	(dollars/unit);	1.00-20.00

Problems were also studied where order frequency (OD_j) was generated randomly using a normal distribution with a mean of 37.5 and a standard deviation of 26.58. This is done to see how different distributions of demand effect the solution.

The warehouse layout was chosen to insure warehouse capacity was greater than the capacity needed to store the EOQ reorder quantities of all products. For the 25 item problems, warehouse capacity approximated 55,123 ft^3 and for the 50 item problems, warehouse capacity approximated 110,246 ft^3 .

Inventory carrying costs and reorder costs were chosen to be realistic, feasible and consistent with respect to individual product characteristics and are respectively valued at $0.006 per week and $5.00 per reorder. These are similar to the values reported by Wilson (18).

Kallina and Lynn (10) state that order picking costs typically dominate stock setting costs, and we have chosen a dollar value for travel distance per foot (DOL) of $0.003, to be consistent with this relationship.

Storage volume of all unique address locations was chosen to be 765.6 ft^3. This is a realistic representation of a warehouse in the fact that approximately fifty 48" x 46" x 24" pallets may be stored in a single location. The aisle width, being ten feet, is large enough to allow many types of fork lift trucks to turn around, thus allowing travel distances free of backtracking by the order picking vehicle.

5.3 RESULTS FROM THE 25 ITEM PROBLEMS

This section will discuss the results of test problems having 25 items and warehouse storage capacity of approximately 55,123 cubic feet. This group of test problems can further be divided into three subgroups. One group consists of two test problems with all item characteristics having a uniform distribution between the ranges previously discussed. These are referred to as TP1 and TP2. Two other test problems referred to as TP3 and TP4, feature a normal distribution for reorder characteristics, keeping all other item characteristics the same as the previously mentioned problems. The final four problems in this group use the Modified Warehouse Procedure (referred to as MW1,MW2,MW3,MW4; reducing five items simultaneously corresponding to problems TP1,TP2,TP3,TP4, respectively) to test the solutions of the previously discussed four problems.

5.3.1 Comparison of Two Uniformly Distributed Problems

This section compares two uniformly distributed problems reduced one item at a time. The data for these problems can be found in Tables 5.3.1 and 5.3.2, while the results are shown in Tables 5.3.3 and 5.3.4.

Both problems produce inventory costs, evaluated at the base case, accounting for approximately 20% of total cost, while travel costs account for the remaining 80%. A total cost reduction of approximately 27% was realized for each of the problems. The inventory costs increased by approximately 16% to account for 36% of the final total cost, while travel costs decreased by 16% to represent 64% of the final

Table 5.3.1. Input Data for TP1

ITEM	CU	AOS	OD	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

Table 5.3.2. Input Data for TP2

ITEM	CU	AOS	OD	V
1	1.03	26	8	59.00
2	6.90	12	17	5.09
3	5.71	23	78	14.72
4	7.46	24	74	5.16
5	1.06	24	11	16.96
6	10.33	20	60	6.85
7	10.92	15	68	19.28
8	2.55	6	18	7.30
9	2.93	10	46	11.23
10	7.10	9	40	6.40
11	1.33	20	76	15.32
12	9.07	3	67	15.94
13	9.60	12	49	13.00
14	8.88	8	66	10.10
15	9.78	25	29	6.29
16	2.46	13	34	3.57
17	1.81	21	29	16.50
18	5.00	20	46	19.40
19	4.53	23	37	5.71
20	6.10	9	27	2.23
21	10.47	30	57	3.94
22	6.50	2	52	12.10
23	9.31	3	62	13.67
24	1.59	22	29	2.83
25	1.96	11	43	16.09

Table 5.3.3. Results of TP1 and MW1.

Problem: TP1 Items: 25 Distribution: Uniform Group Size: 1

	Total Cost	Inventory	Travel
Base Case	2219.04	451.0451	1767.995
Final Solution	1624.15	594.1342	1030.016
Change in percentage	-26.81	+31.72	-41.74
Base case % of Total Cost		20.33	79.67
Final Solution % of Total Cost		36.58	63.42

Number of successful iterations = 75
Base Case Warehouse Utilization = 94.4%
Final Solution Warehouse Utilization = 55.25%

Problem: MW1 Items: 25 Distribution: Uniform Group Size: 5

	Total Cost	Inventory	Travel
Base Case	2219.04	451.0451	1767.995
Final Solution	1636.786	668.0644	968.7221
Change in percentage	-26.24	+48.11	-45.21
Base case % of Total Cost		20.33	79.67
Final Solution % of Total Cost		40.82	59.18

Number of successful iterations = 20
Base Case Warehouse Utilization = 94.4%
Final Solution Warehouse Utilization = 43.02%

Table 5.3.4. Results of TP2 and MW2.

Problem: TP2 Items: 25 Distribution: Uniform Group Size: 1			
	Total Cost	Inventory	Travel
Base Case	2297.407	472.1208	1825.286
Final Solution	1668.745	595.6608	1073.084
Change in percentage	-27.36	+26.17	-41.21
Base case % of Total Cost		20.55	79.45
Final Solution % of Total Cost		35.70	64.30

Number of successful iterations = 71
Base Case Warehouse Utilization = 96.53%
Final Solution Warehouse Utilization = 57.54%

Problem: MW2 Items: 25 Distribution: Uniform Group Size: 5			
	Total Cost	Inventory	Travel
Base Case	2297.407	472.1208	1825.286
Final Solution	1661.388	699.0329	962.3559
Change in percentage	-27.68	+48.06	-47.28
Base case % of Total Cost		20.55	79.45
Final Solution % of Total Cost		42.08	57.92

Number of successful iterations = 21
Base Case Warehouse Utilization = 96.53%
Final Solution Warehouse Utilization = 45.71%

total cost. Warehouse capacity was reduced by approximately 39% in each case. TP1 required 75 successful iterations while TP2 required 71. The COI order of items in each problem did not change.

5.3.2 Analysis of the Modified Warehouse Procedure Applied to a Uniformly Distributed Problem

This section discusses the results of the Modified Warehouse Procedure (MW1 & MW2; reducing five items simultaneously) applied to the two previously discussed uniformly distributed problems. The results of these two problems are shown in Tables 5.3.3 and 5.3.4. The cost curves for these problems are shown in Figures 5.3.1 and 5.3.2.

The results show the final solution, using the Modified Warehouse Procedure applied to the first test problem (MW1) was not quite as good as TP1 being higher by approximately 0.78%. The inventory contribution to the final total cost increased to approximately 41%, while travel cost decreased to 59%. A change of 4% in each case. Warehouse capacity was reduced by 12% which accounts for the reduction in travel costs. The number of successful reductions dropped to 20 while CPU time for MW1 was approximately 17% of the CPU time for TP1.

A similar comparison of TP2 and MW2 shows that total cost of MW2 is lower by 0.4% of the final total cost of TP2. Inventory cost increased by approximately 6%, to account for 42% of the final total cost, while travel cost decreased to 58% of total cost. Warehouse capacity is reduced by 11.8% over TP2. The number of successful reductions in item lot sizes dropped to 21 while CPU time was just 25% of the CPU time for TP2.

Overall, the Modified Warehouse Procedure tested very well against

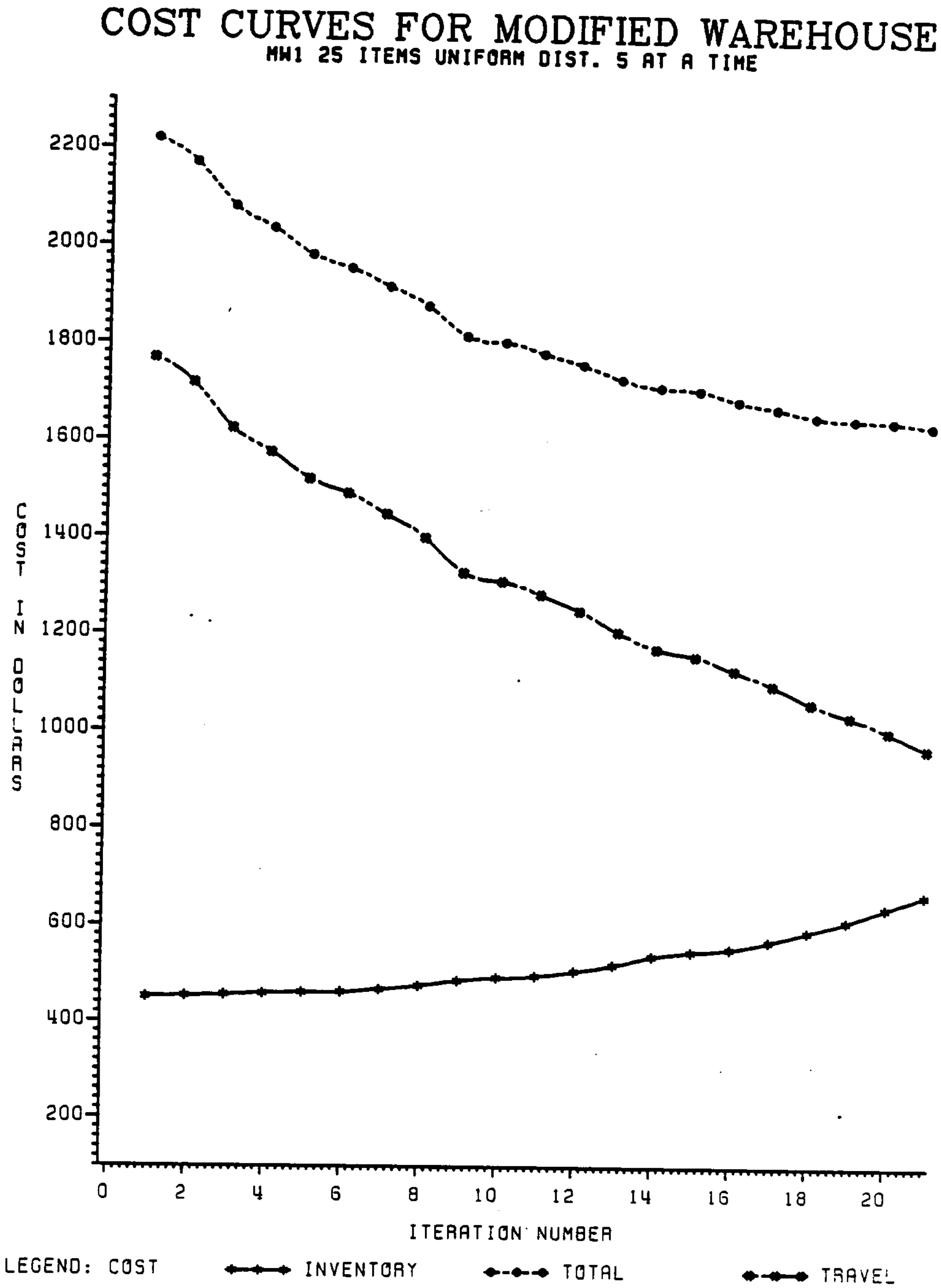

Figure 5.3.1. Cost Curves for Modified Warehouse: MW1.

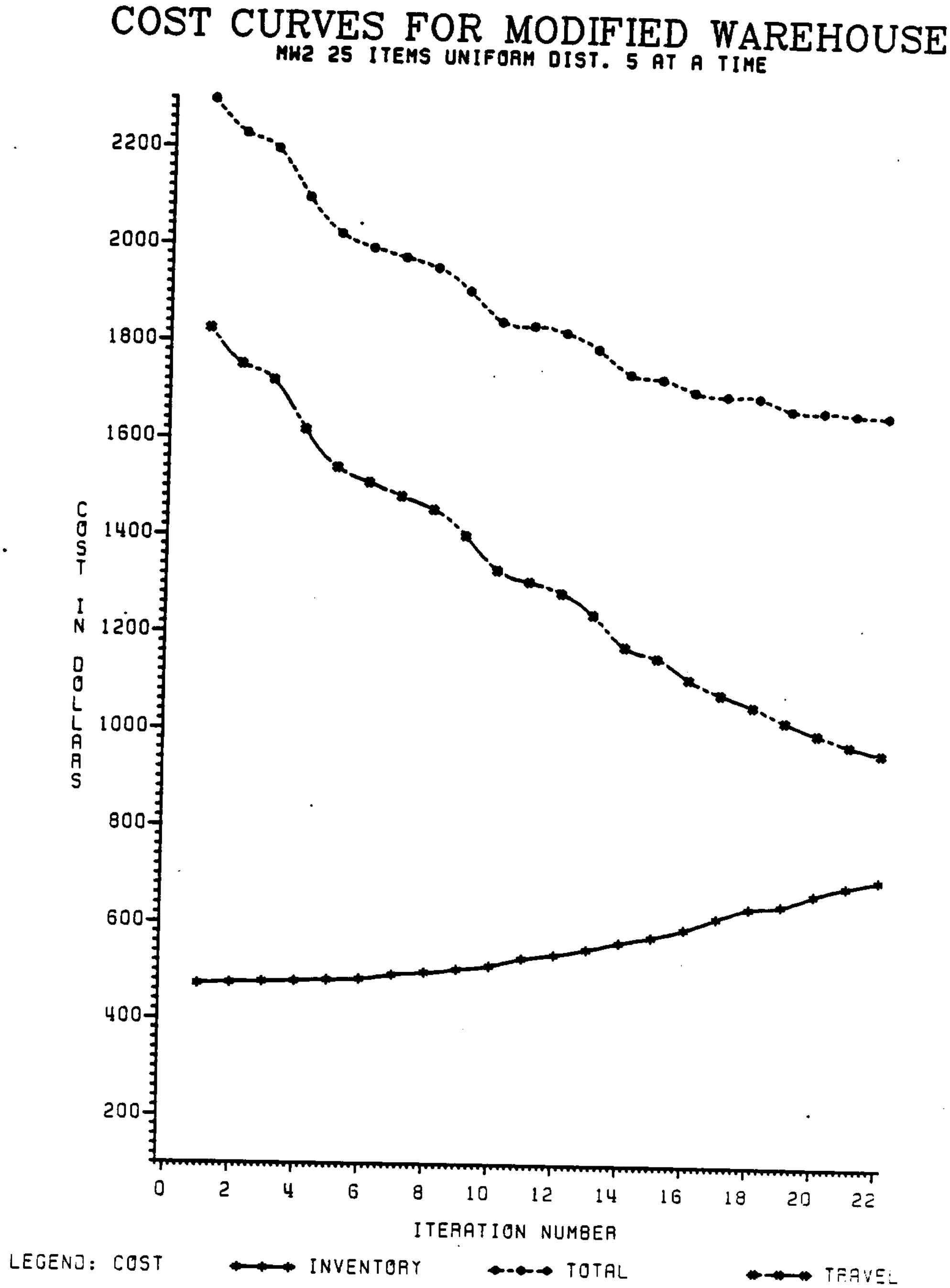

Figure 5.3.2. Cost Curves for Modified Warehouse: MW2.

the single item reduction algorithm using uniform distributions. Both tests being within ± 1% of the original algorithm, while reducing CPU time by a minimum of 75%.

The Modified Warehouse Procedure reduces the five lowest ranked COI items in the first optimization, then proceeds to simultaneously reduce the next five lowest rank COI items, and continues in this manner until the final solution is reached. It is possible a reduction in reorder quantity of a particular item, in the group of five items simultaneously reduced, contributes to an increase in total cost. This could be compensated for by a larger contribution, of the remaining items in the group, to a reduction in total cost, which allows for a successful iteration. The reduction of reorder quantities actually reduces warehouse utilization and travel costs by moving all items closer to the input/output point. Therefore, a particular item's contribution to an increase in total cost will be represented by a greater increase in inventory costs than decrease in travel cost. This is the primary reason, when using the Modified Warehouse Procedure, inventory costs are higher and travel costs are reduced, compared to the original algorithm. It must be remembered the original algorithm reduces items one at a time and only those items which contribute to a reduction in total cost are reduced.

5.3.3 Comparison of Normally Distributed Demand Characteristics to Distributed Demand Characteristics

This section discusses a comparison of two problems with normally distributed demand characteristics (TP3 & TP4) to two identical problems using a uniformly distributed demand characteristic (TP1 &

TP2). The data used in TP3 and TP4 can be found in Tables 5.3.5 and 5.3.6, respectively. The results can be seen in Tables 5.3.7 and 5.3.8.

Solution quality remained the same for both comparisons at approximately a 27% reduction from the base case. As before for the base case solutions, inventory cost provided 21% of total cost and travel cost represented the remaining 79%. There was no appreciable change in the composition of the final total cost where inventory cost accounted for approximately 36% and travel costs the remaining 64%. Warehouse capacity showed no noticable change being reduced by approximately 39% in each case. Again, as with the uniformly distributed problems, the COI ordering did not change from the base case order to the final solution order. Overall, there was not a noticable difference in solution quality or make up between the uniformly and normally distributed demand characteristics problems.

5.3.4 Analysis of the Modified Warehouse Procedure Applied to a Normally Distributed Demand Characteristic

This section applies and compares the Modified Warehouse Procedure to the two normally distributed test problems (MW3 & MW4). Results from these two tests are located in Tables 5.3.7 and 5.3.8. Cost curves can be seen in Figures 5.3.3 and 5.3.4.

Comparing the original algorithm (TP3) with the Modified Warehouse Procedure (MW3) resulted in the total cost of MW3 being approximately 3.55% higher than TP3. The make up of the final total cost was almost identical for both problems where inventory cost was approximately 37% and travel cost accounting for the remaining 63%. Warehouse capacity

Table 5.3.5. Input Data for TP3

ITEM	CU	AOS	OD	V
1	5.12	14	28	7.57
2	3.05	24	73	6.22
3	2.45	27	20	8.73
4	3.46	4	17	18.71
5	9.28	10	8	10.69
6	7.20	12	46	1.24
7	5.34	22	4	13.32
8	6.15	27	44	2.99
9	10.18	21	77	4.48
10	6.34	18	12	16.40
11	3.96	1	66	7.06
12	1.68	20	38	17.63
13	5.78	1	76	6.63
14	4.46	4	55	8.60
15	4.65	17	56	6.47
16	5.08	20	38	3.05
17	5.05	19	30	10.77
18	3.63	27	27	10.37
19	4.81	15	38	17.04
20	5.00	11	83	19.30
21	4.15	15	60	7.59
22	5.19	28	22	19.80
23	3.56	6	4	12.41
24	8.07	8	60	3.56
25	10.11	22	44	8.51

Table 5.3.6. Input Data for TP4

ITEM	CU	AOS	OD	V
1	1.03	26	49	5.09
2	6.90	12	24	14.72
3	5.71	23	36	5.16
4	7.46	24	38	16.96
5	1.06	24	24	6.85
6	10.33	20	26	19.28
7	10.92	15	56	7.30
8	2.55	6	61	11.23
9	2.93	10	29	6.40
10	7.10	9	2	15.32
11	1.33	20	63	15.94
12	9.07	3	80	13.00
13	9.60	12	59	10.10
14	8.88	8	17	6.29
15	9.78	25	45	3.57
16	2.46	13	31	16.50
17	1.81	21	56	19.40
18	5.00	20	104	5.71
19	4.53	23	35	2.23
20	6.10	9	68	3.94
21	10.47	30	115	12.10
22	6.50	2	66	13.67
23	9.31	3	22	2.83
24	1.59	22	63	16.09
25	1.96	11	45	15.94

Table 5.3.7. Results of TP3 and MW3.

Problem: TP3 Items: 25 Distribution: Normal Group Size: 1

	Total Cost	Inventory	Travel
Base Case	1948.604	398.2907	1550.314
Final Solution	1396.996	525.4711	871.5249
Change in percentage	-28.31	+31.93	-43.78
Base case % of Total Cost		20.44	79.56
Final Solution % of Total Cost		37.61	62.39

Number of successful iterations = 76
Base Case Warehouse Utilization = 93.00%
Final Solution Warehouse Utilization = 49.79%

Problem: MW3 Items: 25 Distribution: Normal Group Size: 5

	Total Cost	Inventory	Travel
Base Case	1948.604	398.2907	1550.314
Final Solution	1446.582	540.6962	905.8857
Change in percentage	-25.76	+35.75	-41.57
Base case % of Total Cost		20.44	79.56
Final Solution % of Total Cost		37.38	62.62

Number of successful iterations = 18
Base Case Warehouse Utilization = 93.00%
Final Solution Warehouse Utilization = 45.72%

Table 5.3.8. Results of TP4 and MW4.

Problem: TP4 Items: 25 Distribution: Normal Group Size: 1

	Total Cost	Inventory	Travel
Base Case	2292.339	491.5725	1800.766
Final Solution	1707.731	616.3415	1091.389
Change in percentage	-25.50	+25.38	-39.39
Base case % of Total Cost		21.44	78.56
Final Solution % of Total Cost		36.09	63.91

Number of successful iterations = 67
Base Case Warehouse Utilization = 95.82%
Final Solution Warehouse Utilization = 58.45%

Problem: MW4 Items: 25 Distribution: Normal Group Size: 5

	Total Cost	Inventory	Travel
Base Case	2292.339	491.5725	1800.766
Final Solution	1680.617	721.8166	958.8012
Change in percentage	-26.69	+46.84	-46.76
Base case % of Total Cost		21.44	78.56
Final Solution % of Total Cost		42.95	57.05

Number of successful iterations = 19
Base Case Warehouse Utilization = 95.82%
Final Solution Warehouse Utilization = 49.14%

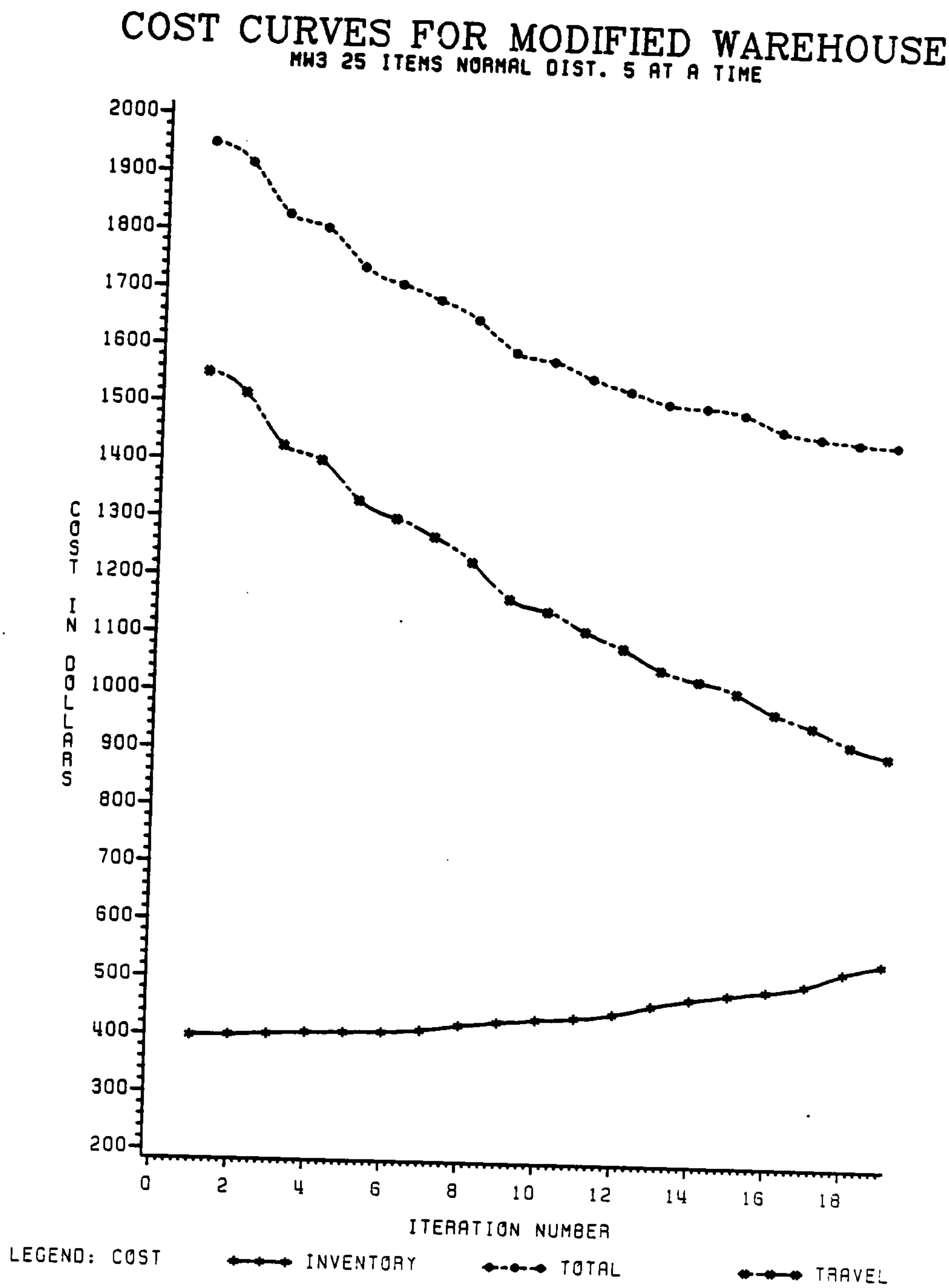

Figure 5.3.3. Cost Curves for Modified Warehouse: MW3.

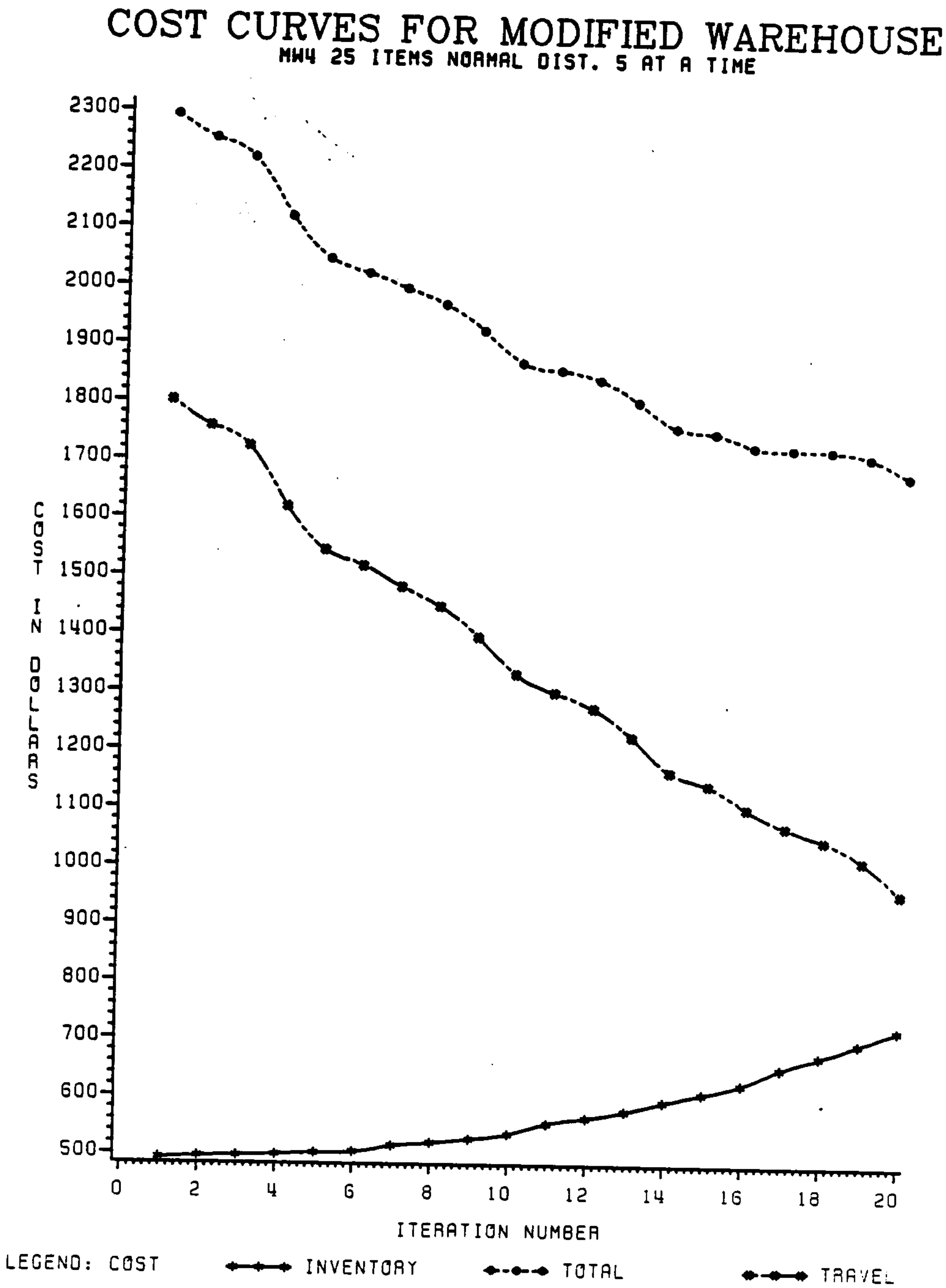

Figure 5.3.4. Cost Curves for Modified Warehouse: MW4.

had only a slight change with MW3 being lower by approximately 4%. CPU time for MW3 was approximately 12% of the CPU time needed for TP3, while the number of successful iterations was reduced 24% to 18.

Comparing TP4 and MW4, the final solution of MW4 was lower by 1.59%. Inventory costs represented 43% of the final total cost for MW4 and 36% of TP4, and travel costs represented 57% and 64% respectively. Warehouse capacity was 9% less for MW4 than for TP4 which could account for the lower travel costs. CPU time for MW4 was 78% less than CPU time for TP4, while the number of successful iterations was 28% of that for TP4.

Overall, for the 25 item problem normally distributed demand (OD_j) shows little difference to a uniformly distributed demand, all other factors being constant. Also, the Modified Warehouse Procedure has proven to be successful in saving computational time while retaining a high quality solution. Comparing the Modified Warehouse solutions with the original algorithm (one item at a time reduction) the worst case found was a solution 3.55% higher than the original and a solution 1.59% lower. Both worst case scenarios occurred when the problem possessed a normally distributed demand characteristic. The reason for this could relate to how a normally distributed demand characteristic effects inventory cost versus uniformly distributed demand characteristics.

5.4 RESULTS FROM THE FIFTY ITEM PROBLEMS

We will now expand the scope of the test problems to include 7200 unknown variables with a warehouse capacity of approximately 110,000 ft^3. Again, we will use test problems of uniformly (TP5) and normally distributed (TP6) demand characteristics. Also, we have reduced the number of items the Modified Warehouse Procedure simultaneously reduces in one iteration to five (MW5A & MW6A), or 10% of the total number of items. We will compare this to a Modified Warehouse Procedure reducing ten items simultaneously in one iteration (MW5B & MW6B).

5.4.1 Comparison of Uniformly Distributed Demand Characteristics to Normally Distributed Demand Characteristics

This section will compare the uniformly distributed demand problem (TP5) to the normally distributed demand problem (TP6) all other item characteristics being equal. The data for these problems can be found in Tables 5.4.1 and 5.4.2, while results are shown in Tables 5.4.3. and 5.4.4.

Again, there are no significant differences in solution quality between the uniformly distributed and normally distributed demand characteristic problems. Problem TP5 recorded a 28.99% reduction in total cost while TP6 reduced total cost by 27.76%. Inventory costs measured at the base case for both problems were within 1% of each other at 16.23% and 17.03% respectively. This is also true for the travel costs being 69.43% and 68.83% respectively. In the final solutions to both problems inventory costs were 30.57% and 31.70% respectively, with travel costs accounting for the remaining 69.43% and 68.83%. Warehouse capacity was reduced 35.40% by TP5 and 38.80% by TP6. Again, as was noticed in all the 25 item problems, the COI rank ordering does not change.

The COI order not changing for such a large problem is rather surprising. Taking a reduction of one item at a time and skipping

Table 5.4.1. Input Data for TP5

ITEM	CU	AOS	OD	V
1	4.02	2	54	6.49
2	4.14	2	20	14.25
3	5.54	6	65	13.10
4	6.08	19	23	3.57
5	7.60	28	18	16.18
6	2.94	3	75	4.72
7	5.81	20	26	8.74
8	7.87	13	12	17.96
9	9.66	26	27	10.66
10	2.34	26	56	15.42
11	9.22	29	66	8.98
12	1.84	20	55	6.26
13	3.16	9	18	16.43
14	1.78	18	78	17.87
15	4.88	6	50	12.42
16	8.82	25	47	14.30
17	8.12	17	35	5.96
18	4.66	16	68	16.95
19	5.87	27	44	8.86
20	9.00	11	8	1.56
21	1.76	14	78	6.50
22	8.90	1	53	14.88
23	1.67	5	64	10.27
24	2.07	7	79	9.91
25	8.62	17	66	9.07
26	5.25	6	40	18.81
27	1.89	26	15	4.86
28	1.95	17	70	19.13
29	6.83	21	77	7.02
30	7.94	26	45	11.43
31	5.61	12	33	6.51
32	10.13	20	76	9.52
33	9.26	25	25	2.94
34	3.98	28	47	13.37
35	5.06	29	55	7.11
36	3.42	29	13	5.24
37	3.86	24	61	2.53
38	7.59	1	33	5.13
39	4.70	4	62	12.41
40	4.43	20	44	9.76
41	10.51	8	58	9.01
42	10.79	6	39	5.23
43	6.31	18	66	11.95
44	7.44	1	48	18.26
45	8.17	26	13	11.72

Table 5.4.1. (Continued)

ITEM	CU	AOS	OD	V
46	9.78	29	35	17.80
47	1.51	26	52	13.38
48	9.08	20	75	15.03
49	6.03	17	25	3.45
50	4.33	25	47	1.08

Table 5.4.2. Input Data for TP6

ITEM	CU	AOS	OD	V
1	4.02	2	30	6.49
2	4.14	2	43	14.25
3	5.54	6	51	13.10
4	6.08	19	7	3.57
5	7.60	28	16	16.18
6	2.94	3	72	4.72
7	5.81	20	34	8.74
8	7.87	13	8	17.96
9	9.66	26	54	10.66
10	2.34	26	10	15.42
11	9.22	29	51	8.98
12	1.84	20	22	6.26
13	3.16	9	59	16.43
14	1.78	18	74	17.87
15	4.88	6	1	12.42
16	8.82	25	68	14.30
17	8.12	17	25	5.96
18	4.66	16	38	16.95
19	5.87	27	44	8.86
20	9.00	11	67	1.56
21	1.76	14	35	6.50
22	8.90	1	53	14.88
23	1.67	5	38	10.27
24	2.07	7	103	9.91
25	8.62	17	41	9.07
26	5.25	6	39	18.81
27	1.89	26	53	4.86
28	1.95	17	90	19.13
29	6.83	21	44	7.02
30	7.94	26	37	11.43
31	5.61	12	24	6.51
32	10.13	20	14	9.52
33	9.26	25	26	2.94
34	3.98	28	46	13.37
35	5.06	29	114	7.11
36	3.42	29	51	5.24
37	3.86	24	13	2.53
38	7.59	1	14	5.13
39	4.70	4	13	12.41
40	4.43	20	36	9.76
41	10.51	8	14	9.01
42	10.79	6	41	5.23
43	6.31	18	20	11.95
44	7.44	1	42	18.26
45	8.17	26	3	11.72

Table 5.4.2. (Continued)

ITEM	CU	AOS	OD	V
46	9.78	29	62	17.80
47	1.51	26	33	13.38
48	9.08	20	37	15.03
49	6.03	17	46	3.45
50	4.33	25	93	1.08

Table 5.4.3. Results of TP5, MW5A and MW5B.

Problem: TP5 Items: 50 Distribution: Uniform Group Size: 1

	Total Cost	Inventory	Travel
Base Case	5911.894	959.5856	4952.312
Final Solution	4198.738	1283.729	2915.011
Change in percentage	-28.99	+33.78	-41.14
Base case % of Total Cost		16.23	83.77
Final Solution % of Total Cost		30.57	69.43

Number of successful iterations = 143
Base Case Warehouse Utilization = 95.27%
Final Solution Warehouse Utilization = 59.80%

Problem: MW5A Items: 50 Distribution: Uniform Group Size: 5

	Total Cost	Inventory	Travel
Base Case	5911.894	959.5856	4952.312
Final Solution	4044.565	1554.375	2490.19
Change in percentage	-31.59	+61.98	-49.72
Base case % of Total Cost		16.23	83.77
Final Solution % of Total Cost		38.43	61.57

Number of successful iterations = 43
Base Case Warehouse Utilization = 95.27%
Final Solution Warehouse Utilization = 44.09%

Table 5.4.3. (Continued)

Problem: MW5B Items: 50 Distribution: Uniform Group Size: 10

	Total Cost	Inventory	Travel
Base Case	5911.894	959.5856	4952.312
Final Solution	4063.109	1605.262	2457.847
Change in percentage	-31.27	+67.29	-50.37
Base case % of Total Cost		16.23	83.77
Final Solution % of Total Cost		39.51	60.49

Number of successful iterations = 24
Base Case Warehouse Utilization = 95.27%
Final Solution Warehouse Utilization = 37.62%

Table 5.4.4. Results of TP6, MW6A and MW6B.

Problem: TP6 Items: 50 Distribution: Normal Group Size: 1

	Total Cost	Inventory	Travel
Base Case	5136.703	874.5397	4262.164
Final Solution	3710.604	1156.618	2553.985
Change in percentage	-27.76	+32.25	-40.08
Base case % of Total Cost		17.03	82.97
Final Solution % of Total Cost		31.17	68.83

Number of successful iterations = 135
Base Case Warehouse Utilization = 92.48%
Final Solution Warehouse Utilization = 53.66%

Problem: MW6A Items: 50 Distribution: Normal Group Size: 5

	Total Cost	Inventory	Travel
Base Case	5136.703	874.5397	4262.164
Final Solution	3653.663	1313.246	2340.416
Change in percentage	-28.87	+50.16	-45.09
Base case % of Total Cost		17.03	82.97
Final Solution % of Total Cost		35.94	64.06

Number of successful iterations = 36
Base Case Warehouse Utilization = 92.48%
Final Solution Warehouse Utilization = 51.86%

Table 5.4.4. (Continued)

Problem: MW6B Items: 50 Distribution: Normal Group Size: 10

	Total Cost	Inventory	Travel
Base Case	5136.703	874.5397	4262.164
Final Solution	3599.537	1422.212	2177.325
Change in percentage	-29.93	+62.62	-48.92
Base case % of Total Cost		17.03	82.97
Final Solution % of Total Cost		39.51	60.49

Number of successful iterations = 22
Base Case Warehouse Utilization = 92.48%
Final Solution Warehouse Utilization = 37.69%

items that do not reduce total cost would seem to be the perfect setting for a change in COI order. For example, consider two items that have close COI values with the values of the numerator and demoninator of the same order of magnitude. If the lower COI valued item is not reduced, while the higher COI valued item is reduced significantly, it would be expected that the higher COI valued item replace the lower. This may not be happening due to the fact our reduction is too small to effect the COI values, or there are no closely valued COI item pairs.

5.4.2 Analysis of Two Modified Warehouse Procedures Applied to a Uniformly Distributed Demand Problem

This section will compare the uniformly distributed demand problem previously discussed (TP5) with two Modified Warehouse Procedures, one reducing five items simultaneously (MW5A) and the other reducing ten items simultaneously (MW5B). Results of this comparison can be seen in Table 5.4.3. The cost curves for MW5A and MW5B are shown in Figures 5.4.1 and 5.4.2.

The comparison of TP5 and the two Modified Warehouse Procedures surprisingly reveals that MW5B (ten items reduced simultaneously) produces a closer solution to the original algorithm. Both final solutions being lower than the original final solution by 3.67% and 3.23% respectively. Intuitively, one would expect the procedure that reduces the fewest items simultaneously to have the closest solution to the original. This not being the case, could be due to the fact a reduction of an individual item could actually contribute to an increase in total cost. This increase in total cost could be

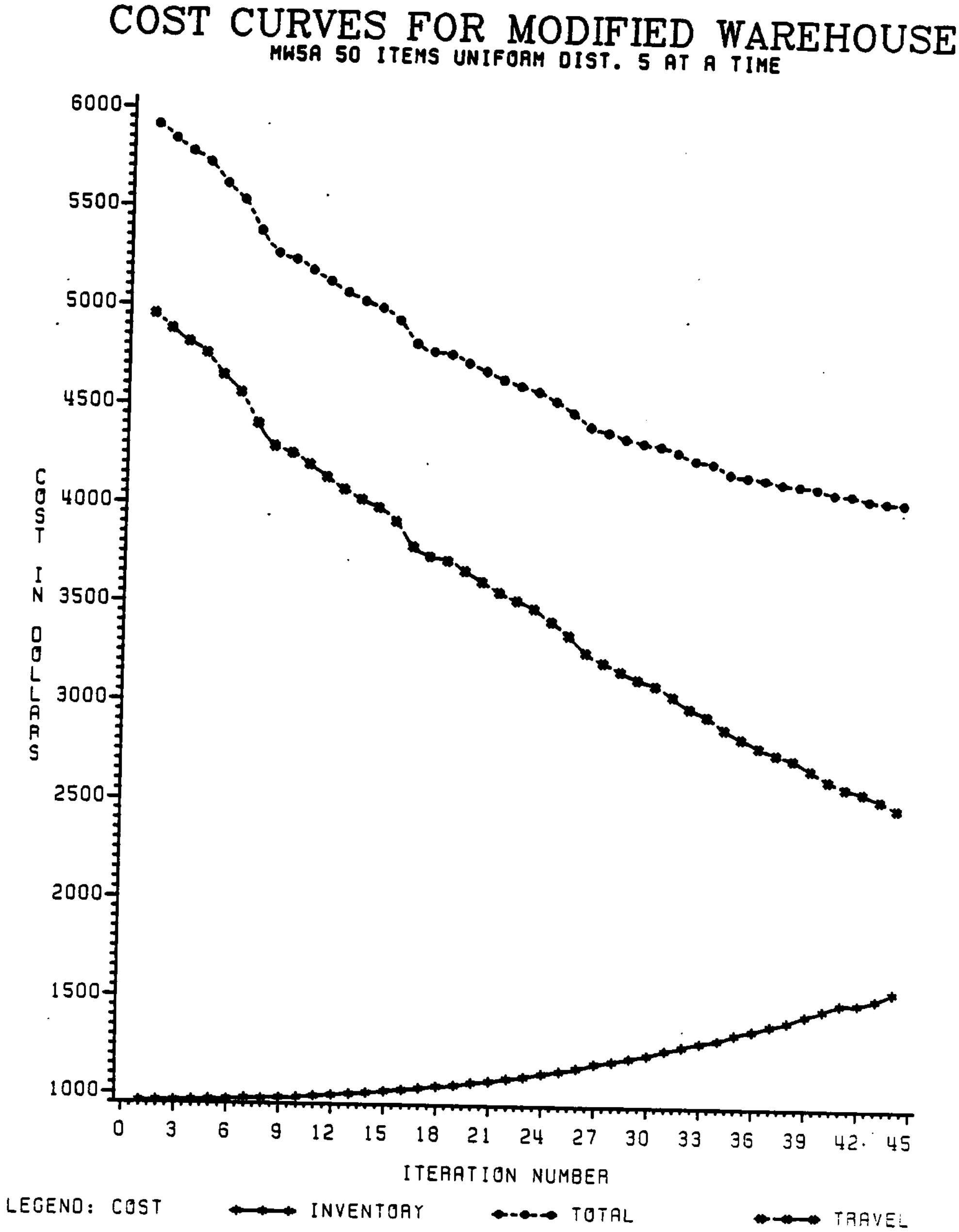

Figure 5.4.1. Cost Curves for Modified Warehouse: MW5A.

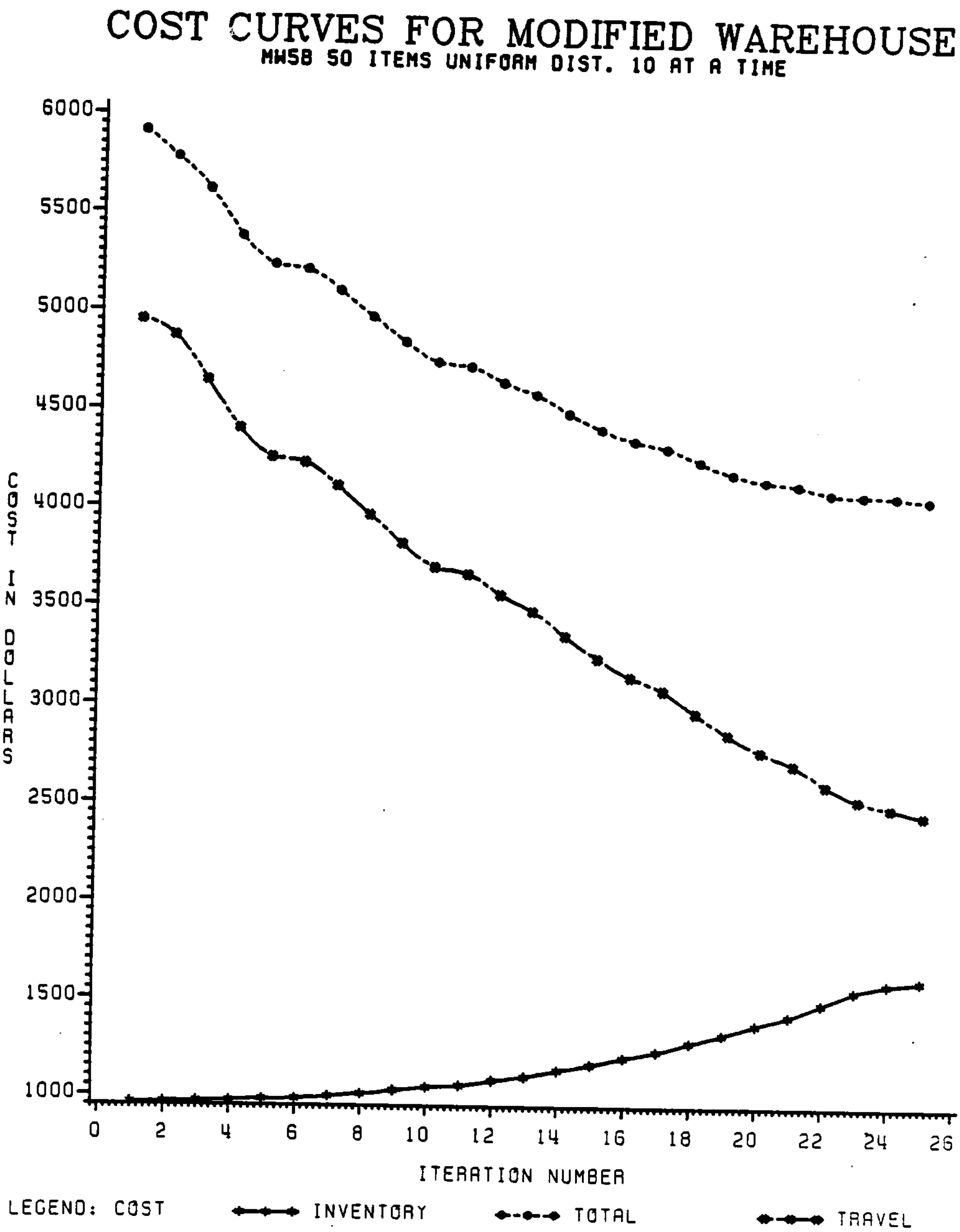

Figure 5.4.2. Cost Curves for Modified Warehouse: MW5B.

overshadowed by the reduced contribution of the other items in the group reduction. However, the solutions are within 0.5% of each other and do not deviate from the original algorithm by a great amount. As the number of items simultaneously reduced increases, the contribution of inventory cost to the total cost increases also, being 38.43% and 39.51% for a five item and ten item reduction, respectively. As previously discussed, this is due to the fact a reduction in reorder quantity increases inventory costs, which is compensated for by a greater reduction in travel costs.

For this larger problem, warehouse utilization is reduced by 51.2% and 57.7% for five item and ten item simultaneous reductions, where a single item reduction reduces warehouse utilization by 35%. This large capacity reduction is reflected in a large reduction in travel costs of 61.57% and 60.49% respectively. The number of successful iterations for MW5A was 43 which is approximately 33% of the original algorithm and 28 for MW5B or approximately 20% of the original. CPU time for MW5A was 18% of the original, while CPU time for MW5B was 12.5% of the original algorithm.

5.4.3 Analysis of Two Modified Warehouse Procedures Applied to a Normally Distributed Demand Problem

This section will compare two Modified Warehouse Procedures, one which reduces five items simultaneously (MW6A), the other reducing ten items simultaneously (MW6B), to the original algorithm applied to a normally distributed demand problem (TP6). The results for this comparison are summarized in Table 5.4.4. Cost curves are shown in Figures 5.4.3 and 5.4.4.

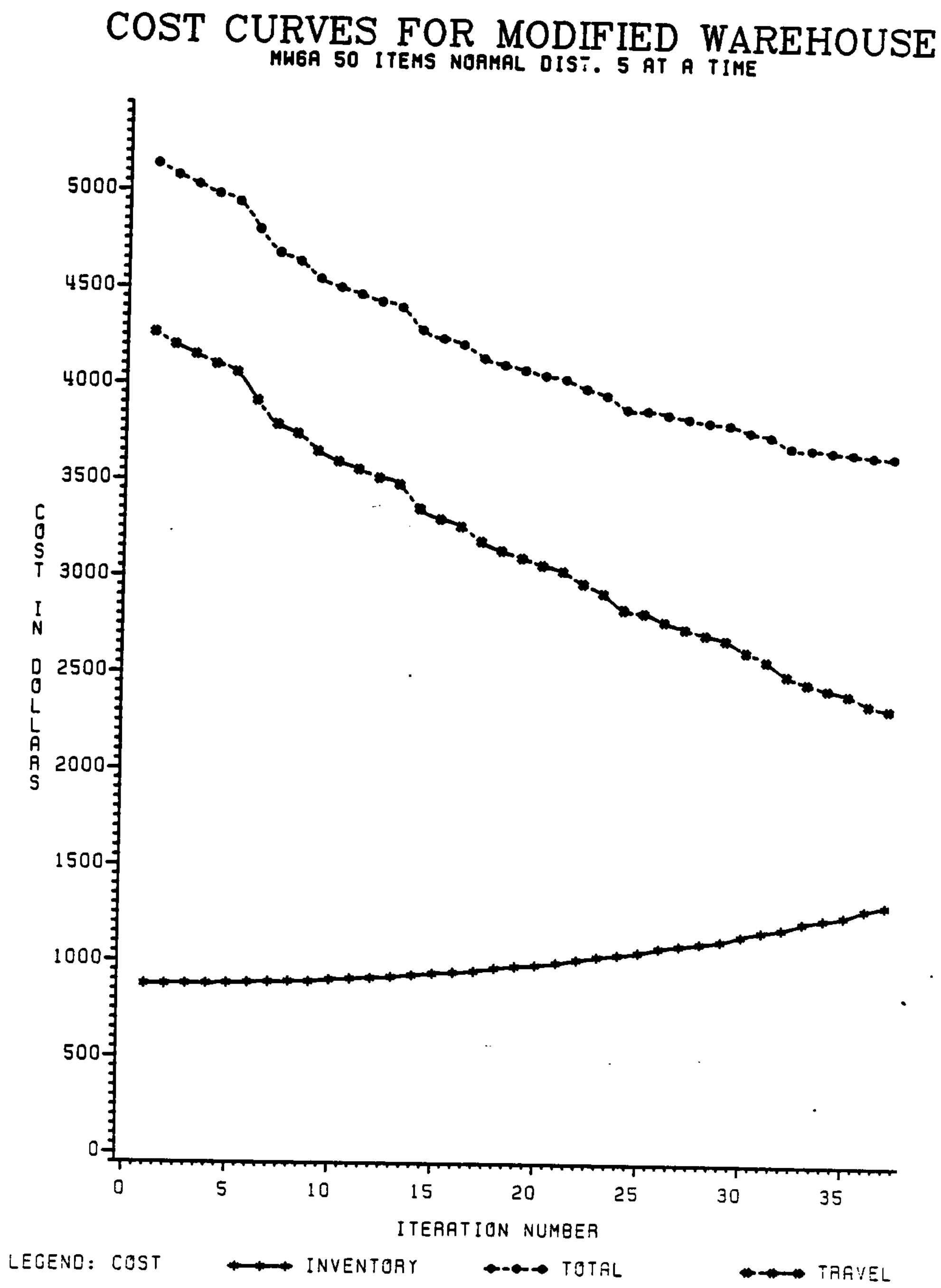

Figure 5.4.3. Cost Curves for Modified Warehouse: MW6A.

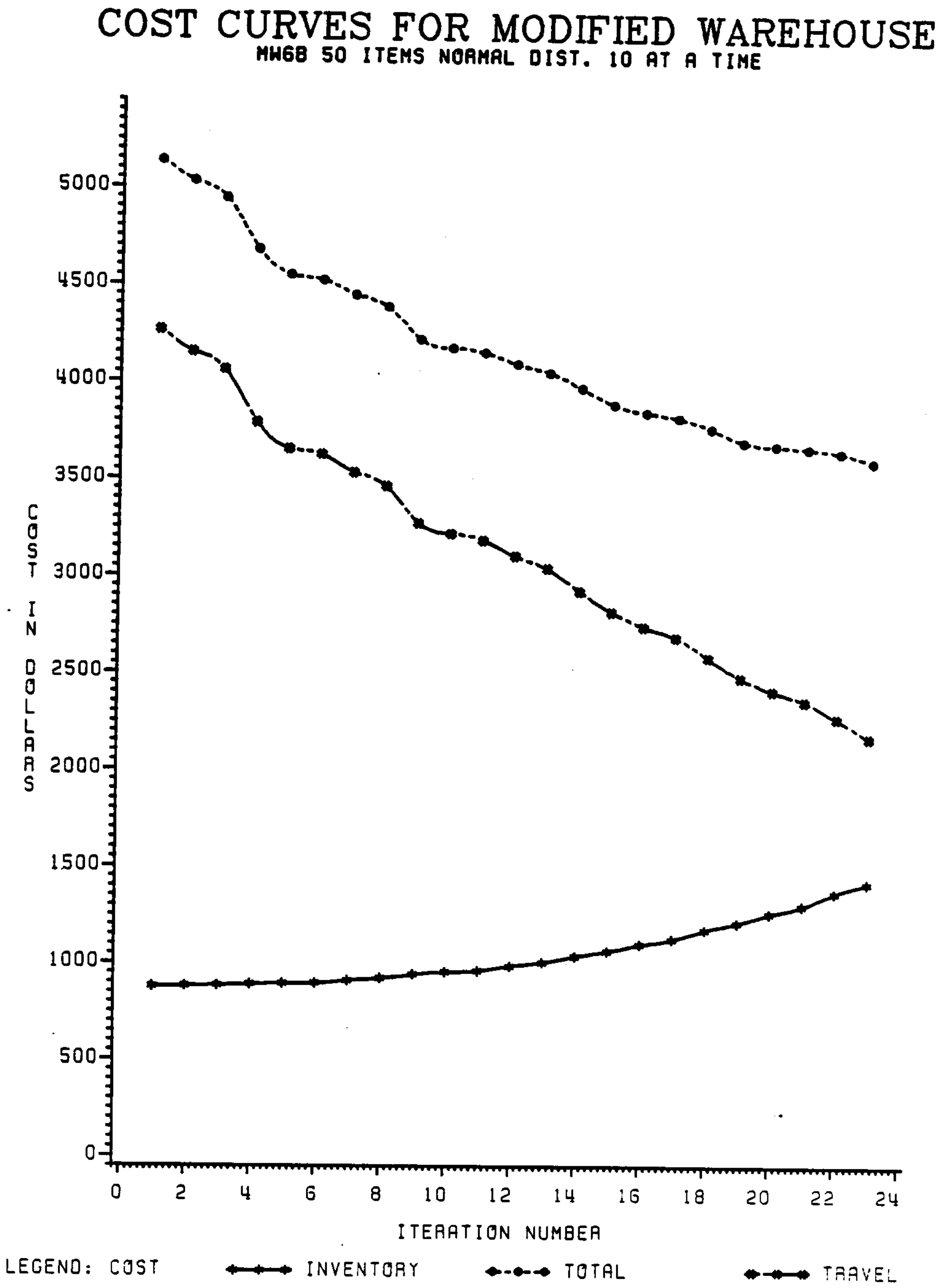

Figure 5.4.4. Cost Curves for Modified Warehouse: MW6B.

This comparison reveals, as expected, a solution closer to the original in the five item simultaneous reduction procedure (MW6A). Both Modified Warehouse Procedures produced solutions lower than the original by 1.5% and 2.99% respectively. Warehouse utilization decreases as the number of items simultaneously reduced increases, with a high volume reduction of 55% for MW6B. Again, as warehouse utilization is decreased travel costs also decreases to 64.06% and 60.49% of the final total costs. Thus, inventory cost rise from an original value of 31.17%, to values of 35.94% and 39.51% of final total costs for the five item and ten item simultaneous reduction problems. The number of successful iterations is reduced to 27% and 16% of the original number for MW6A and MW6B, respectively. CPU time reduces to 21% and 12% for the five and ten item reduction problems, respectively.

5.5 SUMMARY

This section concludes a detailed description of the test problems and results obtained using the Interleaving Warehouse Layout Model and the Modified Warehouse Procedure. Two sizes of test problems were studied, a 50 item 110,000 cubic foot warehouse and a 25 item 55,000 cubic foot warehouse. The solution quality remained approximately the same for both, at 28% and 27% reduction in total cost, respectively. Normally and uniformly distributed demand characteristics were studied and no significant deviations in the solutions were found. The Modified Warehouse Procedure was found to reduce CPU time by as much as 75% while being within ± 4% of the final solution generated by the original algorithm. Warehouse utilization, or the amount of storage space used, reduced by approximately 38% on the average, using the original algorithm, and by approximately 48% using the Modified Warehouse Procedure.

The final phase of this research is discussed in Chapter VI. The chapter presents conclusions and recommendations of model extensions derived from this research. These extensions are specifically concerned with the applicability of the Interleaving Warehouse Layout Model to a wider variety of realistic warehousing problems.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

This research has focused on the quantitative analysis of those factors that determine the operational efficiency of an interleaving warehouse. A detailed analysis of realistic warehouse test problems was performed to obtain the relationship between inventory and travel costs. Although not an optimizing technique, the model implementation functions as a management planning tool to assess the tradeoffs in handling costs among various stock arrangements and reorder quantities, to achieve a minimum total cost.

One of the major simplifications made in most stock location models has been the practice that the item space requirements, derived from the simple economic order quantity inventory model, are taken as given. This research was initiated to provide a means to improve upon this method. If one were to follow the above practice, needed warehouse space would exceed that required under cost optimal operating policy and costs would be in excess of what they could be under the policies suggested by the model.

Considering all the previously discussed single item reduction problems whose results are capsuled in Table 6.1.1, the mean reduction in total cost was 27.46% with a high reduction of 28.99% and a low reduction of 25.5%. Inventory cost increased by an average of 30.21% with the highest increase being 33.78% and the lowest being 25.38%. Travel costs decreased by an average of 41.22% with a high and low decrease of 41.74% and 39.39%, respectively. Warehouse utilization,

Table 6.1.1. Summary Table.

Problem	Items	Distribution	Group Size	Total Cost EOQ	% Inventory Cost	% Travel Cost	Final Total Cost	% Inventory Cost	% Travel Cost	% Δ Total Cost	Inventory Cost EOQ	Inventory Cost Final	% Δ Inventory Cost	Travel Cost EOQ	Travel Cost Final	% Δ Travel Cost	Successful Iterations	% Warehouse Utilization EOQ	% Warehouse Utilization Final	% Δ Warehouse Utilization	Final Solution % Δ from original algorithm
TP1	25	Uniform	1	2219.04	20.33	79.67	1624.15	36.58	63.42	-26.81	451.05	594.13	31.72	1768.00	1030.02	-41.74	75	94.40	55.25	-39.15	
TP2	25	Uniform	1	2297.41	20.55	79.45	1668.75	35.70	64.30	-27.36	472.12	595.66	26.17	1825.29	1073.08	-41.21	71	96.53	57.54	-38.99	
TP3	25	Normal	1	1948.60	20.44	79.56	1397.00	37.61	62.39	-28.31	398.29	525.47	31.93	1550.31	871.52	-43.78	76	93.00	49.79	-43.21	
TP4	25	Normal	1	2292.34	21.44	78.56	1707.73	36.09	63.91	-25.50	491.57	616.34	25.38	1800.77	1091.39	-39.39	67	95.82	58.45	-37.37	
Mean					20.69	79.31		36.50	63.50	-27.00			28.80			-41.53				-39.68	
TP5	50	Uniform	1	5911.89	16.23	83.77	4198.74	30.57	69.43	-28.99	959.59	1283.73	33.78	4952.31	2915.01	-41.14	143	95.27	59.80	-35.47	
TP6	50	Normal	1	5136.70	17.03	82.97	3710.60	31.17	68.83	-27.76	874.54	1156.62	32.25	4262.16	2553.99	-40.08	135	92.48	53.66	-38.82	
Mean					16.63	83.37		30.87	69.13	-28.38			33.02			-40.61				-37.15	
MW1	25	Uniform	5	2219.04	20.33	79.67	1636.79	40.82	59.18	-26.24	451.05	668.06	48.11	1768.00	968.72	-45.21	20	94.40	43.02	-51.38	+0.78
MW2	25	Uniform	5	2297.41	20.55	79.45	1661.39	42.08	57.92	-27.68	472.12	699.03	48.06	1825.29	962.36	-47.28	21	96.53	45.71	-50.82	-0.44
MW3	25	Normal	5	1948.60	20.44	79.56	1446.58	37.38	62.62	-25.76	398.29	540.70	35.75	1550.31	905.89	-41.57	18	93.00	45.72	-47.28	+3.55
MW4	25	Normal	5	2292.34	21.44	78.56	1680.62	42.95	57.05	-26.69	491.57	721.82	46.84	1800.77	958.80	-46.76	19	95.82	49.14	-46.68	-1.59
Mean								40.81	59.19	-26.59			44.69			-45.21				-49.04	
MW5A	50	Uniform	5	5911.89	16.23	83.77	4044.57	38.43	61.57	-31.59	959.59	554.38	61.98	4952.31	2490.19	-49.72	43	95.27	44.09	-51.18	-3.67
MW6A	50	Normal	5	5136.70	17.03	82.97	3653.66	35.94	64.06	-28.87	874.54	1313.25	50.16	4262.16	2340.42	-45.09	36	92.48	51.86	-40.62	-1.53
Mean								37.19	62.82	-30.23			56.07			-47.41				-45.90	
MW5B	50	Uniform	10	5911.89	16.23	83.77	4063.11	39.51	60.49	-31.27	959.59	1605.26	67.29	4952.31	2457.85	-50.37	24	95.27	37.62	-57.65	-3.23
MW6B	50	Normal	10	5136.70	17.03	82.97	3599.54	39.51	60.49	-29.93	874.54	1422.21	62.62	4262.16	2177.33	-48.92	22	92.48	37.69	-54.79	-2.99
Mean								39.51	60.49	-30.60			65.00			-49.65				-56.22	

which can be considered as space used in the warehouse, also decreased by an average of 38.84% with the highest decrease being 43.21% and the lowest decrease being 35.47%.

By reducing reorder quantities we are reducing the amount of space used in the warehouse. Thus, we are moving everything closer to the input/output point and are reducing travel costs. Since travel costs account for approximately 80% of the total cost in the base case EOQ model, it is desirable to concentrate on decreasing travel costs in order to decrease total cost. It is acceptable to decrease reorder quantities from the base case EOQ model (i.e., increase inventory costs) to reduce travel cost in search of the lowest total cost and the optimal reorder quantities. Since inventory costs account for only 20% of the total cost in the base case an increase of 30% is equivalent to an increase in total cost of just 6%. Likewise, travel cost which account for 80% of the total cost in the base case is reduced by 40%, which is equivalent to a 32% decrease in total cost. Thus, the tradeoff between increasing inventory cost and decreasing travel cost becomes very desirable.

A hidden benefit of this algorithm is that it maximizes throughput. Assuming demand stays the same, a reduction in inventory promotes a higher turnover of items. Obviously, by reducing the warehouse space used, this allows the warehouse manager to be able to increase the amount of item types he can store in the warehouse.

The solutions obtained from the original algorithm and the modified procedure were very close. Comparing the Modified Warehouse Procedure to the original algorithm, the worst case scenario yielded solutions 3.55% higher and 3.67% lower. Averaging the percentage difference for all eight comparisons leads to a 2.21% difference. The fact that both algorithms produce such similar solutions leads us to believe the function is approaching a local minimum.

Warehouse utilization is always reduced when using the Modified Warehouse Procedure compared to the original. The average utilization reduction was 11.64% with a high of 22.19% and a low reduction of 1.79%. It was observed, as the size of the simultaneously reduced group increased, so did the utilization reduction.

Generally while using the Modified Warehouse Procedure, inventory cost composed 6.4% more of the final total cost than in the original algorithm and travel cost composed 6.4% less. The underlying meaning here, is the Modified Warehouse Procedure reduces travel costs more than the original, while still coming very close to the same solution.

The Modified Warehouse Procedure was developed to reduce the high computational overhead associated with the reallocation of the warehouse and the estimation of the s* parameter for each item reduction. The procedure works by reducing a group of items simultaneously where the original algorithm would reduce one item at a time. Thus, in the original algorithm, if a reduction in reorder quantity of a particular item did not reduce total cost, it would not be reduced and the optimization procedure would be performed on the next highest COI valued item. Using the Modified Warehouse Procedure it is possible that a reduction in reorder quantity of a particular item in the group, leads to that item contributing to an increase in total cost, that is offset by the contributions of the other items in the group to a reduction in total cost. Thus, not all simultaneously reduced items contribute to a reduction in total cost. Since after each reduction the warehouse is reallocated (i.e., items are moved closer to the input/output point) and travel cost decreases, the only way a particular item can contribute to an increase in total cost is if that item's increase in inventory cost is greater than the decrease in travel costs. Due to the fact travel costs account for such a high proportion of total cost, a group reduction of items will reduce travel cost more rapidly than inventory costs can increase total cost. Of course as we approach a local optimum, the reduction in total cost becomes less and will eventually increase. It should also be noted the cost reductions from applying the algorithms would be more modest for problems with comparatively higher inventory costs versus travel costs.

A surprising result of the research was the COI order in the original algorithm did not change from the base case EOQ model to the final solution. Even though the COI order was only observed in the base case and the final optimization, it seems highly unlikely that for six different test problems the COI order could change and then revert back to the original order for the final optimization. Perhaps if the reduction percentage of reorder quantities were increased, or if original COI values were closer together, there would have been a change in the ordering.

In conclusion, the algorithms derived in this research reduce the total cost of the warehouse by approximately 27%. Inventory and travel costs which comprise an approximate 20%-80% relationship in the total cost of the base case is reduced to comprise an approximate 40%-60% relationship of the final total cost solution. Warehouse utilization is reduced an average of 38%. The Modified Warehouse algorithm saves a great deal of computational overhead and reduces warehouse utilization by approximately 10% over the original algorithm, making this a very useful tool to be used by warehouse managers.

6.2 RECOMMENDATIONS

This study provides warehouse management with the ability to assess tradeoffs in handling costs among various stock arrangements and reorder quantities to achieve a minimum total cost. Further research could be done using this model to obtain the optimal warehouse strategy resulting in the minimum total cost. Offered below are some suggestions to enhance the understanding of materials handling in a warehouse.

1) Implement a wider variety of distributions for input parameters to study their effect on the final solution.

2) Modify the algorithm to use a picker routing order picking method. This is used when several items on an order are to be picked on a single trip through the warehouse, until the order picking vehicle capacity is reached, or an order is filled.

3) Apply the algorithm to a manufacturing warehouse where items are grouped in assembly order and not by the COI rule. One could also include compatability constraints between item pairs.

4) Multiple input/output points.

5) Use simulation as a means of optimizing the decision parameters of the system. Azadivar [21] has developed a stochastic

optimization search technique allowing a simulation model to evaluate the objective function for the levels of decision variables supplied by the search method. He shows that for a discrete rack system better results can be obtained by simulating the real system than approximating it with a mathematical model.

Suggestion #1 is straightforward and no modification to the computer program is needed. Application of the algorithm to suggestion #3 would require a rewriting of the allocation routine to include the appropriate constriants. Implementing suggestions #2 and #4 will be a non-trivial task and will require extensive studies and trials.

REFERENCES

1) Ballou, R. H., "Improving the Physical Layout of Merchandise Warehouse", Journal of Marketing, Vol. 31, July, 1967, pp. 60-64.

2) Fabrycky, W. J., and Thuesen, G. J., Economic Decision Analysis, Prentice-Hall, Inc., 1980.

3) Francis, R. J., McGinnis, L. F., and White, J. A., "Location Analysis", European Journal of Operations Research, Vol. 12, 1983, pp. 220-252.

4) Francis, R. L., "On Some Problems of Réctangular Warehouse Design and Layout", Journal of Industrial Engineering, Vol. 12, October, 1967, pp. 595-604.

5) Francis, R. L., and White, J. A., Facility Layout and Location an Analytical Approach, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.

6) Graves, S. C., Hausman, W. H., and Schwarz, L. B., "Storage Retrieval Interlaving in Automatic Warehousing Systems", Management Science, Vol. 23, May, 1977, pp. 935-945.

7) Harmatuck, D. J., "A Comparison of Two Approaches to Stock Location", The Logistics and Transporation Review, Vol. 12, pp. 282-284.

8) 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.

9) Hodgson, T. J., and Lowe, T. J., "Production Lot Sizing and Material Handling Cost Considerations", IIE Transactions, Vol. 14, No. 1, Mrach, 1982, pp. 44-51.

10) 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.

11) Kay, E., A Mathematical Model for Handling in a Warehouse, Pergamon Press, 1968.

12) Lee, L., Jr., and Dobler, D. W., Purchasing and Material Management Texts and Cases, 3rd edition, McGraw-Hill Book, Co., 1977.

13) Magee, J. F., Production Planning and Inventory Control, McGraw-Hill Book Company, Inc., 1958.

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) Petterson, R., and Silver, E. A., Decision Systems for Inventory Management and Production Planning, John Wiley and Sons, 1979.

16) Schwarz, L. B., Graves, S. C., and Hausman, W. H., "Scheduling Policies for Automatic Warehousing Systems: Simulation Results", AIIE Transactions, Vol. 10, No. 3, September, 1978, pp. 260-270.

17) Vaugh, R. C., Introduction to Industrial Engineering, The Iowa State University Press, Ames, Iowa, 1967.

18) Wilson, H. G., "Order Quantity, Product Popularity, and The Location of Stock in a Warehouse", AIIE Transactions, Vol. 9, No. 3, pp. 230-237.

19) Witt, P. R., "Getting the Most Out of a Stacker System Controls are the Secret", Automation, May, 1974, pp. 75-79.

20) Zimmerman, G. W., "The ABC's of Vilfredo Pareto", Production and Inventory Management, Vol. 16, 1975, pp. 1-9.

21) 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.

APPENDIX

USERS GUIDE AND COMPUTER CODE LISTING

The use of the computer code of the Interleaving Warehouse Layout Model is straightforward. There are only five variables that need to be read into the program in the following order CU_j, AOS_j, OD_j, V_j, and $NUACPC_j$ (number of Unique Address Classifications in Preference class j). For example all CU_j values should be placed on the same data card this being followed by a new data card including all AOS_j values. Continue in this fashion until all the data has been entered.

In the GIVEN DATA portion of the program, data must be entered directly to the computer listing. For example, the warehouse data will include such things as the storage volume of a location (STOVOL), the aisle width (AW) and so on. Variable identification can be done by using the variable list.

The computer program was written to include both the original algorithm (reduction of a reorder quantity one item at a time) and the Modified Warehouse Procedure (reduction of several reorder quantities simultaneously). The variable NUMRED located in the GIVEN DATA set specifies the number of reorder quantities to be reduced simultaneously.

```
C
C                         VARIABLE LIST
C
C CUMPER(J) = CUMULATIVE PERCENTAGE DEMAND FOR ITEM J
C TOTD =TOTAL DEMAND
C M = NUMBER OF ITEMS
C POSI(I) = ITEM NUMBER RANKED AT POSITION I
C CU(I) = THE CUBIC AREA OF ONE UNIT OF ITEM I
C OD(I) = NUMBER OF ORDERS PER PERIOD OF ITEM I
C V(I) = THE DOLLAR VALUE OF ONE UNIT OF ITEM I
C CM = INVENTORY CARRYING COST
C CO = COST OF REORDER
C Q(I) = REORDER QUANTITY OF ITEM I
C AOS(I) = AVERAGE ORDER SIZE OF ITEM I
C PERTOD(I) = PERCENTAGE OF TOTAL DEMAND FOR ITEM I
C SPACER(I) = SPACE REQUIREMENT FOR ITEM I
C CSPACR(I) = CUMULATIVE SPACE REQUIREMENTS STARTING WITH THE LOWEST COI
C             VALUED ITEM INCLUDING ITEM I
C NUMPC = NUMBER OF PREFERRED CLASSIFICATIONS
C NUACPC(J) = NUMBER OF UNIQUE ADDRESS CLASSIFICATIONS IN PREFERRED
C             CLASSIFICATION J
C STOVOL= STORAGE VOLUME OF A LOCATION
C SPAPC(I) = VOLUME OF SPACE IN PREFERENCE CLASSIFICATION I
C CUMSPR(J) = CUMULATIVE SPACE REQUIREMENT OF ITEM J ORDERED BY COI
C ITEMN(J) = ITEM NUMBER OF ITEM J
C CSPAPC(I) = CUMULATIVE SPACE IN A PREFERENCE CLASSIFICATION STARTING
C             WITH 1 AND ENDING WITH I
C PTDPC(I) = PERCENTAGE OF TOTAL DEMAND FOR PREFERENCE CLASSIFICATION
C            NUMBER I
C CPTDPC(I) = CUMULATIVE PERCENTAGE OF TOTAL DEMAND FOR PREFERENCE
C             CLASSIFICATION I
C R = NUMBER OF ROWS
C C = NUMBER OF COLUMNS
C NREM = SPACE REMAINING IN A PREFERENCE CLASSIFICATION
C COST(I,J) = COST TO MOVE ANY PRODUCT FROM THE I/O POINT TO LOCATION
C             (I,J)
C NUACPC(I) = NUMBER OF UNIQUE ADDRESS CLASSIFICATIONS IN A PREFERENCE
C             CLASS
C ITEMS = NUMBER OF ITEMS TO BE STORED
C PTD(I) = PERCENTAGE OF TOTAL DEMAND FOR ITEM I
C DOL = DOLLAR COST CONSTANT OF A MOVE
C AW = AISLE WIDTH
C LS = LENGTH OF SIDE OF A STORAGE AREA
C VARN(NN) = VARIABLE NUMBER TO BE PASSED TO OPTIMIZATION SUBROUTINE
C COUNTR = COUNTER TO DESIGNATE WHICH VARIABLE TO PASS TO
C          OPTIMIZATION ROUTINE
```

```
C NUMPCU = NUMBER OF PREFERENCE CLASSES USED
C SUB1,SUB2,SUB3 = SUBSCRIPTS DESIGNATING LOCATION AND ITEM NUMBER OF
C                  VARIABLE TO BE PASSED TO OPTIMIZATION SUBROUTINE
C IPRIM(C,R)**SSTAR = PROPORTION OF ORDERS FILLED FROM THE CLASS OF
C                     LOCATIONS INCLUDING (C,R) AND THOSE CLASSES OF
C                     LOCATIONS CLOSER IN TRAVEL TIME TO THE I/O POINT
C IDPRIM(C,R)**SSTAR = PROPORTION OF ORDERS FILLED FROM THE NEXT LOWER
C                      CLASS OF LOCATIONS FROM (C,R) AND THOSE CLASSES
C                      OF LOCATIONS CLOSER IN TRAVEL TO THE I/O POINT.
C JTPROB(Z,Y) = THE PROBABILITY OF
C                  RETRIEVE LOCATION (Z,Y) BEING
C                  ACCESSED
C TC = TOTAL COST
C TER3 = INVENTORY COST
C TER2 = THE COST OF INTERLEAVE COMBINED WITH COST OF MOVING TO I/O
C        POINT
C TER1A = COST OF MOVING FROM INPUT/OUTPUT POINT TO STORAGE LOCATION
C COUNTR = USED TO COUNT TOTAL COST CALCULATIONS
C X(C,R,L) =  UNIT L IN LOCATION C,R
C COST(C,R) = COST TO MOVE TO LOCATION C,R
C WHCAP = WAREHOUSE CAPACITY
C NUMUAC = NUMBER OF UNIQUE ADDRESS CLASSIFICATIONS IN THE WAREHOUSE
C SSTAR = FIT PARAMETER FOR MODIFIED LOCATION ABC CURVE
C ITER3(L) = INVENTORY COST OF INDIVIDUAL ITEM L
C ITER2(L) = COST OF INTERLEAVE COMBINED WITH RETRIEVE COST FOR
C            INDIVIDUAL ITEM L
C ITER1A(L) = COST OF MOVING FROM I/O POINT TO STORAGE LOCATION OF
C            INDIVIDUAL ITEM L
C DELTA = REDUCTION FACTOR
C COMPLE = TURN ON SWITCH FOR FINAL PRINT OUT
C ITC(L) = INDIVIDUAL ITEM CONTRIBUTION TO TOTAL COST
C NREM = SPACE REMAINING TO BE FILLED IN U.A.C.
C SS =  POINTS ON MODIFIED LOCATION ABC CURVE THAT ARE AVERAGED TO FIND
C       S*
C COI(L) = CUBE PER ORDER INDEX OF ITEM L
C VARN = VARIABLE NUMBER
C SUB1 = CORRESPONDS TO COLUMN
C SUB2 = CORRESPONDS TO ROW
C SUB3 = CORRESPONDS TO ITEM
C ISUMMM(L), ISUM1(L), ISUM3(L) = SUMMATIONS USED TO CALCULATE INDIVIDUAL
C                                 ITEM COSTS
C STPROB = SUMMATION OF JOINT PROBABILITIES
C STOPCT = COUNTER TO STOP PROGRAM
C II = ITEM COUNTER
C OPTCNT = OPTIMIZATION COUNTER
C TCPREV = TOTAL COST OF PREVIOUS ITEM THAT HAS BEEN ADJUSTED BY 1/10 OF
C          1%
C NUMRED = NUMBER OF ITEMS REDUCED BEFORE REEVALUATING S*
C
C
      COMMON COUNTR,CU(50),Q(50),OD(50),ITEMS,POSI(50),TOTD,PERTOD(50)
      COMMON R,C,NUMPCU,NUMPC,CSPAPC(23)
      COMMON X(12,12,50),COST(12,12),TC(7500),WHCAP,CSPACR(50),NUMUAC
      COMMON SSTAR,NUACPC(23),AW,LS,DOL,CO,CM,V(50),STOVOL,AOS(50)
      COMMON TER1,TER2,TER3,DELTA,KK,CPTDPC(23),COMPLE,ITPCS(50)
```

```
      COMMON ITER1(50),ITER3(50),ITER2(50),ITC(50),TPCS,NUMRED
      COMMON SUB1(7500),SUB2(7500),SUB3(7500),VARN(7500),NN,AAA(10)
      REAL CU,V,Q,CUMPER(50),NREM,X,TC,CSPACR,WHCAP
     REAL SPACER(50),CSPAPC,SPAPC(23),SS(22),IPRIM(12,12),IDPRIM(12,12)
      REAL PERTOD,COI(50),LS,COST,JTPROB(12,12),ITPCS
      REAL SUMM(22)/22*0.0/,PTD(23),PTDPC(23),CPTDPC,VARN
      INTEGER OD,TOTD,NUMPC,NUACPC,NUMPCU,SUB1
      INTEGER POSI,A,C,B,D,NNUM,R,NN,COUNTR,NNN,Z,Y,SUB2
      INTEGER SUB3,AOS,NUMUAC,COMPLE,NUMRED
      READ,(CU(J),J=1,50)
      READ,(AOS(J),J=1,50)
      READ,(OD(J),J=1,50)
      READ,(V(J),J=1,50)
      READ,(NUACPC(J),J=1,23)
C
C GIVEN DATA
C
      STOVOL = 765.6
      ITEMS=50
      NUMPC = 23
      NUMUAC=144
      CM = .006
      CO =5.0
      R=12
      C=12
      DOL=0.003
      AW=10.0
      LS=27.67
      COMPLE=0
      NUMRED=10
C
C PRINT OUT INPUT DATA
C
      PRINT 9964,'REDUCE', NUMRED,'ITEMS AT A TIME OF',ITEMS,' ITEM
     CPROBLEM UNIFORM  DIST.'
 9964  FORMAT('1',T30,A6,1X,I2,1X,A18,1X,I3,1X,A27)
      PRINT 1345,'INPUT ITEM DATA'
 1345  FORMAT(/,T58,A15)
      PRINT 1355,'NUMBER OF VARIABLES = ',ITEMS*NUMUAC
 1355  FORMAT(/,T54,A22,I5)
      PRINT 1356,'WAREHOUSE LAYOUT ',C,' COLUMNS ',R,' ROWS '
 1356  FORMAT(/,T48,A17,I2,A8,I2,A6)
      PRINT 1347,'ITEM','CU','AOS','OD','V'
 1347  FORMAT(/,T11,A4,T49,A2,T65,A3,T90,A2,T117,A1)
      DO 1346 I=1,ITEMS
         PRINT 1348,I,CU(I),AOS(I),OD(I),V(I)
 1348     FORMAT(/,T13,I2,T47,F5.2,T64,I3,T89,I3,T115,F5.2)
 1346  CONTINUE
      PRINT 1349,'WAREHOUSE INPUT DATA'
 1349  FORMAT(//,T56,A20)
      PRINT 1351,'P.C.','NUACPC'
 1351  FORMAT(/,T11,A4,T24,A6)
      DO 1350 I=1,NUMPC
         PRINT 1352,I,NUACPC(I)
 1352     FORMAT(/,T11,I2,T26,I2)
```

```
 1350   CONTINUE
C
C CALCULATION OF EOQ REORDER QUANTITIES
C
      DO 10 J=1,ITEMS
          Q(J)=((2*CO*AOS(J)*OD(J))/(CM*V(J)))**.5
   10 CONTINUE
C
C CALCULATE VOLUME OF SPACE IN A PREFERENCE CLASSIFICATION
C
       SUM = 0
      DO 60 J=1,NUMPC
          SPAPC(J)=NUACPC(J)*STOVOL
          CSPAPC(J)=SUM+SPAPC(J)
          SUM=CSPAPC(J)
  60  CONTINUE
C
C COMPUTE TOTAL DEMAND FOR ALL ITEMS
C
      TOTD=0.0
      DO 30 J=1,ITEMS
        TOTD=TOTD+OD(J)
  30  CONTINUE
C
C CUMPUTE THE PERCENTAGE OF TOTAL DEMAND FOR EACH ITEM
C
      DO 3000 J=1,ITEMS
        PERTOD(J)=FLOAT(OD(J))/TOTD
 3000 CONTINUE
C
C ROUTINE TO CALCULATE COST OF MOVE (COST(C,R)): STORE
C
       DO 170 I=1,R
        DO 180 J=1,C
C CALCULATE COST OF MOVE FOR SPACE (1,1)
          IF (J .EQ. 1 .AND. I .EQ. 1) THEN
             COST(J,I)=2*AW*DOL
             GO TO 180
          ENDIF
        COST(J,I)=((((IABS(J-1))+(IABS(I-1)))*LS)+((IABS(J-1))+
     C(IABS(I-1)))*AW)*DOL
 180    CONTINUE
 170   CONTINUE
C
C EQUATING COST(Z,Y) TO COST(C,R): RETRIEVE
C
      Y=0
      DO 171 I=1,R
       Y=Y+1
       Z=0
        DO 181 J=1,C
          Z=Z+1
          COST(Z,Y)=COST(J,I)
  181    CONTINUE
  171 CONTINUE
```

```
C
C PRINT TITLES
C
         PRINT 9901,'UTILIZATION'
 9901     FORMAT('1',T85,A11)
         PRINT 9902,'FIT','REDUCED','OF','REORDER','CUMMULATIVE'
 9902     FORMAT(1X,T58,A3,T67,A7,T89,A2,T100,A7,T120,A11)
         PRINT 9903,'TOTAL COST','INVENTORY COST','TRAVEL COST',
     C'PARAMETER','ITEM','DELTA','WAREHOUSE','QUANTITIES','PC',
     C'% DEMAND'
 9903     FORMAT(1X,T5,A10,T21,A14,T39,A11,T55,A9,T68,A4,T77,A5,T86,A9,
     CT99,A10,T115,A2,T122,A8)
        COUNTR=0
        CALL LAYOUT
C
C WAREHOUSE CAPACITY EXCEEDED STOP PROGRAM
C
        K=POSI(ITEMS)
        IF(CSPACR(K) .GT. WHCAP ) GO TO 3333
C
C
        CALL TCOST
        CALL FINAL
        CALL OPT
C
 3333   STOP
        END
C
C SUBROUTINE LAYOUT
C
         SUBROUTINE LAYOUT
       COMMON COUNTR,CU(50),Q(50),OD(50),ITEMS,POSI(50),TOTD,PERTOD(50)
       COMMON R,C,NUMPCU,NUMPC,CSPAPC(23)
       COMMON X(12,12,50),COST(12,12),TC(7500),WHCAP,CSPACR(50),NUMUAC
       COMMON SSTAR,NUACPC(23),AW,LS,DOL,CO,CM,V(50),STOVOL,AOS(50)
       COMMON TER1,TER2,TER3,DELTA,KK,CPTDPC(23),COMPLE,ITPCS(50)
       COMMON ITER1(50),ITER3(50),ITER2(50),ITC(50),TPCS,NUMRED
       COMMON SUB1(7500),SUB2(7500),SUB3(7500),VARN(7500),NN,AAA(10)
       REAL ISUM1(50),ISUM2(50),ISUM3(50),ITER1,ITER2,ITER3,ISUMMM(50)
       REAL ISUM(50),ITC
       REAL CU,Q,PERTOD,COI(50),CUMPER(50),SUM,SPACER(50),CSPACR
       REAL X,NREM,VARN,SUMM(50)/50*0.0/,SS(22),SSTAR,CPTDPC,ITPCS
       REAL CSPAPC,PTDPC(23),PTD(23)
       INTEGER NN,M,SUB1,SUB2,SUB3,B,A,NNN,NNUM
       INTEGER D,R,C,OD,TOTD,POSI,NUMPC,COUNTR,AOS,NUMUAC,COMPLE
C
C  CALCULATION OF CUBE PER ORDER INDEX
C
      DO 20 J=1,ITEMS
          COI(J)=CU(J)*Q(J)/OD(J)
  20  CONTINUE
C
C RANK ITEMS ACCORDING TO COI VALUES
C
            CALL RANK(COI,ITEMS,POSI)
```

```
C
C CALCULATE CUMMULATIVE PERCENTAGE DEMAND FOR RANKED ITEMS
C
       SUM=0.0
       DO 40 I=1,ITEMS
          K=POSI(I)
          CUMPER(K)=SUM+(OD(K)/FLOAT(TOTD))
          SUM=CUMPER(K)
  40   CONTINUE
C
C CALCULATE SPACE REQUIREMENTS FOR EACH ITEM
C
       SUM=0.0
       DO 50 I=1,ITEMS
          K=POSI(I)
          SPACER(K)=CU(K)*Q(K)
          CSPACR(K)= SUM+SPACER(K)
          SUM=CSPACR(K)
  50   CONTINUE
C
C CHECK TO SEE IF WAREHOUSE CAPACITY IS EXCEEDED
C
       WHCAP=NUMUAC*STOVOL
       K=POSI(ITEMS)
       IF(CSPACR(K) .GT. WHCAP) THEN
       PRINT,'WAREHOUSE CAPACITY EXCEEDED'
       GO TO 9999
       ENDIF
C
C INITIALIZE ALL  X(C,R,J)
C
       DO 900 K=1,ITEMS
          DO 901 J=1,R
             DO 902 I=1,C
                X(I,J,K)=0.0
 902         CONTINUE
 901      CONTINUE
 900   CONTINUE
C
C ALLOCATION OF ITEMS TO UNIQUE ADDRESSES
C
C THIS ALLOCATION PROCEDURE CAN ONLY HANDLE SQUARE WAREHOUSES AND THOSE
C RECTANGULAR WAREHOUSES WHERE THE NUMBER OF COLUMNS OUT NUMBER
C THE ROWS BY ONE
C
       NN=1
       K=1
       L=POSI(K)
       S=SPACER(L)
       M=2
C
C FOR SQUARE AND RECTANGULAR WAREHOUSES STARTING WITH THE FIRST SPACE
C AND FILLING ALL SPACES OF THE LARGEST DIAGONAL
C
 1200  JJ=M-1
```

```
       DO 200 I=1,JJ
          J=M-I
          NREM=STOVOL
C
C COMPLETE STORAGE
C
 120      IF (S .LT. NREM) THEN
             X(I,J,L)=S/CU(L)
             VARN(NN)=X(I,J,L)
             SUB1(NN)=I
             SUB2(NN)=J
             SUB3(NN)=L
             NN=NN+1
             NREM=NREM-S
             K=K+1
             IF (K .GT. ITEMS) GO TO 4000
             L=POSI(K)
             S=SPACER(L)
             GO TO 120
          ELSE
C
C OVERLAP STORAGE
C
             X(I,J,L)=NREM/CU(L)
             VARN(NN)=X(I,J,L)
             SUB1(NN)=I
             SUB2(NN)=J
             SUB3(NN)=L
             NN=NN+1
             S=S-NREM
          ENDIF
200    CONTINUE
       M=M+1
 2000 IF ((M-1) .LE. R) GO TO 1200
       IF((M-1) .GE. C) GO TO 150
 160    B=M-R
        JJ=M-1
       DO 400 I=B,JJ
          J=M-I
          NREM=STOVOL
C
C COMPLETE STORAGE
C
 140      IF (S .LT. NREM) THEN
             X(I,J,L)=S/CU(L)
             VARN(NN)=X(I,J,L)
             SUB1(NN)=I
             SUB2(NN)=J
             SUB3(NN)=L
             NN=NN+1
             NREM = NREM-S
             K=K+1
             IF (K .GT. ITEMS) GO TO 4000
             L=POSI(K)
             S=SPACER(L)
```

```
            GO TO 140
          ELSE
C
C OVERLAP STORAGE
C
            X(I,J,L)=NREM/CU(L)
            VARN(NN)=X(I,J,L)
            SUB1(NN)=I
            SUB2(NN)=J
            SUB3(NN)=L
            NN=NN+1
            S=S-NREM
          ENDIF
 400    CONTINUE
        M=M+1
        IF ((M-1) .LE. C) GO TO 160
C
C FOR A SQUARE WAREHOUSE ALLOCATION FOR EVERYTHING BEYOND LARGEST
C DIAGONAL.    FOR RECTANGULAR WAREHOUSE ALLOCATION FOR DOUBLE OF THE
C LARGEST DIAGONAL AND EVERYTHING BEYOND.
C
 150    A=M-R
        DO 300 I=A,C
          J=M-I
          NREM=STOVOL
C
C COMPLETE STORAGE
C
 130      IF (S .LT. NREM) THEN
            X(I,J,L)=S/CU(L)
            VARN(NN)=X(I,J,L)
            SUB1(NN)=I
            SUB2(NN)=J
            SUB3(NN)=L
            NN=NN+1
            NREM=NREM-S
            K=K+1
            IF (K .GT. ITEMS) GO TO 4000
            L=POSI(K)
            S=SPACER(L)
            GO TO 130
          ELSE
C
C OVERLAP STORAGE
C
            X(I,J,L)=NREM/CU(L)
            VARN(NN)=X(I,J,L)
            SUB1(NN)=I
            SUB2(NN)=J
            SUB3(NN)=L
            NN=NN+1
            S=S-NREM
          ENDIF
 300    CONTINUE
        M=M+1
```

```
      IF ((M-1) .LE.(R+C)) GO TO 150
 4000 NUMPCU=I+J-1
C
C PRINT OUT FINAL ALLOCATION
C
       IF(COMPLE .EQ. 1) THEN
         PRINT 9916,'OPTIMAL ALLOCATION OF ITEMS'
 9916     FORMAT(//,T52,A27)
         NNN=NN-1
         DO 703 I=1,NNN
            PRINT 9917,'X(',SUB1(I),SUB2(I),SUB3(I),') = ',VARN(I)
 9917        FORMAT(/,A2,I2,I2,I3,A4,F14.7)
 703      CONTINUE
         PRINT 9933,'***************************************************
     C***************************************************************
     C*****************'
 9933     FORMAT(/,T1,A132)
       ENDIF
C
C CALCULATE PERCENTAGE OF DEMAND FOR EACH PREFERENCE CLASSIFICATION
C
C       ZERO OUT VARIABLES NEEDED TO CALCULATE % OF DEMAND FOR P.C.
        DO 7807 K=1,ITEMS
           L=POSI(K)
           DO 7808 J=1,R
              DO 7809 I=1,C
                 NNUM=I+J-1
                 PTD(NNUM)=0.0
                 PTDPC(NNUM)=0.0
                 SUMM(NNUM)=0.0
 7809         CONTINUE
 7808      CONTINUE
 7807    CONTINUE
        DO 1901 K=1,ITEMS
           L=POSI(K)
           DO 1902 J=1,R
              DO 1903 I=1,C
                 NNUM=I+J-1
                 IF (X(I,J,L) .GT. 0) THEN
                    PTD(NNUM)=(X(I,J,L)/Q(L))*PERTOD(L)
                    PTDPC(NNUM)=SUMM(NNUM)+PTD(NNUM)
                    SUMM(NNUM)=PTDPC(NNUM)
                 ENDIF
 1903         CONTINUE
 1902      CONTINUE
 1901   CONTINUE
C
C CALCULATION OF TOTAL CUMULATIVE PERCENTAGE DEMAND FOR EACH PREFERENCE
C                              CLASSIFICATION
C
        SUM=0.0
        DO 1001 I=1,NUMPC
           CPTDPC(I)=PTDPC(I)+SUM
           SUM=CPTDPC(I)
 1001   CONTINUE
```

```
C
C CALCULATION OF S AND S*
C
        SUMS=0
        D=NUMPCU-1
        DO 210 J=1,D
          L=POSI(ITEMS)
          SS(J)=(ALOG10(CPTDPC(J)))/(ALOG10(CSPAPC(J)/CSPACR(L)))
           SUMS=SS(J)+SUMS
 210    CONTINUE
        SSTAR=SUMS/D
 9999   RETURN
        END
C
C
C
C TOTAL COST CALCULATIONS
C
C
          SUBROUTINE TCOST
C
       COMMON COUNTR,CU(50),Q(50),OD(50),ITEMS,POSI(50),TOTD,PERTOD(50)
       COMMON R,C,NUMPCU,NUMPC,CSPAPC(23)
       COMMON X(12,12,50),COST(12,12),TC(7500),WHCAP,CSPACR(50),NUMUAC
       COMMON SSTAR,NUACPC(23),AW,LS,DOL,CO,CM,V(50),STOVOL,AOS(50)
       COMMON TER1,TER2,TER3,DELTA,KK,CPTDPC(23),COMPLE,ITPCS(50)
       COMMON ITER1(50),ITER3(50),ITER2(50),ITC(50),TPCS,NUMRED
       COMMON SUB1(7500),SUB2(7500),SUB3(7500),VARN(7500),NN,AAA(10)
       REAL ISUM1(50),ISUM2(50),ISUM3(50),ITER1,ITER2,ITER3,ISUMMM(50)
       REAL ISUM(50),ITC
       REAL X,COST,SSTAR,AW,LS,DOL,CO,CM,V,JTPROB(12,12),Q
       REAL IPRIM(12,12),IDPRIM(12,12),TC,WHCAP,CSPACR,ISUMMN
       REAL ITPCS,ITER1A(50),ITPC(50)
       INTEGER POSI,R,C,ITEMS,OD,NUMPCU,NUACPC,Y,Z,COUNTR,AOS,NUMUAC
       INTEGER COMPLE
       COUNTR=COUNTR+1
C
C CALCULATION OF Q(J)
C
       DO 903 K=1,ITEMS
          SUM=0
          L=POSI(K)
          DO 930 J=1,R
             DO 931 I=1,C
                Q(L)=SUM+X(I,J,L)
                SUM=Q(L)
 931         CONTINUE
 930      CONTINUE
 903   CONTINUE
C
C CALCULATION OF  I'(Z,Y)**SSTAR
C                 I"(Z,Y)**SSTAR
C
       DO 909 J=1,R
          DO 908 I=1,C
```

```
             IPRIM(I,J)=(I+J-1)/FLOAT(NUMPCU)
             IDPRIM(I,J)=(I+J-2)/FLOAT(NUMPCU)
 908       CONTINUE
 909     CONTINUE
C
C CALCULATION OF  PROBABILITIES
C
        DO 915 Y=1,R
           DO 914 Z=1,C
C
C
C  PROBABILITIES OF U.A.C. USED (RETRIEVE)
C
         IF((Z+Y-1) .GT. NUMPCU) THEN
            JTPROB(Z,Y)=0.0
            GO TO 914
         ENDIF
C
                     JTPROB(Z,Y)=
     C(((IPRIM(Z,Y)**SSTAR)-(IDPRIM(Z,Y)**SSTAR))
     C/NUACPC(Z+Y-1))
 914       CONTINUE
 915     CONTINUE
C
C
C NORMALIZATION OF  PROBABILITIES
C
        SUM=0.0
C HOLD STORAGE LOCATION CONSTANT ADD ALL PROBABILITIES FOR INTERLEAVE
              DO 3003 Z=1,C
                 DO 3004 Y=1,R
                     STPROB=SUM+JTPROB(Z,Y)
                     SUM=STPROB
 3004            CONTINUE
 3003         CONTINUE
              DO 3005 Z=1,C
                 DO 3006 Y=1,R
C DISREGARD PREFERRENCE CLASSES NOT USED
                    IF((Z+Y-1).GT.NUMPCU)GO TO3006
                     JTPROB(Z,Y)= JTPROB(Z,Y)/STPROB
 3006            CONTINUE
 3005         CONTINUE
C
C
C CALCULATION OF TRAVEL COSTS
C
C
         TPCS=0
C COST OF STORE
         DO 907 I=1,C
            DO 906 J=1,R
              SUM=0
              DO 905 K=1,ITEMS
                 TER1A=((X(I,J,K)/Q(K))*OD(K))+SUM
                 SUM=TER1A
```

```
 905          CONTINUE
              TER1=TER1A*COST(I,J)
              SUM=0
C COST OF INTERLEAVE AND RETURN TRAVEL
              DO 912 Z=1,C
                 DO 913 Y=1,R
                  TER2=(((((FLOAT((IABS(I-Z))+(IABS(J-Y)))*(AW+LS))
     C*DOL)+COST(Z,Y))*JTPROB(Z,Y))*TER1A)+SUM
                  SUM=TER2
 913             CONTINUE
 912          CONTINUE
              TPC=TER2+TER1+TPCS
              TPCS=TPC
 906       CONTINUE
 907    CONTINUE
C CALCULATION OF INDIVIDUAL ITEM TRAVEL COSTS
        IF(COUNTR .EQ. 1 .OR. COMPLE .EQ. 1) THEN
          DO 9347 K=1,ITEMS
             ITPCS(K)=0.0
 9347     CONTINUE
C INDIVIDUAL ITEM COST OF STORE
          DO 919 I=1,C
            DO 918 J=1,R
              DO 917 K=1,ITEMS
                 ITER1A(K)=((X(I,J,K)/Q(K))*OD(K))
                 ITER1(K)=ITER1A(K)*COST(I,J)
 917          CONTINUE
              DO 916 K=1,ITEMS
                 ISUM(K)=0.0
 916          CONTINUE
C INDIVIDUAL ITEM COST OF INTERLEAVE AND RETRIEVE
             DO 9342 Z=1,C
                 DO 9343 Y=1,R
                  DO 9345 K=1,ITEMS
                   ITER2(K)=(((((FLOAT((IABS(I-Z))+(IABS(J-Y)))*(AW+LS)
     C)*DOL)+COST(Z,Y))*JTPROB(Z,Y))*ITER1A(K))+ISUM(K)
                   ISUM(K)=ITER2(K)
 9345             CONTINUE
 9343            CONTINUE
 9342          CONTINUE
               DO 9346 K=1,ITEMS
                  ITPC(K)=ITER2(K)+ITER1(K)+ITPCS(K)
                  ITPCS(K)=ITPC(K)
 9346          CONTINUE
 918        CONTINUE
 919      CONTINUE
        ENDIF
       SUMMM=0
C
C CALCULATE INVENTORY COST (TER3) FOR INDIVIDUAL ITEMS AND TOTAL
C
       SUM=0
C INITIALIZE INDIVIDUAL SUMS
        DO 9817 K=1,ITEMS
           L=POSI(K)
```

```
            ISUM3(L)=0.0
 9817     CONTINUE
          DO 920 K=1,ITEMS
             L=POSI(K)
             TER3=(CO*OD(L)*AOS(L)/Q(L))+(CM*V(L)*Q(L)/2)+SUM
             IF(COUNTR .EQ. 1 .OR. COMPLE .EQ. 1) THEN
               ITER3(L)=(CO*OD(L)*AOS(L)/Q(L))+(CM*V(L)*Q(L)/2)+ISUM3(L)
             ENDIF
             SUM=TER3
             IF(COUNTR .EQ. 1 .OR. COMPLE .EQ. 1) THEN
               ISUM3(L)=ITER3(L)
             ENDIF
 920      CONTINUE
C
C THE COMPLETE TOTAL COST EQUATION
C
          TC(COUNTR)=TPC+TER3
C INDIVIDUAL ITEM CONTRIBUTIONS TO TOTAL COST
          IF(COUNTR .EQ. 1 .OR. COMPLE .EQ. 1) THEN
              DO 9818 K=1,ITEMS
                  ITC(K)=ITPCS(K)+ITER3(K)
 9818         CONTINUE
          ENDIF
          IF(COMPLE .EQ. 1 ) THEN
C
C PRINT OUT INDIVIDUAL ITEM CONTRIBUTIONS TO TOTAL COST (E.O.Q.)
C
              PRINT 9922,'INDIVIDUAL ITEM CONTRIBUTIONS TO TOTAL COST'
 9922          FORMAT(/,T44,A43)
              PRINT 9923,'TOTAL COST','INVENTORY COST','TRAVEL COST','ITEM'
 9923          FORMAT(/,1X,T5,A10,T21,A14,T39,A11,T55,A4)
              DO 9924 I=1,ITEMS
                 PRINT 9925, ITC(I),ITER3(I),ITPCS(I),I
 9925             FORMAT(/,2X,F15.7,3X,F14.7,3X,F14.7,T56,I3)
 9924         CONTINUE
             PRINT 9932,'*******************************************************
     C***************************************************************
     C*****************'
 9932         FORMAT(/,T1,A132)
          ENDIF
          RETURN
          END
C
C
          SUBROUTINE OPT
C
C
         COMMON COUNTR,CU(50),Q(50),OD(50),ITEMS,POSI(50),TOTD,PERTOD(50)
         COMMON R,C,NUMPCU,NUMPC,CSPAPC(23)
         COMMON X(12,12,50),COST(12,12),TC(7500),WHCAP,CSPACR(50),NUMUAC
         COMMON SSTAR,NUACPC(23),AW,LS,DOL,CO,CM,V(50),STOVOL,AOS(50)
         COMMON TER1,TER2,TER3,DELTA,KK,CPTDPC(23),COMPLE,ITPCS(50)
         COMMON ITER1(50),ITER3(50),ITER2(50),ITC(50),TPCS,NUMRED
         COMMON SUB1(7500),SUB2(7500),SUB3(7500),VARN(7500),NN,AAA(10)
         REAL ISUM1(50),ISUM2(50),ISUM3(50),ITER1,ITER2,ITER3,ISUMMM(50)
```

```
      REAL ISUM(50),ITC,ITPCS
      INTEGER STOPCT,OPTCNT,II,K,R,C,ITEMS,L,POSI,COUNTR,AOS,STOPER
      INTEGER NUMUAC,COMPLE,AAA,NUMRED
      REAL DELTA,X,SUM,Q,TC,OQ(50),CSPACR,WHCAP,TCPREV
      STOPER=3*ITEMS/10
C
C PICK VARIABLES FOR CHANGE
C
      NOPTSU=0
      STOPCT=0
      II=1
      III=NUMRED
C
C RETAIN ORIGINAL Q VALUES: REDUCED Q VALUE OF ITEM K NOW = Q(K):
C                           ORIGINAL Q VALUE OF ITEM K NOW = OQ(K)
C
 7399 OPTCNT=1
      DELTA=.8
 7999 DO 8800 K=1,ITEMS
         OQ(K)=Q(K)
 8800 CONTINUE
 7499 IF(II .GT. ITEMS) THEN
         II=1
         III=NUMRED
      ENDIF
      KK=POSI(II)
C
C REDUCE ITEM FIVE ITEMS AT A TIME
C
         L=1
      DO 7506 J=II,III
         KK=POSI(J)
         AAA(L)=KK
         L=L+1
         Q(KK)=DELTA*Q(KK)
 7506 CONTINUE
C
C LAYOUT WAREHOUSE AND CALCULATE TOTAL COST
C
      CALL LAYOUT
      CALL TCOST
C
C COMPARE TOTAL COST OF REDUCTION: REDUCTION MUST BE 1/10 OF ONE PERCENT
C OF THE PREVIOUSELY CALCULATED COST TO BE ACCEPTED
C
      TCPREV=0.999*TC(COUNTR-1)
      IF(TC(COUNTR) .LT. TCPREV) THEN
        CALL FINAL
        NOPTSU=NOPTSU+1
        II=II+NUMRED
        III=III+NUMRED
        DELTA=.8
        STOPCT=0
        OPTCNT=1
        GO TO 7999
```

```
         ENDIF
         STOPCT=STOPCT+1
         OPTCNT=OPTCNT+1
C
C BRING Q VALUES BACK TO ORIGINAL VALUES
C
        DO 7505 L=1,ITEMS
          Q(L)=OQ(L)
 7505   CONTINUE
C
C SECOND AND THIRD REDUCTIONS
C
        IF(OPTCNT .EQ. 2) THEN
          DELTA=DELTA+0.1
          COUNTR=COUNTR-1
          GO TO 7499
        ENDIF
        IF (OPTCNT .EQ. 3) THEN
           DELTA=DELTA+0.05
           COUNTR=COUNTR-1
           GO TO 7499
        ENDIF
C
C PROGRAM COMPLETION
C
        IF (STOPCT .GE. STOPER) THEN
           PRINT, 'PROGRAM COMPLETION NUMBER OF SUCCESSFUL OPTIMIZATIONS =
     C ',NOPTSU
           COMPLE=1
           CALL LAYOUT
           CALL TCOST
           GO TO 7555
        ENDIF
           II=II+NUMRED
           III=III+NUMRED
           COUNTR=COUNTR-1
           GO TO 7399
 7555   RETURN
        END
C
C SUBROUTINE FINAL (PRINT OUT RESULTS)
C
              SUBROUTINE FINAL
        COMMON COUNTR,CU(50),Q(50),OD(50),ITEMS,POSI(50),TOTD,PERTOD(50)
        COMMON R,C,NUMPCU,NUMPC,CSPAPC(23)
        COMMON X(12,12,50),COST(12,12),TC(7500),WHCAP,CSPACR(50),NUMUAC
        COMMON SSTAR,NUACPC(23),AW,LS,DOL,CO,CM,V(50),STOVOL,AOS(50)
        COMMON TER1,TER2,TER3,DELTA,KK,CPTDPC(23),COMPLE,ITPCS(50)
        COMMON ITER1(50),ITER3(50),ITER2(50),ITC(50),TPCS,NUMRED
        COMMON SUB1(7500),SUB2(7500),SUB3(7500),VARN(7500),NN,AAA(10)
        REAL ISUM1(50),ISUM2(50),ISUM3(50),ITER1,ITER2,ITER3,ISUMMM(50)
        REAL ISUM(50),ITC,ITPCS
        REAL DELTA,TC,TER1,TER2,TER3,SSTAR,CSPACR,WHCAP,Q
        INTEGER KK,COUNTR,ITEMS,NUMPC,COMPLE,POSI,AAA,SUB1,SUB2,SUB3
        INTEGER NUMRED
```

```
          DO 9913 I=ITTT,NUMPC
             PRINT 9914,I,CPTDPC(I)
 9914         FORMAT(/,T114,I3,T122,F9.7)
 9913     CONTINUE
          PRINT 9917,'***************************************************
     C*******************************************************************
     C****************'
 9917       FORMAT(/,T1,A132)
        ENDIF
C
C PRINT OUT EOQ ALLOCATION AND INDIVIDUAL ITEM
C CONTRIBUTIONS TO TOTAL COST (EOQ)
C
        IF (COUNTR .EQ. 1) THEN
          PRINT 8916,'EOQ ALLOCATION OF ITEMS'
 8916      FORMAT(//,T54,A23)
          NNN=NN-1
          DO 8903 I=1,NNN
             PRINT 8917,'X(',SUB1(I),SUB2(I),SUB3(I),') = ',VARN(I)
 8917         FORMAT(/,A2,I2,I2,I3,A4,F14.7)
 8903      CONTINUE
          PRINT 8933,'***************************************************
     C*******************************************************************
     C****************'
 8933      FORMAT(/,T1,A132)
           PRINT 9918,'INDIVIDUAL ITEM CONTRIBUTIONS TO TOTAL COST'
 9918       FORMAT(/,T44,A43)
           PRINT 9919,'TOTAL COST','INVENTORY COST','TRAVEL COST','ITEM'
 9919       FORMAT(/,1X,T5,A10,T21,A14,T39,A11,T55,A4)
           DO 9921 I=1,ITEMS
              PRINT 9920, ITC(I),ITER3(I),ITPCS(I),I
 9920          FORMAT(/,2X,F15.7,3X,F14.7,3X,F14.7,T56,I3)
 9921      CONTINUE
          PRINT 9931,'***************************************************
     C*******************************************************************
     C****************'
 9931      FORMAT(/,T1,A132)
         ENDIF
       RETURN
       END
C
C
        SUBROUTINE RANK(COI,ITEMS,POSI)
C
C
C LOCAL VARIABLES
       REAL CMAX,COI(50)
       INTEGER STATUS(51)/51*0/,POSI(50),L
C
C
       L=1
 500   K=51
       CMAX=10000
       DO 501 I=1,ITEMS
          IF(COI(I) .LT. CMAX .AND. STATUS(I) .EQ. 0) THEN
```

```
              STATUS(K)=0
              STATUS(I)=1
              CMAX=COI(I)
              POSI(L)=I
              K=I
           ENDIF
 501    CONTINUE
        L=L+1
        IF(L .LE. ITEMS) GO TO 500
        RETURN
        END
//DATA
```

The vita has been removed from
the scanned document