International Journal of Operations Research Vol. 1, No. 1, 37−46 (2004)

Optimal Storage Layout And Order Picking For Warehousing

Chiun-Ming Liu*

Department of Industrial Engineering and Systems Management, Feng Chia University, Taichung 407, Taiwan, R.O.C.

***Abstract*—**In this paper, issues of storage layout and order picking operations problems for warehousing are addressed using optimization techniques. The aim of this study is to develop a dynamic planning system applied for storage layout and order-picking operations problems. The planning system considers dynamic nature of customer order demand, configuration of picking area, and interactive human-machine interface. Heuristic-based optimization technique is utilized to design the planning system. To analyze the dynamic nature of customer order demand, similarity measures among types of items are defined using the entry-order-quantity rule. Based on the characteristics of customer order demand and the configuration of picking area, a zero-one quadratic generalized assignment model is developed. A heuristic procedure is devised to find near-optimal solutions to this problem and coded using Borland C++ computer language. An industrial size application is performed to demonstrate this approach. Results indicate that the developed planning system can be promising for dealing with storage layout and order picking operations problems for warehousing.

***Keywords*—**Warehousing, 0-1 quadratic generalized assignment model, stock location

1. INTRODUCTION

Logistics activities include order processing, storage layout, warehousing, inventory maintenance, transportation and material handling. These activities provide functions for bridging between producers of goods and market consumers, which are separated by time and distance. It has been estimated that logistics operations represent a large portion of a firm's cost dollars. In many local distribution centers, items stored in the slots of racks are picked and distributed according to the huge daily demand order. Most material-handling activities are labor-intensive and repetitive. It follows that the location of stock and the picking operations in the warehouse directly affect the total material-handling cost.

Stock location is the physical layout of items in a distribution center, which meets certain constraints on item location such as security, fire safety, product compatibility, and order picking needs. The main objective of the stock location problem is to minimize the total travel distance or time throughout a distribution center for fulfilling customer demand orders. Several operational considerations have been proposed to improve material-handling efficiency, such as product sequencing, picker zoning, order splitting, and item batching.

Decisions for space determination, storage layout and dock design, warehouse configuration, and stock placement constitute the content of storage systems design problems. For an existing distribution center, the building configuration is always decided and known. Thus, decisions need to be made as to where stock items are to be located and how they should be arranged in the distribution center. These questions are related to the subject of storage systems design focused primarily on determining the location of stock items and order-picking policy within a distribution center.

Another issue related to storage layout problems is the dynamic nature of customer demand order as well as the way to group and sequence products in a warehouse. The traditional approach to storage layout problems within a warehouse ignores dynamic nature of customer demand orders. The demand for items always varies dramatically with seasons. When considering the dynamic nature of customer demand orders, the manager in a distribution center needs to periodically review the characteristics of order demand and modify the stock location accordingly. In most local distribution centers, tens of thousands of picking lists need to be processed daily. Each picking list always consists of several different items and volume. It items are grouped and sequenced in an efficient way, order-picking time can be saved by avoiding backtracking through aisles. Item grouping and sequencing is the arrangement of items on stock location and on pick- ing lists based on the information of dependencies among products, so that they can be picked in a more efficient way than the common random storage layout. The entry-order-quantity rule can be used for analyzing the dependency of products during a period of time. The entry-order-quantity rule states that the quantity of items on an order can be used to measure similarity for any pair of items. These information provide a basis for measuring dependencies among products.

In this study, we develop a planning system for dealing with storage layout and picking operations problems in logistics management. The planning system consists of formula for the enter-order-quantity rule, a zero-one quadratic generalized assignment model and the heuristic procedure for stock location. The developed zero-one quadratic generalized assignment model is formulated

* Corresponding author's email: cmliu@fcu.edu.tw

using a similarity measure between types of items, throughput-to-storage ratio, and distance between storage location and distance from input/output point to storage location. A case study with real-world data collected from a local distribution center is implemented to evaluate the performance of this framework.

The rest of the paper is organized as follows. In section 2, we extensively reviewed past related research in stock location and picking. In section 3, formula for similarity measures are defined and a zero-one quadratic generalized assignment model is presented for storage layout problems. Then, a novel heuristics is explored for solving the zero-one quadratic generalized assignment problem in Section 4. An industrial size application is performed in Section 5. Concluding remarks and areas of improvement are given in Section 6.

2. LITERATURE REVIEW

Ashayeri and Gelders (1985) addressed aggregate design issues using the entry-order-quantity rule and proposed an optimal model for warehouse design. Rosenwein (1994) applied cluster analysis, based on the measure of similarity, to locate items within a warehouse. His result showed the potential benefits of the approach. Dichtl and Beeskow (1980) applied a multi-dimensional scaling method for allocating commodities in a warehouse.

The practical approach to storage systems design problems mainly considers the criteria of where stock items are to be located and how they should be arranged in the distribution center. Wilson (1977) suggested the use of order quantity and product popularity criteria for determining the location of stock items within a distribution center. Ballou (1999) pointed out that the design of stock location can be based on complementarity, compatibility, popularity, and size criteria. Heskeett (1963) combined popularity and size criteria into a cubic-per-order index and applied for warehouse design. Kallina and Lynn (1976) showed that the cubic-per-order index rule can help better stock location. Malmborg and Krishnakumar (1988) modified the cubic-per-order index for designing conventional warehouse with dual command controls. Joseph, Roll and Rosenblatt (1980) applied facility layout technique as well as some stock location policies for internal layout design of a warehouse. Davies et al. (1983) compared four stock location strategies, including alphanumeric placement, fast and often placement, frequency placement, and selection density factor placement. Their results showed that selection density factor placement produced the lowest average distance and time per picking trip. The selection density factor is the ratio of selections per year to the required storage volume in cubic feet. Rosenblatt and Roll (1984) utilized several stock location policies for warehouse design. Harmatuck (1976) compared two approaches for the design of stock location and concluded that the stock location using a throughput-based approach performed better. Francis et al. (1992) considered four storage location policies, that is, dedicated storage, randomized storage, class-based dedicated storage, and shared storage, for determining the assignment of items to storage locations. Due to the dynamic nature of customer demand in most local distribution centers, the class-based dedicated storage policy might provide better design for stock location. Goetshalckx and Ratliff (1990) proposed shared storage policy based on the duration of stay for stock location problems. The shared storage can recognized and take advantage of the inherent differences in lengths of time that individual items remain in storage.

A number of mathematical models for storage layout and order picking operations problems can be found in literature. Francis et al. (1992) presented some mathematical models for determining the size of the storage system and assigning items to storage locations. Ballou (1967) formulated a linear programming model to a similar problem involving reserve storage and order picking areas. Malette and Francis (1972) applied a generalized assignment model to optimal facility layout considering the material-handling cost. Jarvis and McDowell (1991) developed a stochastic model for locating products in an order picking warehouse. Malmborg and Deutch (1988) constructed a stock location model in which the inventory level and cost were considered. Liu (1999) presented a clustering model and developed a closed-form solution for improving stock location and picking operations for a distribution center. Their results showed that the use of clustering techniques as well as mathematical models in solving stock location and order picking problems is quite promising. However, further efforts should be concerned with the investigation of adequate mathematical programming models that can integrate factors related to the dynamic nature of customer order demand, the configuration of picking area, and the dynamic product flow.

Due to the advent of information technology, the questions related to the subject of storage layout problems can be resolved by the applications of simulation technique. Dangelmaier and Bachers (1986) developed a simulation system for material flow and warehouse design using a simulation software package, SIMULAP. Liu (1999) constructed a simulation model using a visual interactive modeling system, WITNESS, for evaluating stock location policies in a distribution center. Although simulation technique can apply for planning storage layout and order picking in a warehouse, this method is limited to account for the dynamic nature of customer order demand and to optimize the stock location. Hence, the entry-order-quantity rule, heuristic-based optimization technique, computer simulation method, and application development tools should be integrated. The integration of the entry-order-quantity rule, heuristic-based optimization technique, computer simulation software, and application development tools might constitute a dynamic stock layout system for design planning and provides a useful tool for decision-makers.

3. ZERO-ONE QUADRATIC GENERALIZED ASSIGNMENT MODEL

The developed planning system consists of formula for the entry-order-quantity rule, an optimization- based heuristic procedure for assigning items to the slots, a simulation model for providing quantitative measures on proposed solutions, and a human- machine interface for decision making. In the warehousing area, the characteristics of order demand can be described by the order entry-order-quantity rule. The entry-order-quantity rule states that if a certain combination of items appears frequently in one common order or picking list, then the probability to simultaneously select these items in one picking trip can be relatively high. Thus, if this group of items can be located in the adjacent storage locations in a warehouse, then the travel distance for the required picking operations can be shortened. Based on the entry-order-quantity rule, we define the similarity measure for pairs of items as follows. Let S_{ik} denote some similarity measure between items i and k. Based on the entry-item-quantity rule, S_{ik} can be defined as follows.

$$s_{ik} = \frac{1}{M}\sum_{m=1}^{M}\frac{\min\{q_m^i, q_m^k\}}{\max\{q_m^i, q_m^k\}} \quad \text{for} \quad i, k=1,\ldots,K \tag{1}$$

where M is the number of demand orders including both items i and k; K is the number of items to be picked within a warehouse; q_m^i denote the quantity of item i in the m^{th} demand order; and $q^i = (q_1, q_2, \ldots q_M)^i$ is the quantity vector of item i in the M demand orders. According to (1), the defined similarity measure for any pairs of items is the ratio of their common order quantity to the maximal order quantity on the order where the two items are listed. Hence, the similarity coefficient represents the probability that the pair of items could appear on one order or picking list.

The throughput-to-storage ratios represent the popularity of products in a distribution center and influence the layout and size of a warehouse. Different throughput-to-storage ratios may be specified for different classes of items. A storage-retrieval rule based on throughput-to-storage ratio states that the item with the largest throughput-to-storage ratio should be assigned to storage locations nearest the outbound area. This rule has been shown to substantially reduce the average trip time. The term throughput is basically used as a measure of the number of storages and retrievals performed per time period for one product in a distribution center. Throughput can also be represented as a measure of the activity or the dynamic nature of storage. The storage size for one product depends on the number of storage locations required. Both the storage capacity and the throughput capacity for a product in a distribution center are also influenced by the layout used.

The design of stock location based on throughput-to-storage ratio can reflect the differences in activity levels and storage requirements among products to be stored. Let T_i denote the throughput per unit time for item i, S_i denote the storage requirement for item i, and t_i denote the throughput-to-storage ratio for item i. Then the throughput-to-storage ratio can be defined as follows.

$$t_i = \frac{T_i}{S_i} \quad \text{for} \quad i=1,\ldots,K \tag{2}$$

According to (2), a larger t_i value implies a greater popularity for item i. Those items with larger throughput-to-storage ratios should be allocated to the storage locations near the outbound area.

The configuration of picking area in a warehouse can be characterized by the travel distance between paired slots and the relative distance of each slot to the input/output point. Let d_{jl} denote the travel distance in meters between slots j and l along the picking route within a warehouse. Also, let r_j denote the relative distance of slot j to the input/output point in a warehouse.

The proposed optimization model considers the characteristics of order demand, the configuration of picking area, and the dynamic product flow. To formulate the stock location problem, we define a binary variable x_{ij} with 1 if item i is assigned to slot j, and 0 otherwise. For the design of storage systems to be feasible, we assume that there are a sufficient number of storage locations in order to dedicate slots to items. The criterion applied in this study is to minimize some function of the distance traveled to pick the assigned items. Hence, a mathematical programming model for the stock location problem may be stated as follows.

$$\text{Minimize } z = \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{P}\sum_{k=1}^{K}\sum_{l=1}^{P} t_i s_{ik} d_{jl} x_{ij} x_{kl} + \sum_{i=1}^{K}\sum_{j=1}^{P} t_i r_j x_{ij} \tag{3}$$

subject to

$$\sum_{i=1}^{K} x_{ij} = 1 \qquad j = 1,\ldots,P \tag{4}$$

$$\sum_{j=1}^{P} x_{ij} = S_i \qquad i = 1,\ldots,K \tag{5}$$

$$x_{ij} = 0,1 \quad i = 1,\ldots,K \quad j = 1,\ldots,P \tag{6}$$

where K is the number of items to be assigned; P is the number of slots available; S_i is the storage requirement for item i, $\sum_{i=1}^{K} S_i \le P$; and $K \le P$.

The objective function (3) gives the expected distance required to perform the required order-picking operations during a time period. In particular, if some item i is assigned to slot j, then it takes r_j distance units to travel from the input/output point to slot j. Since the total number of slots for item i equals S_i, the probability of the picking trip being to slot j is $1/S_i$ for those slots assigned to item i. The total number of picking trips performed per time unit for item i equals T_i. Hence, the expected distance required to travel from the input/output point and slot j is given by the product of

$t_i = T_i/S_i$ and $r_j x_{ij}$. Furthermore, since an order or picking list usually contains several different items, it is possible for a selector to travel from slot j to some other slots during the picking trip. The similarity measure s_{ik} between items i and k gives the probability that a selector travels a distance d_{jl} from slot j to slot l. Thus, the expected distance required to travel from slot j to slot l is given by the product of $t_i = T_i/S_i$ and $s_{ik} d_{jl} x_{ij} x_{kl}$. Summing over all items and slots yields the total expected distance required to perform the picking operations during a time period. Constraint (4) ensures that only one item is assigned to slot j. Constraint (5) ensures that the number of slots assigned to item i equals S_i. Constraint (6) restricts the variable values as zero or one.

The developed optimization model for the storage systems design problem is one type of 0-1 quadratic generalized assignment problems with a nonlinear objective function and linear constraints. Padberg and Rijal (1996) pointed out that this type of problems is NP-hard and suggested the applications of linearlization techniques to streamline the solution methods. Kaku and Thompson (1991) enumerated four types of linearlization approaches for the quadratic problem and concluded that although the linearlization approach could improve the efficiency of solution methods, the resulting core requirements became too large for the mixed integer code they used as the problem size increases. A typical size of problem in this study might involve 17 item types and 52 slots, leading to a formulation having about 781,456 binary variables and 69 structural constraints. The resulting quadratic multi-assignment problem can be formidable to solve to optimality, if the linearlization approach is applied. Hence, a novel heuristics is explored as follows.

4. DEVELOPMENT OF HEURISTIC SOLUTION

The proposed heuristic procedures are based on the developed quadratic multi-assignment model. Due to a special property of the developed model, it is possible to solve this 0-1 quadratic multi- assignment problem without having to utilize the linearlization approach and the standard quadratic problem algorithms. The idea comes from the fact that if there is no existence of similarity among all items, we can set $s_{ik} = 0$, $\forall\ i, k$. Then the problem in (3) accordingly reduces to the linear 0-1 multi-assignment problem.

$$\text{Minimize } z = \sum_{i=1}^{K} \sum_{j=1}^{P} t_i r_j x_{ij} \tag{7}$$

Subject to (4), (5), and (6). (8)

This linear 0-1 multi-assignment problem can easily be solved without appealing to one of the standard algorithms. The motivation is to put the item with the largest throughput-to-storage ratio in the slots with the smallest average travel distance, put the item with the next largest ratio in the slots with the next smallest travel distance, and so on. Below, we present an effective ranking procedure that is capable of finding an optimal multi-assignment for the problem (7).

The developed heuristics procedure includes a ranking phase, a clustering phase, and an interchanging phase. The motivation for the ranking phase is to put the item with the largest throughput-to-storage ratio in the slots with the smallest average travel distance, put the item with the next largest ratio in the slots with the next smallest travel distance, and so on. Below, we present an effective ranking phase that is capable of finding an initial assignment for the problem.

Ranking Phase

Step 1. Number the items according to the throughput-to-storage ratios, such that $t_1 \ge t_2 \ge \cdots \ge t_k$.

Step 2. Number the slots according to the travel distances, such that $r_1 \le r_2 \le \cdots \le r_p$.

Step 3. Assign item 1 to the r_1 to r_{S_1} slots; assign item 2 to the r_{S_1+1} to $r_{S_1+S_2}$; and so on. S_{UV}

Step 4. Compute the total travel distance as the upper bound for the problem.

Next, we observe that if the factor of throughput-to-storage ratios is ignored for all items in the problem, then the objective function (3) reduces to:

$$\text{Minimize } z = \sum_{i=1}^{K} \sum_{j=1}^{P} \sum_{k=1}^{K} \sum_{l=1}^{P} s_{ik} d_{jl} x_{ij} x_{kl} \tag{9}$$

The solution to the resulting quadratic multi- assignment problem requires those items shared with higher similarity measures should be allocated in the adjacent slots. Hence, we need to develop a clustering procedure to obtain the grouping structure of items by the associated similarity information. The clustering procedure is presented as follows.

Then, the solution to the remaining problem requires those items shared with higher similarity measures should be allocated in the adjacent slots. Hence, we can develop a clustering phase to obtain the grouping structure of items by the associated similarity information. The clustering phase is presented as follows.

Clustering Phase

Step 1. Start with *K* groups, each containing a single item, and a *K*×*K* symmetric matrix of similarities $S = \{s_{ik}\}$.

Step 2. Search the largest similarity in the similarity matrix for the nearest pair of groups. Let the similarity measure between most similar groups *U* and *V* be s_{UV}.

Step 3. Merge groups *U* and *V*. Label the newly formed cluster *(UV)*. Update the entries in the similarity matrix by (i) deleting the rows and columns corresponding to groups *U* and *V*, and (ii) adding a row and column giving the similarities between group *(UV)* and the remaining groups.

Step 4. Repeat Steps 2 and 3 a total of *K-1* times. Record and identify the groups that are merged and the levels at which the merging takes place.

The obtained grouping structure of items is then applied to modify the stock location assignment. The idea is to move items allocated in the distanced slots forward to their most similar item. The proposed interchanging phase begins with the first level in the grouping structure, in which items in this group are considered to reshuffle. If those items in one level are already allocated in the adjacent slots, then we proceed to the next level of the grouping structure. Otherwise, those items that are allocated in the distanced slots are considered to move forward to the target items that are allocated nearest the outbound area. The proposed interchanging phase is stated as follows.

Interchanging Phase

Step 1. Begin with the first level of the grouping structure. If all items in the same group are already allocated in the adjacent slots, then go to Step 5. Otherwise, proceed to Step 2.

Step 2. Denote the item with largest throughput- to-storage ratio as the target item and the remaining non-adjacent items as the non-target items. Among the non-target items, select the one with the largest throughput-to-storage ratio as the candidate item. Proceed to Step 3.

Step 3. Move the candidate item forward to the slots next to the target item. Those items that are not in the current level and being currently allocated next to the target item are shifted next to the candidate item. Compute the total expected travel distance. If the computed total expected travel distance is smaller than the previous one, then proceed to Step 4. Otherwise, restore to the current stock location assignment. Go to Step 5.

Step 4. If there exists some non-target item in the same group that can be served as a candidate for rearrangement, select one as the candidate item and return to Step 3. Otherwise, proceed to Step 5.

Step 5. If all of the levels in the grouping structure are examined, stop. Otherwise, enter the next level of the grouping structure and return to Step 2.

The developed heuristic procedures, which include the ranking phase, the clustering phase, and the interchanging phase, were coded in Borland C++ computer language. The compiled program was run on a Pentium II 266 PC with 64 MB RAM

5. IMPLEMENTATION

To conform to the low-volume, multi-items demand market, the distribution center of a multi-branched trading company provides an open-package area for storage and picking of less-than-case-lot-quantity items. Small items that are distributed in less-than-case-lot quantities are stored in gravity-flow racks. Configuration of the open-package area and existing stock location are shown in Figure 1. Fifty-two slots are available in the warehouse. The warehousing situation assumes that an order selector travels between the order-picking area and an input/output point. With a picking list, the order selector travels to the retrieval location, retrieves the entire items on the picking list, and returns to the input/output point. The selected items are then deposited onto a conveyor where it is routed to the outbound area. It is assumed that replenishment of stock items within the order-picking area occurs separately from order picking operations.

In order to improve the material-handling efficiency, similarity measures and throughput- to-storage ratios for each item type were calculated, respectively, based on (1) and (2). The data used for calculation were collected from 17 item types stored in the open-package area. Table 1 shows the similarity measure, the storage requirements, and the throughput-to-storage ratio for each item type. The configuration of picking area was characterized using the travel distance between each paired slots and the relative distance of each slot to the input/output point. Fifty-two slots are available in the warehouse. The picking path is designed using a Z-type traveling route. Based on the definition mentioned before, a pair-wise travel distance matrix and the relative distance to the input/output point for each slot can be constructed accordingly..

Using the similarity measure, the throughput- to-storage ratio, the travel distance, and the relative distance as input data for the problem, the proposed heuristic was utilized to find an optimal stock location assignment. Firstly, the ranking phase was applied to obtain an initial stock location assignment. The initial stock location assignment obtained from the ranking phase is shown in Table 2. The total expected travel distance is 58,498 meters. Then, the clustering phase was used to group items level by level. The hierarchical grouping structure of items is shown in Table 3. Finally, we applied the interchanging phase to modify the stock location assignment obtained from the ranking phase. The phase began with the first level of the grouping structure and proceeded to the last level. The detailed procedure for this phase is shown in Table 4. The final stock location assignment for the storage systems design problem is shown in Figure 2. The total obtained travel distance is 57,523 meters.

From the obtained stock location, we can see that items with higher throughput-to-storage ratios are allocated close to the input/output point. Also those items with higher similarity are allocated in the slots adjacent to each other. Hence the selector can easily search and retrieve the items.

The WITNESS simulator was used to construct the simulation model for the storage system and to demonstrate the performance of the achieved stock location from the heuristic. Figure 3 displays the developed simulation model for the achieved stock location. Given a set of customer orders that are generated according to the resulting pattern from Table 1, simulation experiments were designed to compare the average picking time per order to retrieve. Two order-picking alternatives, including order-picking with the proposed sequencing rule and order-picking with the first-come-first serve rule were adopted in the simulation study. Results from simulation experiments are shown in Table 5, which suggests that the average picking time per order to retrieve is shorter for the achieved stock location under the proposed order-picking sequencing rule. Furthermore, in order to compare the computational efficiency of the developed heuristic procedure, Table 6 displays results of computational efforts for the proposed heuristic procedure and the well-known optimization package, the AMPL-CPLEX software package (1997), for solving 10 test problems. Results in Table 6 suggest that the heuristic procedure gives a better solution when solving smaller quadratic multi-assignment problems. For larger problems, the AMPL-CPLEX package cannot give a solution, while the developed heuristics can provide a good solution within a very short CPU time.

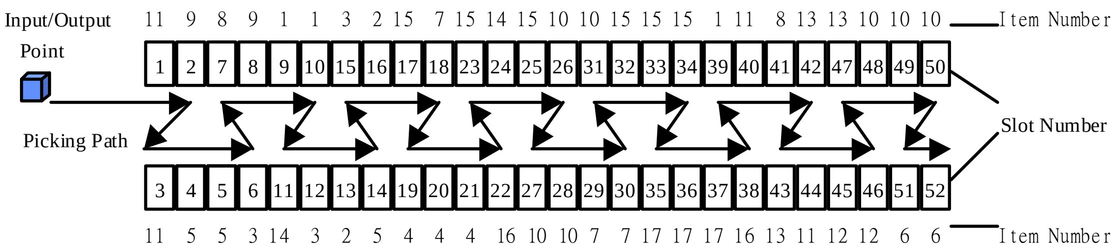

Figure 1. Configuration and existing stock location of open-package area.

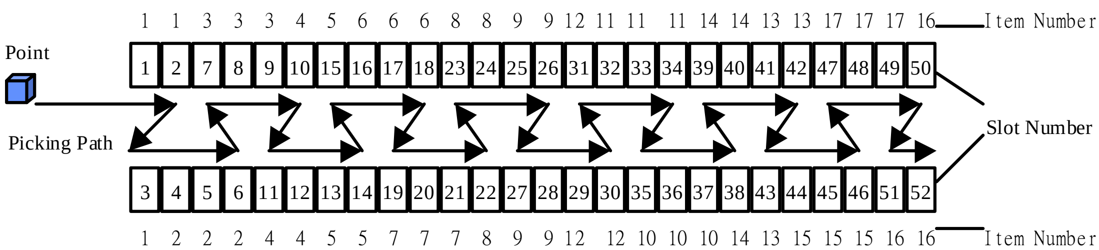

Figure 2. The achieved stock location using the proposed approach.

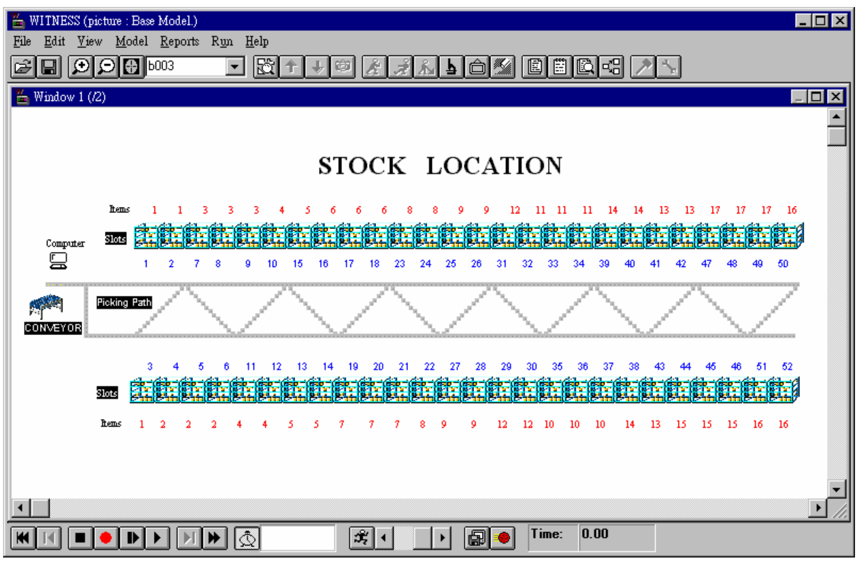

Figure 3. A WITNESS simulation model for the achieved stock location.

Table 1. Similarity measure, s_{ik}, throughput, T_i, storage requirement, S_i, and throughput-to-storage ratio, t_i, of seventeen item types

s_{ik}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1.0	.06	.05	.05	.02	.04	.03	.02	.03	.03	.01	.02	.01	.01	.00	.00	.00
2	.06	1.0	.49	.56	.50	.31	.31	.24	.16	.05	.15	.13	.07	.08	.05	.00	.01
3	.05	.49	1.0	.64	.60	.42	.40	.30	.14	.03	.22	.11	.11	.11	.07	.00	.01
4	.05	.56	.64	1.0	.63	.46	.50	.35	.25	.07	.21	.20	.12	.12	.07	.00	.03
5	.02	.50	.60	.63	1.0	.38	.46	.32	.14	.04	.24	.12	.12	.14	.08	.00	.02
6	.04	.31	.42	.46	.38	1.0	.41	.30	.16	.06	.23	.17	.12	.12	.08	.00	.03
7	.03	.31	.40	.50	.46	.41	1.0	.63	.32	06	.40	.25	.21	.20	.12	.00	.03
8	.02	.24	.30	.35	.32	.30	.63	1.0	.17	.07	.47	.35	.20	.21	.15	.00	.03
9	.03	.16	.14	.25	.14	.16	.32	.17	1.0	.00	.27	.46	.13	.14	.10	.00	.02
10	.03	.05	.03	.07	.04	.06	.06	.07	.00	1.0	.04	.00	.00	.00	.00	.00	.00
11	.01	.15	.22	.21	.24	.23	.40	.47	.27	.04	1.0	.38	.37	.32	.27	.00	.05
12	.02	.13	.11	.20	.12	.17	.25	.35	.46	.00	.38	1.0	.01	.17	.14	.00	.08
13	.01	.07	.11	.12	.12	.12	.21	.20	.13	.00	.37	.01	1.0	.52	.33	.00	.05
14	.01	.08	.11	.12	.14	.12	.20	.21	.14	.00	.32	.17	.52	1.0	.42	.00	.05
15	.00	.05	.07	.07	.08	.08	.12	.15	.10	.00	.27	.14	.33	.42	1.0	.00	.13
16	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	1.0	.00
17	.00	.01	.01	.03	.02	.03	.03	.03	.02	.00	.05	.08	.05	.05	.13	.00	1.0
T_i	514	92	58	57	52	48	31	22	21	18	14	12	8	7	5	5	2
S_i	3	3	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3
t_i	171.3	30.7	19.3	19	17.3	16	10.3	7.3	5.3	6	4.7	4	2.7	2.3	1.7	1.7	.7

Table 2. The stock location assignment obtained from the ranking phase

Slot No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Item Type	1	1	1	2	2	2	3	3	3	4	4	4	5
Slot No	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
Item Type	5	5	6	6	6	7	7	7	8	8	8	10	10
Slot No	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Item Type	10	9	9	9	9	11	11	11	12	12	12	13	13
Slot No	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**
Item Type	13	14	14	14	15	15	15	16	16	16	17	17	17

Table 3. The grouping structure obtained from the clustering phase

Level No.	Clustering Process
1	Merge {3,4} and {5}.
2	Merge {7} and {8}.
3	Merge {3,4,5} and {2}.
4	Merge {13} and {14}.
5	Merge {3,4,5,2} and {7,8}.
6	Merge {3,4,5,2,7,8} and {11}.
7	Merge {3,4,5,2,7,8,11} and {6}.
8	Merge {9} and {12}.
9	Merge {13,14} and {15}.
10	Merge {3,4,5,2,7,8,11,6} and {9,12}.
11	Merge {3,4,5,2,7,8,11,6,9,12} and {13,14,15}.
12	Merge {3,4,5,2,7,8,11,6,9,12,13,14,15} and {17}.
13	Merge {3,4,5,2,7,8,11,6,9,12,13,14,15,17} and {10}.
14	Merge {3,4,5,2,7,8,11,6,9,12,13,14,15,17,10} and {1}.
15	Merge {3,4,5,2,7,8,11,6,9,12,13,14,15,17,10,1} and {16}.

Table 4. The detailed procedure for the interchanging phase

Loop No.	Description	Objective Value	Is Improve?
0	Interchange {3} with {4}.	58,680	No
1	Interchange {3,4} with {5}.	58,680	No
2	Interchange {7} with {8}.	58,869	No
3	Interchange {3,4,5} with {2}.	58,869	No
4	Interchange {13} with {14}.	58,491	Yes
5	Interchange {3,4,5,2} with {7,8}.	59,495	No
6	Interchange {3,4,5,2,7,8} with {11}.	58,240	Yes
7	Interchange {3,4,5,2,7,8,11} with {6}.	59,677	No
8	Interchange {9} with {12}.	58,422	No
9	Interchange {13,14} with {15}.	58,429	No
10	Interchange {3,4,5,2,7,8,11,6} with {9,12}.	57,580	Yes
11	Interchange {3,4,5,2,7,8,11,6,9,12} with {13,14,15}.	64,631	No
12	Interchange {3,4,5,2,7,8,11,6,9,12,13,14,15} with {17}.	57,523	Yes
13	Interchange {3,4,5,2,7,8,11,6,9,12,13,14,15,17} with {10}.	61,343	No
14	Interchange {3,4,5,2,7,8,11,6,9,12,13,14,15,17,10} with {1}.	152,127	No
15	Interchange {3,4,5,2,7,8,11,6,9,12,13,14,15,17,10,1} with {16}.	60,319	No

Table 5. Results of simulation experiments for pickings in the existing stock location assignment and the obtained stock location assignment, measured by average time in minute per order

Order Processing Type	Existing Stock Location	Obtained Stock Location
Sequencing Customer Orders	13.6	11.2
First-Come-First-Serve Orders	14.0	13.9

Table 6. A comparison of CPU times for the AMPL-CPLEX package and the proposed heuristics

Prob. No.	No. of Items	No. of Slots	AMPL- CPLEX			The Heuristics			Gap between Two Objective Values in %
			No. of Branch	CPU in sec.	Obj. Value	No. of Loops	CPU in sec.	Obj. Value	
1	6	9	0	0.028	565	5	0.000	562	0.5
2	8	10	0	0.068	654	7	0.011	652	0.3
3	9	10	0	0.108	621	8	0.009	620	0.2
4	10	15	*	*	*	9	0.005	1769	NA
5	12	18	*	*	*	11	0.317	2467	NA
6	15	30	*	*	*	14	0.428	17952	NA
7	16	45	*	*	*	15	2.865	46221	NA
8	17	52	*	*	*	16	0.182	57523	NA
9	20	52	*	*	*	19	0.465	103142	NA
10	30	52	*	*	*	29	0.360	101598	NA

6. CONCLUSIONS

Decisions related to where stock items are to be located and how they should be arranged in the warehouse play an important role in a distribution center. Due to the dynamic nature of customer order demand, we develop dynamic stock layout systems for storage layout and order picking operations problems. This planning system considers the dynamic nature of customer demand, the throughput-to-storage ratio, and the configuration of picking area in a warehouse and consists of formula for the entry-order-quantity rule, and a novel heuristic procedure. The devised heuristics can be efficiently used for solving the stock location problem. Results indicate that the developed approach can account for the dynamic nature and provide potential benefits for the storage layout and order picking operations problem.

REFERENCES

1. Ashayeri, J. and Gelders, L.F. (1985). Warehouse design optimization. *European Journal of Operational Research*, 21: 285-294.
2. Ballou, R.H. (1967). Improving the physical layout of merchandise in warehouses. *Journal of Marketing*, 31: 60-64.
3. Ballou, R.H. (1999). *Business logistics management. 4th Edition*, Prentice-Hall International Inc., Englewood Cliffs, N.J.
4. Dangelmaier, W. and Bachers, R. (1986). SIMULAP: a simulation system for material flow and warehouse design. *Material Flow*, 3: 207-214.
5. Davies, A.L., Gabbard, M.C., and Reinhold, E.F. (1983). Storage method saves space and labor in open-package-area picking operations. *Industrial Engineering*, 68-74.
6. Dicht, E. and Beeskow, W. (1980). Optimal allocation of commodities in storehouse by means of multi-dimensional scaling. *Zeitschrift fur Operations Research*, 24: B51-B64.
7. Francis, R.L., McGinis, L.F., and White, J.A. (1992). *Facility layout and location: an analytical approach. 2nd Edition*, Prentice-Hall International Inc., Englewood Cliffs, N.J.
8. Goetshalckx, M. and Ratliff, H.D. (1990). Shared storage policies based on the duration of stay. *Management Science*, 36: 1120-1132.
9. Harmatuck, D.J. (1976). A comparison for two approaches to stock location. *Logistics and Transportation Review*, 12: 282-284.
10. Heskeet, J.L. (1963). Cubic-per-order index: a key to warehouse stock location. *Transportation and Distribution Management*, 3: 27-31.
11. Jarvis, J.M. and McDowell, E.D. (1991). Optimal product layout in an order picking warehouse. *IIE Transactions*, 23: 92-102.
12. Joseph, B., Roll, Y., and Rosenblatt, M.J. (1980). Internal layout design of a warehouse. *AIIE Transactions*, 24: 317-322.
13. Kaku, B.K. and Thompson, G.L. (1991). An exact algorithm for the general quadratic assignment problem. *European Journal of Operational Research*, 23: 382-390.
14. Kallina, C. and Lynn, J. (1976). Application of the cube-per-order index rule for stock location in a distribution warehouse. *Interfaces*, 7: 37-46.
15. Malette, A.J. and Francis, R.L. (1972). A generalized assignment approach to optimal facility layout. *AIIE Transactions*, 4: 144-147.
16. Malmborg, C.J. and Deutsch, S.T. (1988). A stock location model for dual address order picking systems. *IIE Transactions*, 20: 44-52.
17. Malmborg, C.J. and Krishnakumar, B. (1987). On the optimality of the cube per order index for conventional warehouse with dual command controls. *Material Flow*, 4: 169-176.
18. Liu, C.M. (1999). Clustering techniques for stock location and order-picking in a distribution center. *Computer and Operations Research*, 26: 989-1002.
19. Padberg, M.W. and Rijal, M.P. (1996). *Location, scheduling, design and integer programming*. Kluwer Academic

Publishers, London.
20. Robert, F., David, M.G., and Brian, W.K. (1997). *AMPL : a modeling language for mathematical programming*. Duxbury Press.
21. Rosenblatt, M.J. and Roll, Y. (1984). Warehouse design with storage policy considerations. *International Journal of Production Research*, 22: 809-821.
22. Rosenwein, M.B. (1994). An application of cluster analysis to the problem of locating items within a warehouse. *IIE Transactions*, 26: 101-103.
23. Wilson, H.G. (1977). Order quantity, product popularity, and the location of stock in the warehouse. AIIE Transactions, 9: 230-237.