

# Space Optimization for Warehousing Problem: A Methodology for Decision Support System

A. L. N. Murthy\*  
Indian Statistical Institute

(Received: August 7, 2008 / Revised: August 25, 2009 / Accepted: April 17, 2012)

---

## ABSTRACT

This article presents a way of tackling a special class of space optimization problems that arise in a number of practical applications in industry and elsewhere. It presents an elegant solution to a problem that was considered by (Das, 2005) in optimizing storage space in warehouse of a footwear manufacturing company. In (Das, 2005), the problem was formulated as a nonlinear programming problem. In this article, it is shown that the problem can be formulated as a generalized transportation problem which is a special case of generalized network flow problems. Further, an elegant scheme is devised to handle the dynamic situation of warehousing problem which can be easily translated into a decision support system for the warehouse management system. Also, the article points out certain obscurities and gaps in (Das, 2005).

Keywords: Bin Packing, Optimization, Warehousing, Decision Support System

\* Corresponding Author, E-mail: simhaaln@rediffmail.com

---

## 1. INTRODUCTION

Storage optimization problems are a commonplace in warehouses, provisional stores, footwear shops, ship container loading and the like. These problems are known as bin packing problems in the literature. For a review of these and related problems such as strip packing and cutting stock problems, see (Dyckhoff, 1990; Dyckhoff and Finke, 1992; Hinxman, 1980; weeney and Paternoster, 1992). Optimization techniques play a crucial role in space management problems. Establishments dealing with such problems day in and day out require good decision support systems backed by mathematical programming techniques with efficient formulations and solutions. This article considers a special class of storage optimization problems that deal with storing cuboidal boxes in cuboidal compartments. Das (2005) considered this type of problem while addressing a space optimization problem for footwear manufacturing company. In (Das, 2005), the problem is given a solution using a heuristic approach to solve a part of the problem and the remaining part of the problem is only presented as a nonlinear program-

ming problem without a solution methodology. This article presents a complete solution to this problem. Identifying the right decision variables and efficient modeling in handling practical problems are very crucial in optimization. It is shown that the part of the problem presented as nonlinear programming problem in (Das, 2005) can indeed be formulated as a *generalized transportation problem* which is a special case of generalized network flow problems (Murty, 1992). Routine warehouse management problems involve continual storage and retrieval (issues) of goods from the warehouse. Therefore, a scheme is required to handle the dynamic storage and retrieval of goods optimally from the warehouse is in need. This article presents a clever solution to this dynamic problem and recommends a decision support system for the warehouse management problem.

The organization of this article is as follows. Section 2 describes the general problem. Section 3 reviews the heuristic and the nonlinear programming formulation presented in (Das, 2005). Section 4 introduces the generalized transportation problem. A complete solution methodology for the storage optimization problem is

presented using the generalized transportation problem formulation. Section 5 presents a procedure to handle the dynamic nature of the problem. Section 6 provides a guideline for development of a decision support system for the warehouse management system. Section 7 summarizes the article.

## 2. PROBLEM DESCRIPTION

A number of practical applications involve storing objects in compartments in which objects as well as the compartments are cuboidal in shape. Applications include storing shoe boxes in racks in warehouses of footwear manufacturing companies, medicines in cartons, and so on. Typically, the problem involves determination of which boxes to be stored in which compartments. One can consider two objectives in this case: (i) minimize the volume of racks to be used or (ii) minimize the number of racks to be used. In other words, the problem can be stated as: given the boxes to be stored along with their sizes and given the compartments available for storing along with their sizes, determine a way of storing boxes in compartments so as minimize either the volume of the compartments to be used or the number of compartments to be used. However, in practical applications, some natural constraints arise from the view point of operational conveniences. For instance, one such constraint is not to allow boxes of different sizes to be stored in the same compartment. In other words, each compartment can store only one size of boxes. Another type of constraint could be to store the boxes so that one side of the box is always in the vertical direction. For examples, when television sets are placed in racks one must ensure that the base of the television must rest on the horizontal surface. This article considers only the class of problems in which cuboidal boxes are stored in cuboidal compartments with the two types of constraints described above. The mathematical description of the problem is given below.

Consider  $m$  types of cuboidal boxes with dimensions of  $i^{\text{th}}$  type being  $l_i \times b_i \times h_i$ , where  $l_i$ ,  $b_i$  and  $h_i$  are length, breadth and height of the  $i^{\text{th}}$  type box respectively,  $i = 1, 2, \dots, m$ . Consider  $n$  types of cuboidal compartments with dimensions of  $j^{\text{th}}$  type being  $L_j \times B_j \times H_j$ , where  $L_j$ ,  $B_j$  and  $H_j$  are length, breadth and height of the  $j^{\text{th}}$  type compartment respectively,  $j = 1, 2, \dots, n$ . Let  $V_j$  be the volume of the  $j^{\text{th}}$  compartment. There are  $a_i$  boxes of type  $i$ ,  $i = 1, 2, \dots, m$ , and  $d_j$  compartments of type  $j$ ,  $j = 1, 2, \dots, n$ . Consider the problem of arranging the boxes in the compartments subject to the following constraints.

**Constraint 1:** Each compartment can be used only for one type of box. That is, two different size boxes cannot be stored in a single compartment.

**Constraint 2:** The placement of any box in any com-

partment should be such that the side representing the height of the box should be parallel to the side of the compartment representing the height of the compartment, and the base sides of the boxes should be either parallel or vertical to the base sides of the compartment.

Let  $k_j$  be the number of compartments of type  $j$  to be used for storing,  $j = 1, 2, \dots, n$ . Consider the following optimization problems.

**Problem 1:** Minimize  $\sum k_j V_j$  such that Constraint 1 and Constraint 2 are satisfied. Here  $\sum k_j V_j$  is the total volume of the compartments used.

**Problem 2:** Minimize  $\sum k_j$  such that Constraint 1 and Constraint 2 are satisfied. Here  $\sum k_j$  is the total number of the compartments used.

**Remark 1:** The problem described here can be used in a very general setup. Consider the problem of storing footwear boxes in a warehouse and consider the position of compartments in the warehouse at a given point in time. Supposing the warehouse gets new consignment of boxes to be stored. The problem of storing the new boxes without disturbing the existing position of the storage in the warehouse can also be tackled easily by updating the  $a_i$ s and  $d_j$ s. It is understood that this scenario has been referred to as dynamic situation in (Das, 2005) for which a solution is being sought with nonlinear programming formulation. This article provides a complete solution for this problem by formulating the same as generalized transportation problem which is a special case of generalized network flow problem.

## 3. REVIEW OF THE HEURISTIC

The heuristic approach suggested in (Das, 2005) to solve the problem described Section 2 needs a review as there are certain obscurities and gaps in the presentation of the algorithm (refer to section 4.2 of (Das, 2005)).

Consider Problem 1 described in Section 2. From the presentation of the algorithm in (Das, 2005), it is understood that the algorithm is meant for the following problem: Given a particular type of box (designated by the dimensions  $X_1 \times Y_1 \times Z_1$ ), and given a set of compartments (the algorithm is described with five compartments in mind), determine the storage pattern for the given compartments with an aim to achieve the maximum space utilization. The algorithm does not spell out as to how the boxes and compartments are to be picked up to solve Problem 1. Perhaps what Das had in mind was the following: pick up any box type and use the algorithm to assign all the boxes of this type by picking up the compartments arbitrarily; pick up the next box type and repeat the procedure with the leftover compartments; and continue this process until all boxes of all types are exhausted. If this is the approach, then it

must be remarked that the heuristic is weak and has no proper scientific logic or intuition.

Furthermore, there are some obscurities in the presentation of the algorithm in (Das, 2005). Some of them are pointed below (refer to step numbers as described in (Das, 2005)).

**Step 0:** Here  $i$  is referred to as the index of the box but in Step 1 it is referred to as the compartment number which is correct.

**Step 2:** Here it is said that ‘... obtain the optimal solutions ...’ It must be noted that optimal solution need not be unique. The algorithm must spell out what should be done when there is ambiguity.

**Step 3:** Consider the following sentence in this step: ‘The choice of orientation depends upon the number of boxes that can be obtained.’ Perhaps it should read as follows: ‘The choice of orientation depends upon the maximum number of boxes that can be fitted in Sector III.’ Further, the formula for the dimensions of Sector IV in the last sentence of Step 3 should be  $(X - X_1 \cdot n_1^0) \times (Y - X_1 \cdot n_3^0)$  and not  $(X - X_1 \cdot n_1^0) \times (Y - X_1 \cdot n_4^0)$ .

**Step 5:** This step must read as follows: Let  $A$  and  $B$  be the horizontal and vertical dimensions of Sector IV respectively. Set  $X = A$  and  $Y = B$ . If  $\max(X, Y) < X_1$  or  $\min(X, Y) < Y_1$ , go to Step 6, otherwise go to Step 2.

One more obscurity in (Das, 2005) is that  $Z$  and  $Z_1$  are supposed to represent the heights of compartment and box respectively. But for the calculation of Table 1 of (Das, 2005),  $X_1$  is taken as the height of the box while  $Z$  remains the height of the compartment.

Next, the claim in Step 4 of the algorithm claims optimality for the sector with which Step 4 is reached. To examine this aspect, consider the following example. Take the box and compartment sizes as: box dimensions are  $l = 3.5$ ,  $b = 2$  and  $h = 8$ , and the compartment dimensions are  $L = 48$ ,  $B = 24$  and  $H = 24$  (all the dimensions are in inches). In Step 2 of the first iteration, there are three solutions to Problem (I) of (Das, 2005). These are:  $(n_1^0, n_2^0) = (4, 17)$ ,  $(n_1^0, n_2^0) = (8, 10)$  and  $(n_1^0, n_2^0) = (12, 3)$ . Since there is no specific choice, take the first solution as solution to Problem (I), that is,  $(n_1^0, n_2^0) = (4, 17)$ . However, the second problem, Problem (II), has a unique solution in this case, namely,  $(n_3^0, n_4^0) = (4, 5)$ . Consider the following solutions of Problem (I) and Problem (II):  $n_1^0 = 4$ ,  $n_2^0 = 17$ ,  $n_3^0 = 4$ ,  $n_4^0 = 5$ . Since the condition to go to Step 4 is not satisfied, Step 3 is executed. This leads to the four sectors with the following dimensions: Sector I ( $34 \times 14$ ), Sector II ( $14 \times 10$ ), Sector III ( $14 \times 14$ ) and Sector IV ( $34 \times 10$ ). This in turn leads to Step 5, and Step 5 leads to Step 2 with  $X = 34$  and  $Y = 10$ . In the second iteration, solving the two integer-linear program-

ming problems as per Step 2 results in the following solutions:  $n_1^0 = 4$ ,  $n_2^0 = 10$ ,  $n_3^0 = 1$ ,  $n_4^0 = 3$ . Again, the algorithm leads to Step 3 and the sector dimensions are as follows: Sector I ( $20 \times 3.5$ ), Sector II ( $14 \times 6$ ), Sector III ( $14 \times 4$ ) and Sector IV ( $20 \times 6.5$ ). This once again leads to Step 2 with  $X = 20$  and  $Y = 6.5$ . Note that a set of solutions for the two integer-linear programming problems are given by  $n_1^0 = 4$ ,  $n_2^0 = 3$ ,  $n_3^0 = 0$ ,  $n_4^0 = 3$ . Since the condition to go to Step 4 is not met, Step 3 is executed. But this leads to only three sectors as follows: Sector II ( $20 \times 6$ ), Sector III ( $20 \times 0.5$ ) and Sector IV ( $6 \times 6.5$ ). This leads to Step 5, and Step 5 leads to Step 2 with  $X = 6$  and  $Y = 6.5$ . The solutions to Problem (I) and Problem (II) are:  $n_1^0 = 0$ ,  $n_2^0 = 3$ ,  $n_3^0 = 0$ ,  $n_4^0 = 2$ . This leads to Step 4 which asserts that vertical or horizontal arrangement in Sector I will be optimal. However, the definition of Sector I is not given. If one assumes, by default, that the dimension of Sector I is  $(X \times Y)(6 \times 6.5$  in this case), then only 3 boxes can be accommodated either in vertical arrangement or in horizontal arrangement. But it is easy to see in this case that four boxes can be accommodated in mixed fashion (three horizontal and one vertical). Thus, a claim made in the heuristic is not correct, and it may lead to inefficient solutions.

#### 4. THE GENERALIZED TRANSPORTATION PROBLEM (GTP)

This section describes the generalized transportation problem which is useful in solving the class of optimization problems considered in this article. It must be noted that GTP is a special case of generalized network flow problem—a well studied problem in the literature of Operations Research. The network flow problems can be solved efficiently. An implementation of the primal simplex algorithm without the need for computing the inverses can be used to compute the solutions of GTP. Murty (1992) presents a detailed discussion on the treatment of generalized network flow problems.

Let  $W = [w_{ij}]$  be an  $m \times n$  matrix with its entries as nonnegative integers. Let  $a$  and  $d$  be column vectors of dimensions  $m$  and  $n$  respectively with entries as non-negative integers. Let  $u$  be an  $n$ -dimensional column vector with real entries. The generalized transportation problem is defined as follows: find nonnegative integers  $x_{ij}$ ,  $i = 1, 2, \dots, m$ ;  $j = 1, 2, \dots, n$ , so as to

$$\begin{aligned} & \text{Maximize } \sum_j u_j \sum_i x_{ij} \\ & \text{subject to } \sum_j w_{ij} x_{ij} \geq a_i \end{aligned} \quad (1)$$

$$\sum_i x_{ij} \leq d_j. \quad (2)$$

Call the matrix  $W$  the **Weights Matrix** and call its elements as weights. Denote the GTP described above by  $GTP(W, a, d, u)$ .

To see the resemblance with the transportation problem, consider the following transportation problem with  $m$  buyers (of a product) and  $n$  sellers with some special restrictions. Imagine that the sellers hire dedicated trucks for each buyer with specific capacities. That is, assume that any truck of  $j^{\text{th}}$  seller for  $i^{\text{th}}$  buyer will carry exactly  $w_{ij}$  units of the product. Let  $d_j$ , the  $j^{\text{th}}$  coordinate of  $d$ , be the maximum number of trucks that seller can hire. Let  $a$  denote the minimum purchase quantity vector of the buyers. Finally, assume that the owner of the trucks, from whom all the sellers hire the trucks, charges  $u_j$  dollars per truck from seller  $j$ . Define  $x_{ij}$  as the number of trucks used for supplying the product from seller  $j$  to buyer  $i$ . Then, the problem of maximizing the total truck charges subject to meeting the minimum purchase quantities, not exceeding the maximum number of trucks available to each seller, and the trucks carrying exact number of goods ( $w_{ij}$ s), is exactly the GTP described above.

It may be noted that a special case of the GTP in which all the weights are equal to 1 is the usual transportation problem with special cost structure in which the sellers dictate the costs. The rest of this section will present solution methodology for solving Problem 1 and Problem 2 of Section 2.

Note that the difference between Problem 1 and Problem 2 is that the former minimizes the volume of the compartments used while the latter minimizes the total number of compartments used. Both the problems can be solved by simply changing the objective function. The method being proposed here to solve both these problems involves a two stage approach. The first stage involves solving two-dimensional bin packing problems. The second stage involves solving a GTP. Since the first stage involves solving of bin packing problems, a brief review of bin packing problems is presented in the next paragraph.

#### 4.1 Bin Packing Problems

Bin packing problems are well known and well studied problems in the literature of Operations Research. Bin packing problems are NP-hard problems. Numerous articles in the literature have dealt with algorithms, heuristics and metaheuristics for solving the two-dimensional bin packing problems. For details see (Bengtsson, 1982; Chung *et al.*, 1982; Coffman *et al.*, 1998; Coffman *et al.*, 1996; Sweeney and Paternoster, 1992). Solvers (commercial as well as public domain software) are available for solving the type of bin packing problems that need to be solved in Stage 1 while solving Problem 1 and Problem 2. For brevity, the presentation is confined to this type of bin packing problems. Consider two rectangular sheets of dimensions  $L \times B$  and  $l \times b$  in which the sheet with dimensions  $L \times B$  is bigger than the other. The two-dimensional bin packing problem (also popularly known as the two dimensional cutting stock problem) seeks a way of cutting maximum number of sheets of smaller size (with dimensions  $l \times b$ ) from the bigger sheet.

This problem will be denoted by  $BPP(L, B, l, b)$ . A solution to this problem simply specifies a cutting pattern (the layout) on the bigger sheet that gives maximum number of smaller sheets. As an illustration, consider the problem encountered in the earlier section in which maximum number of sheets of dimensions  $3.5 \times 2$  are to be cut from a sheet of size  $6 \times 6.5$ . According to the notation introduced, this problem is  $BPP(6, 6.5, 3.5, 2)$ .

#### 4.2 Solution to Problems 1 and 2

The stage is now set for presenting the solution methodology for solving Problem 1 and Problem 2. Consider the problem of storing shoe boxes of dimensions  $l \times b \times h$  in a compartment of size  $L \times B \times H$ . Since it is insisted that the storage orientation should satisfy the condition that the box side representing  $h$  should be aligned with the compartment side representing  $H$ , the problem of finding the storage pattern of the boxes in the compartment boils down to the two-dimensional bin packing problem described above with bigger sheet having dimensions  $L \times B$  and the smaller sheet having the dimensions  $l \times b$ , that is  $BPP(L, B, l, b)$ .

#### 4.3 Stage 1

Consider box type  $i$  and compartment type  $j$  with dimensions  $l_i \times b_i \times h_i$  and  $L_j \times B_j \times H_j$  respectively. Solve the bin packing problem  $BPP(L_j, B_j, l_i, b_i)$  for each combination of  $i$  and  $j$ ,  $i = 1, 2, \dots, m$ , and  $j = 1, 2, \dots, n$ , using any *bin packing solver*. Note that this solution may be optimal or near-optimal. It depends upon the solver that is used for solving the problem. If  $p_{ij}$  is the number of smaller sheets that can be accommodated in the bigger sheet according to the solution obtained, then  $w_{ij} = p_{ij} [H/h]$  boxes (of type  $i$ ) can be stored in any compartment of type  $j$ . Here,  $[H/h]$  is the largest integer less than or equal to  $H/h$ .

#### 4.4 Stage 2

Recall Problem 1 of storing  $a_i$  boxes of type  $i$  in  $\sum_j d_j$  compartments with  $d_j$  compartments of type  $j$ . Set  $u_j = V_j$ , the volume of type  $j$  compartment. For each combination of  $i$  and  $j$ ,  $i = 1, 2, \dots, m$ , and  $j = 1, 2, \dots, n$ , let  $w_{ij}$  be the number of boxes of type  $i$  that can be stored in type  $j$  compartment as per the solutions obtained in Stage 1. Let  $x_{ij}$  be the number of compartments of type  $j$  assigned to store boxes of type  $i$ . Note that  $x_{ij}$ s are the decision variables. Solve the  $GTP(W, a, d, u)$ . The solution is a solution to Problem 1. To solve Problem 2, set  $u_j = 1$  for each  $j$ , and solve the  $GTP(W, a, d, u)$ . Its solution is a solution to Problem 2.

To illustrate the above solution procedure, consider the problem presented in (Das, 2005) with 6 compartment types and 28 box types. For the sake of completeness, the dimensions of boxes and compartments as well as the values the parameters  $a$ ,  $d$  and  $u$  are reproduced in

Annexure-I from (Das, 2005). Note that there are three sets of data in this case. The first set corresponds to  $w_{ij}s$  where the boxes in compartments are stored using only vertical pattern of storing. In second and third cases, the data correspond to  $w_{ij}s$  where the boxes in compartments are stored according to horizontal and mixed (heuristic) patterns respectively (see Das, 2005). Therefore, for Problem 1, where the objective is to minimize the volume of the used compartments, there are three choices: use any of the three patterns (vertical, horizontal or mixed) and get a solution that will minimize the volume of the used compartments. Thus, there are three optimization problems for Problem 1. Similarly, there are three optimization problems for Problem 2 where the objective function is to minimize the number of assigned compartments. Thus, there are six GTPs under consideration here.

All the six GTPs mentioned above have been solved using a professional OR package. The solutions are presented in Annexure-II. Since GTPs encountered in this process were solved as integer-linear programming problems by the solver, determination of optimal solutions was time consuming. Therefore, the solution processes had to be terminated before obtaining optimal solutions. Each of the six problems was terminated around one minute and the best solutions obtained within one minute were recorded. Comparing the objective values of the solutions with respective lower bounds (the solver presents the lower bounds while solving), the solutions obtained are pretty close to optimality. It is possible that these solutions are indeed optimal but there is no guarantee.

A summary of the solutions obtained is presented in Table 1. From Table 1, taking the number of compartments as the objective function seems to be better compared to taking the volume of the used compartments as the objective function. This way, more empty compartments will be available for future use. It may be noted that the volumes of the assigned compartments for Problem 2 are almost close to the volumes of assigned compartments for Problems 1.

In (Das, 2005), efficiency of optimization (of the heuristic) was presented in terms of comparing the mixed type with vertical or horizontal type of arrangement. However, a right perspective would be to take a holistic view of storing boxes in the compartments for a given requirement of boxes (that is, for a given  $a$  vector). For example, efficiency of mixed pattern over horizontal pattern can be computed as reduction either with respect to number of compartments or with respect to volume of assigned compartments. For instance, the reduction in number from horizontal to mixed is equal to 5.29% ( $= \frac{699-662}{699} \times 100$ ). The reduction with respect to volume in this case is equal to 4.23% ( $= \frac{11905-11401}{11905} \times 100$ ). It must be noted that in Table 3 of (Das, 2005), utilization percentages are uniformly better in horizontal arrangement compared to vertical arrangement. But from Table 1, vertical arrangement is better compared to horizontal arrangement. The reason is that while calculating the efficiency percentages in (Das, 2005) the storage requirement vector  $a$  is ignored.

## 5. THE WAREHOUSING PROBLEM

Consider the problem of storing the footwear boxes in compartments as described in Problem 1 or Problem 2 of Section 2. Two situations can be considered here. In Situation 1, all the compartments are empty and the boxes are ready for storage (a time when the warehouse is just setup or a time when all the existing boxes from the warehouse are cleared). In this case, the problem can be solved using the procedure described in the previous section.

Consider the second situation, Situation 2, where the problem is dynamic. That is, the warehouse is already in use and there will be periodic inward and outward movement of boxes in the warehouse. Assume that the consignment of boxes to be stored in the warehouse arrives on every Friday, and that the boxes from the

Table 1. Summary of Optimization Results

Bin Packing Type	Number of Compartments as Objective							Total Number	Total Volume
	C1	C2	C3	C4	C5	C6			
Vertical	390	303	0	0	6	0	699	11905	
Horizontal	390	325	4	0	0	0	719	12219	
Mix (Heuristic)	390	270	2	0	0	0	662	11401	
Bin Packing Type	Volume of Compartments as Objective							Total Number	Total Volume
	C1	C2	C3	C4	C5	C6			
Vertical	275	179	107	178	52	44	835	11677	
Horizontal	300	360	13	15	52	49	789	12027	
Mix (Heuristic)	157	392	13	178	52	39	831	11244	

Note: Figures in the columns under compartment types are the number of compartments.



warehouse are withdrawn on a daily basis by the retail stores. In such a situation, the warehouse manager has to decide, on every Friday, how to store the boxes in the compartments. Unlike in Situation 1, compartments for storage in this case may already contain certain boxes. To get an insight of the problem, consider a situation in which two compartments of same type, say type C1 which can hold 496 B01 type boxes according to the best known storage pattern, are in the following states when a new consignment of boxes arrived: the first compartment has 300 B01 boxes and can accommodate 196 more B01 boxes from the new consignment, and the second compartment had 200 B01 boxes and can accommodate 296 more B01 boxes from the new consignment. Thus, the space availability in compartments for storage is dynamic in this situation. Optimization in this dynamic situation not only depends upon the storage decisions but also on the way in which the boxes are withdrawn, the issuing procedure, from the compartments. Intuitively it is clear that optimization will be effective if more space is made available for storing during each consignment. Therefore, this aspect is also taken into consideration while providing the solution to the dynamic situation of the warehousing problem (see Step 1 of the procedure described below).

The solution procedure using *GTP* described earlier cannot handle this situation directly. However, it can be made to work by carrying out a simple procedure. The complete solution to the dynamic situation of the warehousing problem is described below.

### 5.1 Solution Procedure for the Dynamic Warehousing Problem

**Step 1:** While issuing boxes, always draw boxes from the largest size compartments and continue issues from the same compartments until they become empty. This will ensure that at any point of time, the number of partially filled compartments of a given type of boxes is at most one. Furthermore, that partially filled compartment is the largest (in terms of containing the boxes) among all compartments containing the given type of boxes.

**Step 2:** Look at all the compartments which are partially filled when the new consignment arrives. Fill these compartments as much as possible with boxes from the new consignment using the same type of boxes (that is, ensure that none of the partially filled boxes will have mixed box types after filling from the new consignment).

**Step 3:** Look at box types that are still remaining to be stored from the new consignment. It is easy to see that no compartment will be partially filled with the box types that are yet to be stored. For if there is partially filled compartment with a particular type of boxes and such type of boxes are still to be stored, it means that the previous step has not been executed correctly.

**Step 4:** Define the vectors  $a$  and  $d$  as follows:  $d_j$  is the number of empty compartments of type  $j$  after carrying out steps 2 and 3 above;  $a_i$  is the number of boxes of type  $i$  that are yet to be stored after carrying out the steps 2 and 3 above. Solve the *GTP* ( $W, a, d, u$ ).

### 5.2 Economic Procurement of Compartments

An issue that was raised in (Das, 2005) is: how to plan procurement of compartments for the warehouse? Again this was addressed in (Das, 2005) by considering the average efficiencies of compartments over different types of boxes (see Annexure 5 of (Das, 2005) for details). This approach is not quite appropriate as it does not consider the storage requirements (the  $a$  vector) in any way. Perhaps, the correct way is to consider the variation in the storage requirement vector  $a$  and determine the procurement of compartments. This method is elaborated in the next paragraph.

Consider the storage vector  $a$ , that is the inventory position of the warehouse. Clearly,  $a$  is a random vector. According to earlier discussion, let us say that the  $a$  vector gets updated on every Friday. Let  $a^1, a^2, \dots$  be the vectors corresponding to the first, second,  $\dots$  Fridays respectively. Suppose these data are available for  $p$  weeks. Solve the corresponding *GTPs*,  $GTP(W, a^q, d, u)$ , for each  $a^q, q = 1, 2, \dots, p$ . The choice of the vector  $d$  can be made subjectively. For instance,  $d_j$  can be taken as the number of  $j^{\text{th}}$  type compartments already existing plus  $c_j$ , where  $c_j$  is an upper bound on the number of  $j^{\text{th}}$  type compartments that can be procured additionally. Let  $g^q$  be the vector representing number of compartments according to the optimal solution for the *GTP* with  $a^q$  as the storage requirement vector. That is,  $g_j^q$  is the number of  $j^{\text{th}}$  type compartments required according to optimal solution. It is clear that the choice on the number of compartments to be procured should depend upon the  $g^q$  vectors. There could be different strategies for this choice. For instance, one such strategy could be to procure  $c_j^*$  compartments of  $j^{\text{th}}$  type where  $c_j^*$  is maximum of  $(\max_q g_j^q - d_j^*)$  and 0, where  $d_j^*$  is the existing number of type  $j$  compartments.

## 6. DECISION SUPPORT SYSTEM FOR THE WAREHOUSING PROBLEM

The solution methodology provided in this article can be effectively used for managing a warehouse system such as those encountered by footwear manufacturing companies. It calls for a good computerized inventory management system and good inventory management practices. Once a computerized inventory management system is in place to monitor the receipts and issues from the warehouse along with the maintenance of compartment-wise stock positions, a decision support system (a computer package) can be developed using the method described in this article and can be integra-

ted with the warehouse inventory management system.

It should be noted here that the method of solving the storage problem described in this article has two sets of optimization problems: (i) bin packing problems and (ii) the GTPs. An important point that should be observed here is that the bin packing problems need to be solved only once. However, if new types of boxes and/or compartments are introduced into the warehouse, then the bin packing problems need to be solved (once) for the new types introduced.

## 7. SUMMARY

This article considers a class of warehousing problems in which cuboidal objects are stored in cuboidal compartments with some restrictions on storage pattern that are natural and common in practical applications. Das (2005) provided a partial solution to the problem through a heuristic and suggested a nonlinear programming formulation to solve the problem. In this article, it is shown that the problem can be solved as a generalized transportation problem. Finally, the article considers the dynamic version of the warehouse problem which deals with the inventory management system of the warehouse. A solution procedure for this problem is presented. This procedure can be easily translated into a decision support system for the dynamic inventory management problem. The problem of economic procurement of compartments discussed in (Das, 2005) is reviewed and a method to handle this problem is outlined.

## ACKNOWLEDGEMENT

The author wishes to thank Dr. G.S.R. Murthy for suggesting him to read Das, 2005 and work on it. The author records his deep gratitude to Professor Katta G. Murthy for his valuable comments. Further, the author wishes to thank the referee for reviewing the manuscript submitted and also the editor of the journal for their valuable comments made that have helped to enrich the contents of this paper.

## REFERENCES

- Bengtsson., B. E., "Packing rectangular pieces-a heuristic approach," *The Computer Journal* 25 (1982), 353-357.
- Chung, F. K. R., M. R. Garey, and D. S. Johnson, "On packing two- dimensional bins," *SIAM Journal of Algebraic and Discrete Methods* 3 (1982), 66-76.
- Coffman Jr., E. G., G. Galambos, S. Martello, and D. Vigo, "Bin packing approximation algorithms: combinatorial analysis," In *Handbook of Combinatorial Optimization*, edited by D. Z. Du and P. M. Pardalos: Kluwer Academic Publishers, 1998.
- Coffman, Jr., E. G., M. R. Garey, and D. S. Johnson, "Bin packing approximation algorithms: a survey," *In Approximation Algorithms for NP-Hard Problems*, editor by D. Hochhaum. Boston: PWS Publishing Company, 1996.
- Das, P., "A Heuristic Approach for arrangement of Footwear Boxes to maximize Space Utilization and related Business Issues," *International Journal of Management Science* 11, 2 (2005), 61-84.
- Dyckhoff, H., "A Typology of Cutting and Packing Problem," *European Journal of Operational Research* 44 (1990), 145-159.
- Dyckhoff, H. and U. Finke, "Cutting and Packing in Production and Distribution: A Typology and Bibliography," Physica-Verlag, Heidelberg, 1992.
- El-Bouri, A., N. Popplewell, S. Balakrishnan, and A. Alfa, "A search based heuristic for the two-dimensional bin-packing problem," *INFOR* 32 (1994), 265-274.
- Haessler, R. W. and P. E. Sweeney, "Cutting Stock Problems and Solution Procedures," *European Journal of Operational Research* 54 (1991), 141-150.
- Hinxman, A. I., "The Trim-Loss and Assortment Problems: A Survey," *European Journal of Operational Research* 5 (1980), 8-18.
- Murty, K. G., *Network Programming*, Prentice-Hall, Englewood Cliffs, NJ, 1992.
- Sweeney, P. E. and E. R. Paternoster, "Cutting and Packing Problems: A Categorized, Application-Orientated Research Bibliography," *Journal of the Operational Research Society* 43, 7 (1992), 691-706.

### ANNEXURE-I : SAMPLE PROBLEM

Table 2. Warehouse Problem Data

Box Dimensions	Box Type	Compartment Type/Dimension						No. of Boxes ( <i>a</i> )
		C1	C2	C3	C4	C5	C6	
8.5×2×3.5	B1	496	400	310	250	210	150	1228
7×4.5×3	B2	297	234	198	156	144	96	3704
9.5×3.5×3	B3	297	216	198	144	126	84	682
7.5×4.5×3	B4	288	225	192	150	144	96	1832
9×4×3.5	B5	240	192	150	120	91	65	200
10.5×3.5×3.5	B6	208	160	130	100	91	65	354
7.5×5×3.5	B7	216	168	135	105	98	70	5783
11×3×4	B8	238	182	136	104	96	64	1382
11.5×3×4	B9	224	168	128	96	90	60	1326
11.5×3.5×3.5	B10	208	160	130	100	91	65	398
9×5×3.5	B11	192	152	120	95	84	60	421
10.5×4.5×4	B12	154	126	88	72	66	44	1813
10.5×5×4	B13	126	98	72	56	60	40	11842
10×6×4	B14	126	98	72	56	42	28	1765
11×5.5×4	B15	112	84	64	48	36	24	7366
10.5×6×4	B16	126	84	72	48	36	24	4365
11.5×5×4.5	B17	108	84	72	56	54	36	16637
12×5.5×4	B18	112	84	64	48	36	24	3665
12×5×4.5	B19	108	84	72	56	54	36	670
11.5×6×4	B20	112	84	64	48	36	24	5724
11.5×5.5×4.5	B21	96	72	64	48	36	24	749
11×7.5×4	B22	91	63	52	36	36	24	354
12×7×4	B23	84	70	48	40	36	24	1078
11×8×4	B24	84	63	48	36	24	16	192
12.5×7×4.5	B25	54	36	36	24	30	20	402
11.5×7.5×5	B26	60	45	36	27	30	18	1205
12.5×10×4	B27	49	35	28	20	18	12	306
13×8×5	B28	45	35	27	21	20	12	192
Max. no. of Compartments, $d^t$		390	534	130	178	52	52	

Note: Compartment Dimensions: C1(4×2×2.4); C2(3×2×2.4); C3(4×2×1.6); C4(3×2×1.6); C5(3×1.3×2.3); C6(3×1.3×1.6).



### ANNEXURE-II : NEAR OPTIMAL SOLUTIONS

Table 3. Solutions Using 3 Types of Cutting Patterns to Minimize Volume

	Vertical						Horizontal						Mixed (Heuristic)					
B1	1	3	0	0	0	0	1	2	0	0	0	0	1	2	0	0	0	0
B2	1	16	0	0	0	0	3	13	0	0	0	0	0	16	0	0	0	0
B3	3	0	0	0	0	0	0	4	0	0	0	0	1	0	2	0	0	0
B4	2	6	0	0	0	0	4	4	0	0	0	0	1	7	0	0	0	0
B5	0	1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
B6	1	1	0	0	0	0	2	0	0	0	0	0	1	1	0	0	0	0
B7	25	3	0	0	0	0	29	2	0	0	0	0	0	35	0	0	0	0
B8	4	3	0	0	0	0	4	3	0	0	0	0	6	0	0	0	0	0
B9	6	0	0	0	0	0	6	0	0	0	0	0	6	0	0	0	0	0
B10	2	0	0	0	0	0	0	3	0	0	0	0	2	0	0	0	0	0
B11	3	0	0	0	0	0	2	1	0	0	0	0	0	3	0	0	0	0
B12	0	17	0	0	0	0	13	0	0	0	0	0	0	15	0	0	0	0
B13	94	0	0	0	0	0	0	132	0	0	0	0	0	121	0	0	0	0
B14	16	0	0	0	0	0	16	0	0	0	0	0	11	4	0	0	0	0
B15	0	88	0	0	0	0	66	0	0	0	0	0	66	0	0	0	0	0
B16	39	0	0	0	0	0	39	0	0	0	0	0	34	1	0	0	0	0
B17	100	73	0	0	0	0	64	140	0	0	0	0	130	31	0	0	0	0
B18	2	41	0	0	0	0	32	1	0	0	0	0	32	1	0	0	0	0
B19	2	6	0	0	0	0	7	0	0	0	0	0	4	3	0	0	0	0
B20	50	2	0	0	0	0	50	2	0	0	0	0	49	3	0	0	0	0
B21	0	11	0	0	0	0	8	0	0	0	0	0	8	0	0	0	0	0
B22	3	2	0	0	0	0	3	2	0	0	0	0	4	0	0	0	0	0
B23	1	16	0	0	0	0	2	13	0	0	0	0	2	13	0	0	0	0
B24	1	2	0	0	0	0	1	2	0	0	0	0	1	2	0	0	0	0
B25	0	7	0	0	5	0	7	0	1	0	0	0	7	1	0	0	0	0
B26	18	3	0	0	0	0	19	0	2	0	0	0	18	3	0	0	0	0
B27	11	0	0	0	0	0	7	1	0	0	0	0	5	2	0	0	0	0
B28	5	2	0	0	0	0	4	0	1	0	0	0	0	6	0	0	0	0
	390	303	0	0	6	0	390	325	4	0	0	0	390	270	2	0	0	0
A/B/P	11677/11638/0.33						12027/11978/0.41						11244/11181/0.56					

Note: A = achieved, B = Bound, P = 100×(A-B)/B.

### ANNEXURE-II : NEAR OPTIMAL SOLUTIONS

Table 4. Solutions Using 3 Types of Cutting Patterns to Minimize Number of Compartments

	Vertical						Horizontal						Mixed (Heuristic)					
B1	1	3	0	0	0	0	1	2	0	0	0	0	1	2	0	0	0	0
B2	1	16	0	0	0	0	3	13	0	0	0	0	0	16	0	0	0	0
B3	3	0	0	0	0	0	0	4	0	0	0	0	1	0	2	0	0	0
B4	2	6	0	0	0	0	4	4	0	0	0	0	1	7	0	0	0	0
B5	0	1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
B6	1	1	0	0	0	0	2	0	0	0	0	0	1	1	0	0	0	0
B7	25	3	0	0	0	0	29	2	0	0	0	0	0	35	0	0	0	0
B8	4	3	0	0	0	0	4	3	0	0	0	0	6	0	0	0	0	0
B9	6	0	0	0	0	0	6	0	0	0	0	0	6	0	0	0	0	0
B10	2	0	0	0	0	0	0	3	0	0	0	0	2	0	0	0	0	0
B11	3	0	0	0	0	0	2	1	0	0	0	0	0	3	0	0	0	0
B12	0	17	0	0	0	0	13	0	0	0	0	0	0	15	0	0	0	0
B13	94	0	0	0	0	0	0	132	0	0	0	0	0	121	0	0	0	0
B14	16	0	0	0	0	0	16	0	0	0	0	0	11	4	0	0	0	0
B15	0	88	0	0	0	0	66	0	0	0	0	0	66	0	0	0	0	0
B16	39	0	0	0	0	0	39	0	0	0	0	0	34	1	0	0	0	0
B17	100	73	0	0	0	0	64	140	0	0	0	0	130	31	0	0	0	0
B18	2	41	0	0	0	0	32	1	0	0	0	0	32	1	0	0	0	0
B19	2	6	0	0	0	0	7	0	0	0	0	0	4	3	0	0	0	0
B20	50	2	0	0	0	0	50	2	0	0	0	0	49	3	0	0	0	0
B21	0	11	0	0	0	0	8	0	0	0	0	0	8	0	0	0	0	0
B22	3	2	0	0	0	0	3	2	0	0	0	0	4	0	0	0	0	0
B23	1	16	0	0	0	0	2	13	0	0	0	0	2	13	0	0	0	0
B24	1	2	0	0	0	0	1	2	0	0	0	0	1	2	0	0	0	0
B25	0	7	0	0	5	0	7	0	1	0	0	0	7	1	0	0	0	0
B26	18	3	0	0	0	0	19	0	2	0	0	0	18	3	0	0	0	0
B27	11	0	0	0	0	0	7	1	0	0	0	0	5	2	0	0	0	0
B28	5	2	0	0	0	0	4	0	1	0	0	0	0	6	0	0	0	0
	390	303	0	0	6	0	390	325	4	0	0	0	390	270	2	0	0	0
A/B/P	699/696/0.43						719/716/0.42						662/658/0.61					

Note: A = achieved, B = Bound, P = 100×(A-B)/B.