

Solving the Three-Dimensional Palet-Paking Problem Using Mixed 0 - 1 Model

Adel Mohammed Al-Shayea

Industrial Engineering Department, College of Engineering, King Saud University, Riyadh, Saudi Arabia.
E-mail: alshayea@ksu.edu.sa

Received October 6th, 2011; revised November 16th, 2011; accepted November 28th, 2011.

ABSTRACT

The distribution of Pallet Packing Problem is to load a set of distinct boxes with given dimensions on pallets or in containers to maximize volume utilization. This problem is still in its early stages of research, but there is a high level of interest in developing effective models to solve this NP-hard problem to reduce the time, energy and other resources spent in packing pallets. In this paper, the three-dimensional pallet loading with mixed box sizes model has been developed. This loading model allows many boxes of various sizes to be placed onto the same pallet. The model also considers the number or proportion of each box size that can be loaded on a pallet. No restrictions are placed on the dimensions of the boxes, the pallets, or the number of different box sizes that can be considered. Therefore, the objective of this work is to determine how to most efficiently load a given pallet by maximizing the volume occupied by its load of boxes. Tests on several problems were implemented using OR library in order to show the validation of the proposed model. The results showed that the formulated mixed 0 - 1 models provide exact solutions for the pallet-packing problem. The computational time requirements of the developed model prevent its use in real-time palletizing applications. As microcomputer chip technology continues to evolve the lengthy computation time may prove to be less of a problem in real time applications.

Keywords: Pallet, Packing Problem, Optimization, Mixed 0 - 1 Models, 3-D Pallet Loading

1. Introduction

Everyday many items are shipped from one place to another. These items are put in containers or pallets. To ship more items while spending less energy, time, and money, the items should be packed optimally or at least near optimally [1]. This problem becomes even more important in the field of air shipping [2]. The pallet-loading problem is a problem with many different variants. The early form of this type of problem was the one-dimensional loading or packing problem, in which a set of n positive values w_j , e.g. weight values, must be partitioned into the minimum number of subsets so that the total value in each subset does not exceed a given pallet capacity W . The two-dimensional of pallet-loading problem extends the one-dimensional pallet-loading problem. Instead of considering only one set of positive values, two different sets of positive values are considered, namely two different dimensions, e.g. width and length of the rectangular pieces to be cut out. As expected, this problem is harder to solve than the one or two-dimensional pallet-loading problems [3-5].

These pallet-loading problems are NP-hard problems. NP stands for 'non-deterministic polynomial'. NP-hard means the solution time increases exponentially as the size of the problem increases. The three-dimensional pallet-loading problem is strongly NP-hard because the three-dimensional pallet-loading problem is a special case of the one-dimensional problem [6,7].

The three-dimensional packing problem is a natural generalization of the classical one- and two-dimensional problems. In general, optimal solutions are computationally impractical to achieve [8]. For this reason, most of the studies have focused on the practical aspects of loading a container and developing heuristic solutions based on the concept of filling out the container with boxes organized in layers, walls, and columns. In other cases, two-dimensional pallet packing heuristics are applied to the general three-dimensional container-loading problem. These heuristics are, in general, on-line packing algorithms, which mean they pack boxes one-by-one in a given order. More precisely, when the algorithm is packing a box, it has information only about the boxes previously packed, and

once a box is packed, it cannot be moved to another place. This technique is not efficient and is also not applicable, when applying the load balance and other constraints. Since the pallet-packing problem has a large solution space, it is extremely difficult to prove that a solution is the global optimum. Only with many different sets of boxes can an algorithm be tested and its performance evaluated.

2. Review of Relevant Literature

Gehring *et al.*'s paper [9] presents a heuristic for packing non-identical items within a container. They, like George and Robinson [4], utilized the idea of packing sections of the container across the full width and height. They utilized an ordering based on decreasing volume, having placed the first block in a section (layer), and the layer determining box (LDB). They also developed a packing across the container floor first and then upwards. This tends at first to produce something of a decreasing wedge across the width of the container. This approach does ensure that cargo sections can be moved around so as to provide appropriate weight redistribution, but it will clearly lead in some instances to reduced volume utilization. A further aspect associated with not allowing boxes to straddle sections is that of load stability. Packing where boxes do straddle between layers can produce a more cohesive load. Han, Knott and Egbelu [10] showed that the idea of walls needs not to be restricted to the vertical sides of the container. They described an algorithm in which the container (major prism) is packed with identical boxes (minor prisms). The algorithm as described is designed for only a single box type that is constant in both size and shape and no practical constraints are considered. The approach is to produce packing of L-shaped modules, with the initial module considered spanning the whole of the container base, and one of the container walls. The arrangement within the "L" is determined by dynamic programming (similar to the approach of Steudel [11], which maximizes the edge utilization). The idea of building walls along any of the six faces of the container is an interesting one; however, the example they used fits one less box than that obtained by stacking multiples of two different 'wall' arrangements on the floor of the container.

The weakness in the approach of Han *et al.* [10] is a result of maximizing the utilization of the perimeter of the "L" module. No evidence was presented to suggest why an L-shaped module approach should be adopted. Their example consists of packing a container of size 48" by 42" and 40" with boxes 11" by 6" by 6". They were able to fit 195 boxes, a 95.16% volume utilization of the container. They quoted the US General Services Administration whose published results (1966) for the same problem only provide 82.5% utilization).

Mohanty *et al.* [12] proposed a multi-dimensional knapsack problem approach to the three-dimensional packing problem dealing with filling up various containers with boxes. Their objective was to maximize utilization of the space in the containers or the value of the contents of the containers. They used a column generating procedure which heuristically uses a "greedy approach" to generate columns one at a time, without considering any constraints other than overlapping and dimensions of the containers. Since they used a "greedy approach", their approach was not robust and was strongly affected by the number of different items to be packed.

Kocjan and Holmström [13] developed a model producing a high degree of stability. The results obtained during evaluation showed great improvement in the number of stable patterns in comparison with results reported earlier. Moreover, most of the solved cases also ensured optimality in terms of utilization of a pallet. Recently, Junqueira *et al.* [14] developed mixed integer linear programming models for the container loading problem that consider the vertical and horizontal stability of the cargo and the load bearing strength of the cargo. The models can also be used for loading rectangular boxes on pallets where the boxes do not need to be arranged in horizontal layers on the pallet. A comprehensive performance analysis using optimization software with 100s of randomly generated instances showed that those developed models are able to handle only problems of a moderate size.

Terno *et al.* [15] employed a different heuristic algorithm. In addition to the dimension and overlapping constraint, they took total weight limit of the pallet and the stability constraints into account. They employed a layering approach while packing each layer by using a branch and bound solution method. They solved 700 problem sets among the problems that Bischoff *et al.* [16] solved and made comparisons with past work. Their solutions were better than Bischoff *et al.*'s solutions, but since their model was mainly designed for the "Manufacturer's Pallet Packing Problem", as the number of different items increases the volume utilization declines.

Martello *et al.* [6] developed a branch-and-bound method to solve the three-dimensional packing problem. They tried to orthogonally pack all the items into the minimum number of pallets. A computational test was presented showing that problems with the number of boxes less than 30 and 50 were solved. One weakness of their method is that when the average number of items per pallet gets bigger, the problem becomes harder to solve. Another weakness was that they assumed that the items might not be rotated. They considered only basic type of constraints (overlapping and pallet dimension limits).

Ballew [17] developed a mathematical formulation similar to the analytical method of Chen *et al.* [18], by us-

ing nonlinear integer programming on a simplified version of the problem. He developed a general mathematical formulation. Unfortunately, when implemented, the solver package hyper lingo found a local optimum to a very simplified and small problem of just three boxes without considering several important constraints. The formulation of a bigger problem with more boxes was unrealistic because the number of variables and constraints increases incredibly fast as the number of boxes increases.

3. Statement of the Problem

The problem is a three-dimensional pallet-packing problem which is related to the two general types of pallet packing problems. These types are the “manufacturer’s pallet packing problem” and the “distributor’s pallet packing problem.” The ‘manufacturer’s pallet packing problem’ is easier to solve since it seeks the optimum layout of identical rectangular boxes on a rectangularly shaped pallet. On the other hand, the “distributor’s pallet packing problem,” is more difficult to solve since the objective is to load boxes of varying dimensions onto as few pallets as possible [19]. This objective changes to minimize unused pallet space for the case in which only one pallet is loaded. However, most of the time, the items packed are rectangular, and this property makes the problem easier to solve, compared to trying to pack items with different shapes.

The problem at hand is to pack as many boxes as possible from a given set of rectangular-shaped items into a three-dimensional rectangular pallet. The objective is to minimize the unused pallet volume while considering many different kinds of constraints. These constraints are explained in the Scope and Methodology Section. The purpose of this paper is to develop a three-dimensional pallet-packing model that can be solved using LINDO software.

4. Solution Methodology

This paper proposes a zero-one mixed integer linear programming model for the general three-dimensional container-loading problem. The problem involves packing a set of non-uniform cartons into unequal-sized containers. The model considers the issues of carton orientations, multiple carton sizes, multiple container sizes, avoidance of carton overlapping, and space utilization. This model is a modified version of the general pallet-packing model. The modification to the general model is represented by introducing to the model other concerns of the container-loading problem such as weight restriction.

4.1. The Proposed Three-Dimensional Pallet Loading Model

The three-dimensional pallet-loading model is a mixed 0

- 1 integer-programming model which generates an exact optimal solution. The solution of the mixed 0 - 1 model explicitly defines the desired number of boxes of each size and the x, y, z coordinates of each box’s placement location on the pallet. A branch-and-bound technique is employed to solve the mixed 0 - 1 integer-programming model.

4.2. The Mixed 0 - 1 Model

Consider a collection of n boxes, expressed by set $S = \{b_1, b_2, \dots, b_n\}$. Each box b_i has length l_i , width w_i , and height h_i . A loading of S into a pallet of length L , width W , and height limit H is an assignment of boxes to a position within the pallet such that:

- 1) No two boxes in the pallet overlap.
- 2) Each box is contained entirely with the pallet, with its sides parallel to the sides of the pallet.
- 3) The proportion of the number of boxes of a given size to the total number of boxes of a full pallet load must closely approximate the user’s specification.
- 4) The total of boxes’ weights must be less than the weight allowed to be in the pallet.

An optimal loading is achieved if the use of pallet space is maximized under the consideration of the above constraints.

Boxes in set S may or may not have the same dimensions. In addition, the orientations of the boxes in set S are fixed permanently. The length, width, and height of a box must be aligned with the length, width, and height of the pallet, respectively. Length is defined as the dimension along the X-axis. Width is the dimension along the Y-axis, and height is the dimension along Z-axis in Cartesian coordinate space. In this paper, the orientation of box height is assumed to be fixed. Only the length and width of a box are interchangeable. Thus, a box can be placed either in the $(l \times w \times h)$ or $(w \times l \times h)$ directions on the pallet. Individual elements representing these two orientations must be separately included in set S . In addition, the placement location of a box in Cartesian coordinate space is measured relative to the front bottom left corner of the box.

4.3. Initial Notation

The following notation defines all symbols used for the formulations of three-dimensional model. Denote:

- S : A collection of n boxes to be considered, in particular, it is $\{b_1, b_2, \dots, b_n\}$.
- (l_i, w_i, h_i) : The dimensions of $box_i(b_i)$ in set S , and they are length, width, and height, respectively.
- (L, W, H) : The dimensions of a pallet cube, and they are length, width, and height, respectively.
- $(X^\circ, Y^\circ, Z^\circ)$: Pallet location in Cartesian coordinate

space along the x -, the y -, and the z -axis, respectively.

(x_i, y_i, z_i) : Decision variables; the x, y, z coordinates of placement location of the front bottom left corner of box i .

P_k : A binary decision variable associated with the k -th box in set S .

Where

$$P_k = \{1 \text{ if Box } k \text{ is loaded onto the pallet } \}$$

$$P_k = \{0 \text{ if Box } k \text{ is discarded from set } S \}$$

V : The volume of the pallet = $L \times W \times H$

V_k : The volume of box $k = l_i \times w_i \times h_i$

R_g : The desired box proportion of type g .

G_k : The weight of box k .

G : Total boxes' weight allowed.

C_g : A subset of S ; consists of all boxes of size g regardless of box orientation.

$$C_g = b_k (l_k \times w_k \times h_k) = (l_g \times w_g \times h_g) \text{ or } (w_g \times l_g \times h_g), 1 \leq k \leq n$$

M : An extremely large number.

r : Total number of box types, $r \leq n$.

Each box with different orientations, either in $l \times w \times h$ or $w \times l \times h$ must be individually considered in set S .

4.4. Preventing Box Overlaps

This section describes the process of converting the requirements of box overlap avoidance into mathematical constraints. Consider two partially overlapped boxes, A and B shown in **Figure 1 (a)**. The projections of these boxes on the x - y , the x - z and the y - z planes are illustrated in **Figures 1 (b), (c) and (d)**, respectively.

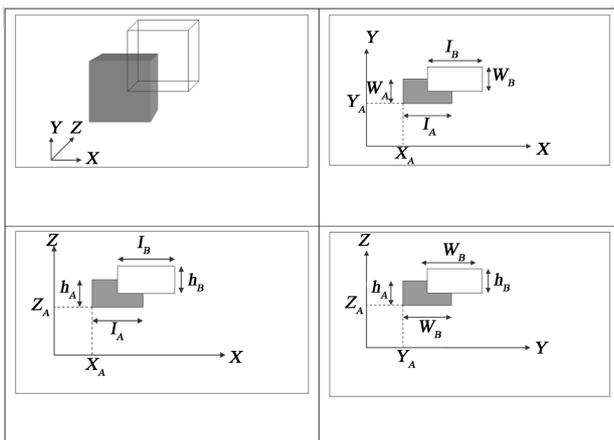


Figure 1. Boxes Overlaps. (a): Two overlapped boxes; (b): Illustrates overlap condition in X and Y ; (c): Illustrates overlap condition in X and Z ; (d): Illustrates overlap condition in Y and Z .

Suppose the location of box A is fixed, and that box B is free to move arbitrarily in Cartesian coordinate space. To avoid overlap of these two boxes, the following conditions must be satisfied:

$$x_B - x_A \geq l_A \tag{1}$$

or

$$x_A - x_B \geq l_B \tag{2}$$

$$y_B - y_A \geq w_A \tag{3}$$

or

$$y_A - y_B \geq w_B \tag{4}$$

$$z_B - z_A \geq h_A \tag{5}$$

or

$$z_A - z_B \geq h_B \tag{6}$$

where

l_A, l_B : Lengths of boxes A and B , respectively.

w_A, w_B : Widths of boxes A and B .

h_A, h_B : Heights of boxes A and B .

(x_A, y_A, z_A) : Front bottom left corner coordinate of box A .

(x_B, y_B, z_B) : Front bottom left corner coordinate of box B .

At least one of these six constraints must hold to prevent overlap of the two boxes.

4.5. Determination of Proportion of Assigned Number of Boxes in a Pallet

The number of boxes of each type to be considered in set S can be determined using the following two equations.

$$n_g / n_i = R_g \tag{7}$$

$$\sum_{i=1}^r n_i \cdot \sum_{i=1}^r v_i = V \tag{8}$$

where

n_g : The number of boxes of type g to be considered in set S .

Equation (7) states that the ratio of the number of type g boxes to the total number of boxes on a pallet should equal to R_g , the desired box proportion of type g . Equation (8) indicates that the total cumulative box volumes should equal to the pallet's volume. By solving Equations (7) and (8), the number of boxes for type g can be obtained as:

$$n_g = \frac{R_g V}{\sum_{i=1}^r R_i v_i} \tag{9}$$

4.6. Formulation of Three-Dimensional Model

The three-dimensional pallet loading problem can now be formulated as a mixed 0 - 1 integer programming model.

$$\text{Maximize } Z = \sum_{k=1}^n v_k P_k \tag{10}$$

Subject to:

1) Avoid overlap of boxes

$$x_j - x_i \leq -l_j \quad \forall ij \tag{11}$$

or

$$x_i - x_j \leq -l_i \quad \forall ij \tag{12}$$

or

$$y_j - y_i \leq -w_j \quad \forall ij \tag{13}$$

or

$$y_i - y_j \leq -w_i \quad \forall ij \tag{14}$$

or

$$z_j - z_i \leq -h_j \quad \forall ij \tag{15}$$

or

$$z_i - z_j \leq -h_i \quad \forall ij \tag{16}$$

2) Confine placement boundary

$$x_k \geq X^\circ P_k \quad \forall k \tag{17}$$

$$y_k \geq Y^\circ P_k \quad \forall k \tag{18}$$

$$z_k \geq Z^\circ P_k \quad \forall k \tag{19}$$

$$x_k \leq (X^\circ + L) - l_k \quad \forall k \tag{20}$$

$$y_k \leq (Y^\circ + W) - w_k \quad \forall k \tag{21}$$

$$z_k \leq (Z^\circ + H) - h_k \quad \forall k \tag{22}$$

3) Weight limitation

$$\sum_{k=1}^n G_k \cdot P_k \leq G \left(\sum_{m=1}^m P_m \right) \tag{23}$$

4) Proportion of boxes assigned

$$\sum_{m \in C_g} P_m \leq R_g \tag{24}$$

$$P_k \in \{0,1\}$$

$$X_k, Y_k, Z_k \geq 0$$

$$i = 1, 2, \dots, n-1$$

$$j = i+1, i+2, \dots, n$$

$$k = 1, 2, \dots, n$$

$$g = 1, 2, \dots, r$$

The front bottom left corner of the pallet is located at the coordinate $(X^\circ, Y^\circ, Z^\circ)$ in Cartesian coordinate spa-

ce. The location of the pallet $(X^\circ, Y^\circ, Z^\circ)$ is selected such that all boxes in set S can be completely placed into the large cube.

The values of $X^\circ, Y^\circ, Z^\circ$ may be determined using the following expressions:

$$X^\circ = n \dots \dots \max(l_i),$$

$$Y^\circ = n \dots \dots \max(w_i),$$

$$Z^\circ = n \dots \dots \max(h_i).$$

The objective function, Equation (10), maximizes the total pallet volume occupied by boxes to be loaded.

In the final solution, any box having $P_k = 1$ is used to construct a pallet pattern. Any box having $P_k = 0$ is discarded from consideration. This formulated mixed 0 - 1 integer-programming model for the three-dimensional pallet-loading problem thus gives the required number of boxes of each type, *i.e.*, every box whose associated P_k equals to one. It also generates the exact placement of a box on the pallet, namely the coordinate (x_k, y_k, z_k) .

4.7. Converting Multiple-Choice Constraints

Equations (11) - (16) in the model make the problem one of multiple-choice programming. To apply existing algorithms for mixed 0 - 1 integer programming, the multiple choice (either/or) constraints must be converted to standard "AND" constraints. The conversion can be accomplished by introducing additional binary variables (Table 1) for each set of multiple-choice constraints.

The six possible combinations of different binary values are:

u_1	u_2	u_3
1	0	0
0	1	0
0	0	1
1	1	0
0	1	1
1	0	1

Table 1. Binary variable and associated RHS values.

Binary variables			RHS values of equations						Applicable Constraint Equation
U1	U2	U3	(25)	(26)	(27)	(28)	(29)	(30)	
1	0	0	-l _j	M	M	M	2M	M	(25)
0	1	0	M	-l _i	M	M	M	2M	(26)
0	0	1	M	M	-w _j	2M	M	M	(27)
1	1	0	M	M	2M	-w _i	M	M	(28)
0	1	1	2M	M	M	M	-h _j	M	(29)
1	0	1	M	2M	M	M	M	-h _i	(30)

The multiple choice constraints of Equations (11) - (16) in the model are equivalent to:

$$x_j - x_i \leq -l_j + M(u_2 + u_3) \tag{25}$$

$$x_i - x_j \leq -l_j + M(u_1 + u_3) \tag{26}$$

$$y_j - y_i \leq -w_j + M(u_1 + u_2) \tag{27}$$

$$y_i - y_j \leq -w_i + M[2 - (u_1 + u_2)] \tag{28}$$

$$z_j - z_i \leq -h_j + M[2 - (u_2 + u_3)] \tag{29}$$

$$z_i - z_j \leq -h_i + M[2 - (u_1 + u_3)] \tag{30}$$

where: $1 \leq u_1 + u_2 + u_3 \leq 2$

$$u_1, u_2, u_3, \{0, 1\}$$

4.8. Box Proportions

Users of the proposed developed model can determine the value of R_g (box proportions) by using the following equation:

$$R_g = \frac{\text{Number of boxes of type } g}{\sum_{i=1}^n \text{Number of boxes of type } i} \tag{30}$$

Where (r) is the number of distinct box types.

Since different box orientations must be considered separately in set $B = \{b_1, b_2, \dots, b_n\}$, set B contains entries which exceed the value of r . If R_g is not placed in the model's constraints, the optimal solution may generate a pallet pattern that contains only identical boxes.

5. Testing and Validating the Proposed Model

In order to test the efficiency of the proposed model, an illustrative example is presented first to explain how the model formatted. Then, the model is tested utilizing five problems randomly selected from the OR library that is shown in the reference.

5.1. Numerical Example

Consider a distribution warehouse using 36" by 24" pallets for shipment. The stacking height limit of a pallet load is 16". The maximum load capacity allowed is 60 lb. A customer order requests 100 units of product A and 200 units of product B. Product A is packaged using 12"×24" carton with the height of 16" and weigh 15 lb. Product B uses 24"×24" carton with the height of 8" and weigh 20 lb.

Prior to formulating the 3-dimensional model, a collection of cartons, sets, and the location of the pallet in Cartesian coordinate space $(X^\circ, Y^\circ, Z^\circ)$ must be determined. Carton proportion of product A is equal to;

Table 2. Example parameter.

Box	Product	Length	Width	Height	Volume	Coordinate	P_i
b_i		b_i	b_i	b_i	b_i		
b_1	A	12	24	16	4608	(x_1, y_1, z_1)	P_1
b_2	A	24	12	16	4608	(x_2, y_2, z_2)	P_2
b_3	B	24	24	8	4608	(x_3, y_3, z_3)	P_3
b_4	B	24	24	8	4608	(x_4, y_4, z_4)	P_4

$R_1 = 100/(100 + 200) = 1/3$ and carton proportion of B is equal to; $R_2 = 200/(100 + 200) = 2/3$. The number of boxes of individual types to be considered in set S can also determined $n_A = 1, n_B = 2$.

Since a carton 12"×24" is not square, a carton with dimensions 24"×12" must be included in set S, therefore:

$$S = \{b_1, b_2, b_3, b_4\}$$

Where:

$$b_1 = 12" \times 24" \times 16"$$

$$b_2 = 24" \times 12" \times 16"$$

$$b_3 = 24" \times 24" \times 8"$$

$$b_4 = 24" \times 24" \times 8"$$

Let $(X^\circ, Y^\circ, Z^\circ) = (100, 100, 100)$ and $M = 500$

The model will be as following:

Maximize $Z = 4608P_1 + 4608P_2 + 4608P_3 + 4608P_4$

Subject to:

$$x_2 - x_1 \leq -24 + 500(u_{12} + u_{13})$$

$$x_1 - x_2 \leq -12 + 500(u_{11} + u_{13})$$

$$y_2 - y_1 \leq -12 + 500(u_{11} + u_{12})$$

$$y_1 - y_2 \leq -24 + 500[2 - (u_{11} + u_{12})]$$

$$z_2 - z_1 \leq -16 + 500[2 - (u_{12} + u_{13})]$$

$$z_1 - z_2 \leq -16 + 500[2 - (u_{11} + u_{13})]$$

$$1 \leq u_{11} + u_{12} + u_{13} \leq 2$$

$$x_3 - x_1 \leq -24 + 500(u_{22} + u_{23})$$

$$x_1 - x_3 \leq -12 + 500(u_{21} + u_{23})$$

$$y_3 - y_1 \leq -24 + 500(u_{21} + u_{22})$$

$$y_1 - y_3 \leq -24 + 500[2 - (u_{21} + u_{22})]$$

$$z_3 - z_1 \leq -8 + 500[2 - (u_{22} + u_{23})]$$

$$z_1 - z_3 \leq -16 + 500[2 - (u_{21} + u_{23})]$$

$$1 \leq u_{21} + u_{22} + u_{23} \leq 2$$

$$x_4 - x_1 \leq -24 + 500(u_{32} + u_{33})$$

$$x_1 - x_4 \leq -12 + 500(u_{31} + u_{33})$$

$$y_4 - y_1 \leq -24 + 500(u_{31} + u_{32})$$

$$y_1 - y_4 \leq -24 + 500[2 - (u_{31} + u_{32})]$$

$$z_4 - z_1 \leq -8 + 500[2 - (u_{32} + u_{33})]$$

$$z_1 - z_4 \leq -16 + 500[2 - (u_{31} + u_{33})]$$

$$1 \leq u_{31} + u_{32} + u_{33} \leq 2$$

$$x_3 - x_2 \leq -24 + 500(u_{42} + u_{43})$$

$$x_2 - x_3 \leq -24 + 500(u_{41} + u_{43})$$

$$y_3 - y_2 \leq -24 + 500(u_{41} + u_{42})$$

$$y_2 - y_3 \leq -12 + 500[2 - (u_{41} + u_{42})]$$

$$z_3 - z_2 \leq -8 + 500[2 - (u_{42} + u_{43})]$$

$$z_2 - z_3 \leq -16 + 500[2 - (u_{41} + u_{43})]$$

$$1 \leq u_{41} + u_{42} + u_{43} \leq 2$$

$$x_4 - x_2 \leq -24 + 500(u_{52} + u_{53})$$

$$x_2 - x_4 \leq -24 + 500(u_{51} + u_{53})$$

$$y_4 - y_2 \leq -24 + 500(u_{51} + u_{52})$$

$$y_2 - y_4 \leq -12 + 500[2 - (u_{51} + u_{52})]$$

$$z_4 - z_2 \leq -8 + 500[2 - (u_{52} + u_{53})]$$

$$z_2 - z_4 \leq -16 + 500[2 - (u_{51} + u_{53})]$$

$$1 \leq u_{51} + u_{52} + u_{53} \leq 2$$

$$x_4 - x_3 \leq -24 + 500(u_{62} + u_{63})$$

$$x_3 - x_4 \leq -24 + 500(u_{61} + u_{63})$$

$$y_4 - y_3 \leq -24 + 500(u_{61} + u_{62})$$

$$y_3 - y_4 \leq -24 + 500[2 - (u_{61} + u_{62})]$$

$$z_4 - z_3 \leq -8 + 500[2 - (u_{62} + u_{63})]$$

$$z_3 - z_4 \leq -8 + 500[2 - (u_{61} + u_{63})]$$

$$1 \leq u_{61} + u_{62} + u_{63} \leq 2$$

$$x_1 \geq 100P_1$$

$$y_1 \geq 100P_1$$

$$z_1 \geq 100P$$

$$x_1 \leq (100 + 36) - 12$$

$$y_1 \leq (100 + 24) - 24$$

$$z_1 \leq (100 + 16) - 16$$

$$x_2 \geq 100P_2$$

$$y_2 \geq 100P_2$$

$$z_2 \geq 100P_2$$

$$x_2 \leq (100 + 36) - 24$$

$$y_2 \leq (100 + 24) - 12$$

$$z_2 \leq (100 + 16) - 16$$

$$x_3 \geq 100P_3$$

$$y_3 \geq 100P_3$$

$$z_3 \geq 100P_3$$

$$x_3 \leq (100 + 36) - 24$$

$$y_3 \leq (100 + 24) - 24$$

$$z_3 \leq (100 + 16) - 8$$

$$x_4 \geq 100P_4$$

$$y_4 \geq 100P_4$$

$$z_4 \geq 100P_4$$

$$x_4 \leq (100 + 36) - 24$$

$$y_4 \leq (100 + 24) - 24$$

$$z_4 \leq (100 + 16) - 8$$

$$(P_1 + P_2) \leq (1/3)(P_1 + P_2 + P_3 + P_4)$$

$$(P_3 + P_4) \leq (2/3)(P_1 + P_2 + P_3 + P_4)$$

$$15P_1 + 15P_2 + 20P_3 + 20P_4 \leq 60$$

$$u_{i1}, u_{i2}, u_{i3} \in \{0, 1\}$$

$$u_{i1}, u_{i2}, u_{i3} \in \{0, 1\}$$

$$x_i, y_i, z_i \geq 0$$

$$i = 1, 2, 3, 4$$

One of possible optimal solutions for the formulated problem is:

$$(P_1, P_2, P_3, P_4) = (1, 0, 1, 1)$$

$$(x_1, y_1, z_1) = (124, 100, 100)$$

$$(x_2, y_2, z_2) = (100, 100, 84)$$

$$(x_3, y_3, z_3) = (100, 100, 108)$$

$$(x_4, y_4, z_4) = (100, 100, 100)$$

$$(u_{11}, u_{12}, u_{13}) = (0, 1, 1)$$

$$(u_{21}, u_{22}, u_{23}) = (1, 0, 0)$$

$$(u_{31}, u_{32}, u_{33}) = (1, 0, 0)$$

$$(u_{41}, u_{42}, u_{43}) = (1, 0, 1)$$

$$(u_{51}, u_{52}, u_{53}) = (1, 0, 1)$$

$$(u_{61}, u_{62}, u_{63}) = (0, 1, 1)$$

Note that $P_2 = 0$ and $(x_2, y_2, z_2) = (100, 100, 84)$. This indicates that box 2 (b_2) is placed outside the valid pallet space. Therefore, only boxes 1, 3 and 4 (b_1, b_3 and b_4) are considered in the resulting pallet loading pattern.

Recall that the pallet's corner is placed at coordinate $(X^\circ, Y^\circ, Z^\circ) = (100, 100, 100)$. By subtracting $(X^\circ, Y^\circ,$

Z°) the resulting placement location of a box (x_i, y_i, z_i) can be converted back to the original coordinates. This yield the following results that are illustrated in **Figure 2**:

$$(x_1, y_1, z_1) = (24, 0, 0)$$

$$(x_3, y_3, z_3) = (0, 0, 8)$$

$$(x_4, y_4, z_4) = (0, 0, 0)$$

and

$$(x_2, y_2, z_2) = (0, 0, -16)$$

5.2. The Efficiency of the Proposed Model

Another four problems were randomly selected from OR library that is shown in the reference and run in LINDO software. It was noticed with increasing number of boxes included in the pallet, the execution time is significantly increased so that in the last problem the execution time exceeds 3 hours and the computer stopped running showing out a sign “out of memory.” All problems were running in Pentium(R) 4 CPU 1.7 GHz with 256 MB of RAM. The following table shows the change in the number of orientations associated with execution time. (**Table 3**).

Therefore, another six problems were chosen in which 2 problems with two different box sizes, another two problems with three different box sizes and finally two problems with four different box sizes. All have different orientations forming 12 positions shapes. Unfortunately, none of them succeed to present a solution except when one of the constraint is removed such eliminating proportion constraints. It may get a reasonable solution in a reasonable time.

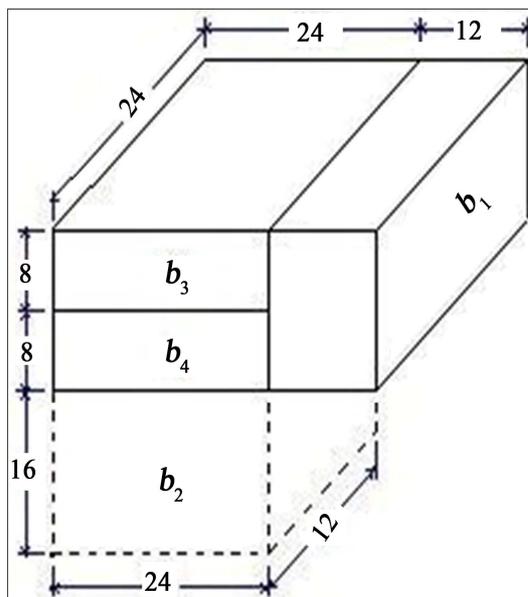


Figure 2. The optimal pallet pattern of the numerical example.

Table 3. Execution time for selected problems.

Problem number	Number of orientation	Execution time	Result
1	4	5 seconds	Optimum solution is obtained
2	6	15 seconds	Optimum solution is obtained
3	10	24 minutes	Optimum solution is obtained
4	12	3 hours and 40 minutes	Stopped because of “out of memory.”

6. Results and Discussions

The developed mixed 0 - 1 integer model for the three dimensional problem has been solved by using a branch-and bound procedure, which employed linear programming techniques to solve for continuous solutions. The branch-and-bound procedure did not destroy the primal feasibility of the LP solution. As a result, less computer memory and computation time were required.

The formulated mixed 0 - 1 models provided exact solutions for the pallet-packing problem. However, use of 0 - 1 integer variables and multiple choice constraints may require extremely long computation times to reach final optimal solutions. The issue of the developed model’s computation time requirements is addressed in more detail in the following:

Computation Time Requirements

The developed model has robust computation time requirements. The computation time increases significantly as the number of boxes increases. The size of the mixed 0 - 1 integer-programming problem is related to both the number of different box sizes and the number of different entries in set S , previously described. Recall that.

- n = the number of different boxes in set S , and
- r = the number of different box sizes.

The developed model considers the location relationship between every pair of entries in set S for each box size. Each pair of placement alternatives for each box may be denoted as:

$$b_i, \text{ and } b_j, \quad i < j; \quad i, j = 1, 2, \dots, n.$$

It follows that there are $n(n-1)/2$ combinations to be formulated. The problem size may now be formulated in terms of the number of variables and constraint equations. The problem size is a function of both the number of different box sizes and the number of elements in set S . If only one box orientation is permitted, the number of elements in set S may be reduced by half. Alternatively, if only boxes with identical lengths or widths or heights are considered, the problem may be reduced to two dimensions. Computation time increase exponentially as the number of different box sizes increases.

An illustrated example was conducted to evaluate the effectiveness of the developed model. Two different box sizes corresponding to four elements in set S were used. Computation times range exceeds 45 seconds of processor time on a microcomputer. Also, it was obvious from **Table 3** that it was hard to get appropriate answer for the problems with four different box sizes forming 12 different positions shapes because of memory limitations. Therefore, these computational requirements limit the model's use in real-time palletizing applications. The model was responsive to significant and immediate changes in the distributions of box sizes to be loaded. The continuing evolvement of faster microcomputer hardware may remove the model's limitation for use in real-time palletizing applications in the foreseeable future.

7. Conclusions

This paper has presented a transformation procedure for converting the three dimensional pallet loading problem to an exact mixed 0 - 1 integer programming model in which all position relationships among boxes must be considered. The position constraints can be determined by establishing the constraints specified by Equations (1) - (6). Since only two boxes are considered in each constraint set of Equations (1) - (6), there are $n(n-1)/2$ combinations to be formulated.

The developed model does not address the issue of load stability. Use of some sets of carton dimensions may result in some partial voids in the pallet pattern. These voids can be minimized or eliminated thorough the use of filler cartons or standardized carton dimensions. The computational time requirements of the developed model prevent its use in real-time palletizing applications. As microcomputer chip technology continues to evolve the lengthy computation time may prove to be less of a problem in real time applications.

REFERENCES

- [1] E. G. Jr. Coffman and P. W. Shor, "Average-Case Analysis of Cutting and Packing in Two-Dimensions," *European Journal of Operational Research*, Vol. 44, No. 2, 1990, pp. 134-145. [doi:10.1016/0377-2217\(90\)90349-G](https://doi.org/10.1016/0377-2217(90)90349-G)
- [2] J. Verstichel, W. Vancroonenburg, W. Souffriau and G. V. Berghe, "A Mixed Integer Programming Approach to the Aircraft Weight and Balance Problem," *Procedia Social and Behavioral Sciences*, No. 20, 2011, pp. 1051-1059.
- [3] W. Q. Huang and K. He, "An Efficient Algorithm for Solving the Container Loading Problem," 2007. <http://mscmga.ms.ic.ac.uk/jeb/orlib/thpackinfo.html>
- [4] J. A. George and D. F. Robinson, "A Heuristic for Packing Boxes into a Container," *Computers and Operational Research*, Vol. 7, 1980, pp. 147-156. [doi:10.1016/0305-0548\(80\)90001-5](https://doi.org/10.1016/0305-0548(80)90001-5)
- [5] C. H. Che, W. Huang, A. Lim and W. Zhu, "The Multiple Container Loading Cost Minimization Problem," *European Journal of Operational Research*, Vol. 214, 2011, pp. 501-511. [doi:10.1016/j.ejor.2011.04.017](https://doi.org/10.1016/j.ejor.2011.04.017)
- [6] S. Martello, D. Pisinger and D. Vigo, "The Three-Dimensional Bin Packing Problem," *Operations Research, Informatics*, Vol. 48, No. 2, 2000, pp. 256-267. [doi:10.1287/opre.48.2.256.12386](https://doi.org/10.1287/opre.48.2.256.12386)
- [7] T. Dereli and G. S. Das, "A Hybrid 'Bee(s) Algorithm' for Solving Container Loading Problems," *Applied Soft Computing*, No. 11, 2011, pp. 2854-2862. [doi:10.1016/j.asoc.2010.11.017](https://doi.org/10.1016/j.asoc.2010.11.017)
- [8] J. Liu, Y. Yue, Z. Dong, C. Maple and M. Keech, "A Novel Hybrid Tabu Search Approach to Container Loading," *Computers & Operations Research*, No. 38, 2011, pp. 797-807.
- [9] H. Gehring, K. Menschner and M. Meyer, "A Computer-Based Heuristic for Packing Pooled Shipment Containers," *European Journal of Operational Research*, Vol. 44, No. 2, 1990, pp. 277-289. [doi:10.1016/0377-2217\(90\)90363-G](https://doi.org/10.1016/0377-2217(90)90363-G)
- [10] C. P. Han, K. Knott and P. J. Egbel, "A Heuristic Approach to the Three-Dimensional Cargo-Loading Problem," *International Journal of Production Research*, Vol. 27, No. 5, 1989, pp. 757-774. [doi:10.1080/00207548908942585](https://doi.org/10.1080/00207548908942585)
- [11] H. J. Steudel, "Generating Pallet Loading Patterns: A Special Case of the Two-Dimensional Cutting Stock Problem," *Management Science*, Vol. 25, No. 10, 1979, pp. 997-1004. [doi:10.1287/mnsc.25.10.997](https://doi.org/10.1287/mnsc.25.10.997)
- [12] B. B. Mohanty, K. Mathur and N. J. Ivancic, "Value Considerations in Three-Dimensional Packing—A Heuristic Procedure Using the Fractional Knapsack Problem," *European Journal of Operational Research*, Vol. 74, No. 1, 1994, pp. 143-151. [doi:10.1016/0377-2217\(94\)90212-7](https://doi.org/10.1016/0377-2217(94)90212-7)
- [13] W. Kocjan and K. Holmstrm, "Computing Stable Loads for Pallets," *European Journal of Operational Research*, Vol. 207, No. 2, 2010, pp. 980-985. [doi:10.1016/j.ejor.2010.05.005](https://doi.org/10.1016/j.ejor.2010.05.005)
- [14] L. Junqueira, R. Morabito and D. S. Yamashita, "Three-Dimensional Container Loading Models with Cargo Stability and Load Bearing Constraints," *Computers & Operations Research*, Vol. 39, No. 3, 2012, pp. 74-85. [doi:10.1016/j.ejor.2010.05.005](https://doi.org/10.1016/j.ejor.2010.05.005)
- [15] J. Terno, G. Scheithauer, U. Sommerweiß and J. Riehme, "An Efficient Approach for the Multi-Pallet Loading Problem," *Institute for Numerical Mathematics*, Technical University Dresden Mommsenstr, Dresden, Vol. 13, 1997.
- [16] E. E. Bischoff and M. S. W. Ratcliff, "Issues in the Development of Approaches to Container Loading," *Omega Institute*, Vol. 23, No. 4, 1995, pp. 377-390. [doi:10.1016/0305-0483\(95\)00015-G](https://doi.org/10.1016/0305-0483(95)00015-G)
- [17] P. B. Ballew, "The Distributor's Three-Dimensional Palet-Packing Problem: A Mathematical Formulation and Heuristic Solution Approach," *MS Thesis*, Graduate School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, 2000.
- [18] C. S. Chen, S. M. Lee and Q. S. Shen, "An Analytical

Model for Container Loading Problem,” *European Journal of Operational Research*, Vol. 80, No. 1, 1995, pp. 68-76. [doi:10.1016/0377-2217\(94\)00002-T](https://doi.org/10.1016/0377-2217(94)00002-T)

[19] R. G. Askin, and R. S. Charles, “Modeling and Analysis of Manufacturing Systems,” John Wiley and Sons Inc., New York, 1993, pp. 320-321.