Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

# Layer-layout-based heuristics for loading homogeneous items into a single container*

WANG Zhou-jing[1], LI Kevin W.[†‡2]

([1]Department of Automation, Xiamen University, Xiamen 361005, China)

([2]Odette School of Business, University of Windsor, Windsor, Ontario N9B 3P4, Canada)

[†]E-mail: kwli@uwindsor.ca

Received July 24, 2007; revision accepted Sept. 17, 2007

**Abstract:** The container loading problem (CLP) is a well-known NP-hard problem. Due to the computation complexity, heuristics is an often-sought approach. This article proposes two heuristics to pack homogeneous rectangular boxes into a single container. Both algorithms adopt the concept of building layers on one face of the container, but the first heuristic determines the layer face once for all, while the second treats the remaining container space as a reduced-sized container after one layer is loaded and, hence, selects the layer face dynamically. To handle the layout design problem at a layer's level, a block-based 2D packing procedure is also developed. Numerical studies demonstrate the efficiency of the heuristics.

**Key words:** Container loading problem (CLP), Heuristic, Layer, Packing, Optimization

**doi:**10.1631/jzus.2007.A1944 **Document code:** A **CLC number:** N949; O22; TP3

## INTRODUCTION

With the continuing globalization of the economy, the volume of container shipment has significantly increased. A key concern in the container transportation industry is to design efficient and effective loading schemes for maximizing the container space utilization, thereby reducing the container shipment volume and saving logistics costs (Bortfeldt *et al.*, 2003; Mack *et al.*, 2004; Moura and Oliveira, 2005).

In many cases, the items to be loaded into a container are packed in various standard-sized rectangular carton boxes (Ngoi *et al.*, 1994). Depending on whether all items must be completely loaded, a container loading problem (CLP) may involve either a single container (Dyckhoff, 1990) or multiple containers (Takahara, 2005). An alternative way to distinguish different CLPs is to examine the involved box sizes. For the boxes having identical size along the three dimensions, it is called a homogeneous CLP, otherwise, it is a heterogeneous CLP, which may be further refined as a weakly or strongly heterogeneous CLP (Bischoff and Ratcliff, 1995; Wang *et al.*, 2006; 2007).

As CLPs are widely known as NP-hard and do not have exact solutions in polynomial time (Scheithauer, 1992), heuristics are often employed to obtain near-optimal solutions in reasonable time (Ngoi *et al.*, 1994; Mack *et al.*, 2004). Recently, different intelligent techniques, such as genetic algorithms (Bortfeldt and Gehring, 2001), simulated annealing (Jin *et al.*, 2004), and tabu search algorithms (Bortfeldt *et al.*, 2003), have been entertained to solve CLPs. In addition, some practical constraints are now being taken into account. For instance, the weight distribution of boxes loaded into a container is considered by Davies and Bischoff (1999); the impact of limited load bearing strength is the focus of Bischoff (2006); and the loading stability has been examined

---

‡ Corresponding author

* Project supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the Discovery Grant Program

by many authors such as Bortfeldt and Gehring (2001), Bortfeldt *et al*.(2003), and Terno *et al*.(2000).

This article proposes two layer-layout-based heuristics for loading homogeneous rectangular boxes into a single container. These layer-based techniques can be treated as variants of the wall-building approach introduced by George and Robinson (1980) and adapted by many other authors (Loh and Nee, 1992; Bischoff and Ratcliff, 1995; Pisinger, 2002). Instead of fixing the container face to build layers horizontally from back to front (George and Robinson, 1980) or vertically from bottom to top (Loh and Nee, 1992), our heuristics will determine the optimal container face for building layers.

Let $LL$ (along the coordinate $X$, as shown in Fig.2), $WW$ (along the coordinate $Y$), and $HH$ (along the coordinate $Z$) be the three edges of the container and $l$, $w$, and $h$ be the three sides of the box. Layers may be built on the plane $XY$ from the bottom to the top, or on the plane $XZ$ from the right to the left, or on the plane $YZ$ from the back to the front (assuming that the container door is open at the front). For a given plane on which layers are to be stacked, each of the three box dimensions will be sequentially fixed as the individual layer height.

For our first heuristic, the layer face is determined once for all in a static manner (Wang and Li, 2007), and all individual layers with the same box dimension as the height are then aggregated into a thick layer, and a procedure is furnished to determine the height of each aggregated layer such that the usage of the container dimension orthogonal to the layers is maximized.

Our second heuristic selects the layer face iteratively. The process starts with the full container space $LL\times WW\times HH$ and a minimal base-area loss approach is adopted to determine an optimal combination of the layer face and the box orientation to build a layer. After this layer is loaded, the residual container space, denoted by $L\times W\times H$ along the axes of $X$, $Y$, and $Z$ (as shown in Fig.2), is treated as a reduced-sized container and the iteration continues until the residual space is too small to hold additional boxes.

For each individual or aggregated layer, the arrangement of rectangular boxes can be treated as a pallet loading problem due to the uniform height of the layer (Birgin *et al.*, 2005). In this case, the concern is to find a maximal layout plan to arrange identical small rectangles onto a larger rectangle (Scheithauer and Terno, 1996). Numerous studies have been conducted on this important distribution problem (Scheithauer and Terno, 1996; Birgin *et al.*, 2005; Pureza and Morabito, 2005), and the $G$-4 structure of Scheithauer and Terno (1996) has been recognized as an effective way to develop block-based heuristics for the pallet loading problem (Li and Ye, 2002; Alvarez-Valdes *et al.*, 2005). Based on the $G$-4 heuristic in (Li and Ye, 2002), a modified block-based algorithm is proposed to obtain a near-optimal layout design in an expeditious manner.

The rest of the paper is organized as follows. We first describe the problem and assumptions, followed by a 2D $G$-4 based heuristic to handle the loading problem at a layer's level. This 2D layout design heuristic is then integrated into a static layer-based heuristic for loading homogeneous boxes into a container. The iterative heuristic is furnished next, followed by comparative studies between our heuristics and a demo version of a commercial software package. The paper concludes with some remarks.

## PROBLEM FORMULATION AND ASSUMPTIONS

The heuristic algorithms are designed to load homogeneous rectangular boxes into a single container with the objective of maximizing the number of boxes that can be loaded into the container (Pisinger, 2002). Alternatively, our objective is to find a loading scheme such that

$$\max_N \left\{ \frac{N\cdot(l\cdot w\cdot h)}{LL\cdot WW\cdot HH}\times 100\% \right\}. \tag{1}$$

Similar to the assumptions given in (Ngoi *et al.*, 1994; Li and Ye, 2002), our heuristics assume that:

(i) All goods are packaged in homogeneous rectangular cardboard boxes only. Other shapes are excluded.

(ii) All items are shipped to the same destination and, hence, there is no need to prioritize the loading sequence.

(iii) All boxes can be rotated about any of the three dimensions, i.e., there is no arrow to indicate that a particular side must face up.

(iv) Limited bearing strength (Bischoff, 2006) is not considered and all boxes are assumed to be strong and firm enough to fit into the loading scheme generated by the algorithm.

(v) Weight distributions of individual boxes and layers are relatively uniform so that it is not necessary to concern about the loading stability (Davies and Bischoff, 1999).

(vi) Boxes are orthogonally packed into a container (De Cani, 1978), i.e., when a box is placed in the container, its edges are either parallel or orthogonal to those of the container.

These assumptions make the CLP more tractable and fit into many practical situations in the real world. As both static and iterative 3D loading heuristics deal with the arrangement of boxes at a layer's level, we first present a heuristic to handle the 2D layout design problem.

## A HEURISTIC FOR 2D LAYOUT DESIGN

This heuristic is proposed to handle the layout design at the layer's level (Wang and Li, 2007).

### Problem statement

For this 2D loading problem, the concern is to arrange as many small rectangles ($l$, $w$) as possible onto a large rectangle ($L$, $W$) (Scheithauer and Terno, 1996), that is,

$$\max_{n}\left\{\frac{n\cdot(l\cdot w)}{L\cdot W}\times 100\%\right\}. \quad (2)$$

Let $L$ and $l$ be the longer edges of the large and small rectangles, and $W$ and $w$ be the shorter ones, respectively. Note that $L$ and $W$ here are not necessarily the same $L$ and $W$ of the (residual) container space, while $l$ and $w$ are not always the same $l$ and $w$ of the boxes. Given the above Assumption (vi), each small rectangle has two possible orientations in the large rectangle: $l$ parallel to $L$ (referred to as orientation 1 hereafter) and $w$ parallel to $L$ (referred to as orientation 2 hereafter).

### Problem analysis

Li and Ye (2002) put forward a block-based heuristic with the so-called $G$-4 structure (Scheithauer and Terno, 1996). A close examination of the heuristic of (Li and Ye, 2002) indicates that their layout design may result in space waste in the middle of the large rectangle surrounded by the four possible blocks as shown in Fig.1 in (Li and Ye, 2002). A modified heuristic is developed here by revising the ranges of some block sizes. The graphical illustration of our layout is shown in Fig.1, which is different from that in (Li and Ye, 2002) and is designed to eliminate space waste in the middle.

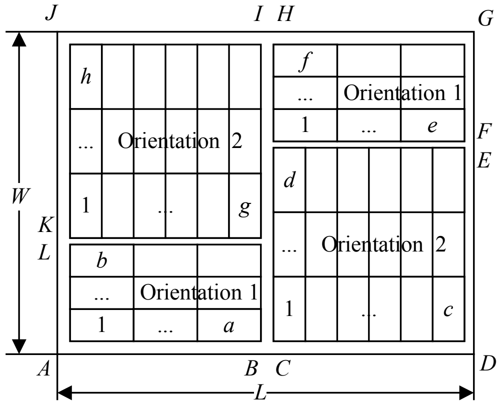


**Fig.1 A two-dimensional layout structure**

Fig.1 illustrates the essence of this heuristic. Boxes with the same orientations are merged into a large block, and at most four blocks may be constructed with each starting from one corner of the large rectangle. It is possible that one block may be extended from one corner all the way to the other corner along a particular edge. It is clear that small rectangles in block 1 starting from corner $A$ are placed as per orientation 1, that $a$ small rectangles are laid side-by-side from $A$ to $B$ along edge $AD$, and that $b$ small rectangles are placed side-by-side from $A$ to $L$ along $AJ$ with remaining boxes being organized in the similar pattern to make $BL$ a rectangular block. On the other hand, in block 2 starting from the corner $D$, orientation 2 is applied to each small rectangle, and $d$ rows and $c$ columns of small rectangles are arranged in this orientation. Similarly, block 3 starting from $G$ consists of $e$ columns and $f$ rows of small rectangles with orientation 1, and block 4 starting from $J$ contains $g$ columns and $h$ rows of small rectangles with orientation 2. Next the ranges of $a$, $b$, $c$, $d$, $e$, $f$, $g$, and $h$ are analyzed.

(1) Ranges of $a$ and $b$

As $a$ represents the number of columns of small rectangles in orientation 1 from $A$ to $B$ along $AD$ and $b$

stands for the number of rows from $A$ to $L$ along $AJ$, it is clear that

$$0 \le a \le \text{int}(L/l), \quad (3)$$

$$0 \le b \le \text{int}(W/w), \quad (4)$$

where int($x$) is the floor function that takes the largest integer less than or equal to $x$. It is understandable that $a$ and $b$ are both 0 if any one assumes zero.

(2) Ranges of $c$ and $d$

As $c$ is the number of columns of small rectangles as per orientation 2, it is fixed after the value of $a$ is determined

$$c = \text{int}((L - al)/w). \quad (5)$$

For $d$, the only constraint is $W$ as shown below:

$$0 \le d \le \text{int}(W/l). \quad (6)$$

(3) Ranges of $e$ and $f$

This procedure assumes that the aggregated length $GH$ of block 3 is no longer than $CD$, therefore, the values of $e$ and $f$ are determined as

$$e = \text{int}(cw/l), \quad (7)$$

$$f = \text{int}((W - dl)/w). \quad (8)$$

(4) Ranges of $g$ and $h$

When it comes to the last block, the number of rows and columns are given by

$$g = \text{int}(al/w), \quad (9)$$

$$h = \text{int}((W - bw)/l). \quad (10)$$

It is clear that the total number of small rectangles ($l$, $w$) that can be arranged onto ($L$, $W$) is $ab+cd+ef+gh$. Therefore, this 2D layout design is converted to an optimization problem such that

$$\max_{a,b,d} \{n = ab + cd + ef + gh\}. \quad (11)$$

**2D layout heuristic**

Given these analyses, the 2D layout design heuristic is formulated as follows:

(1) Input the values of $L$, $W$, $l$, $w$ and compute $a_{\max}$= int($L/l$), $b_{\max}$=int($W/w$), and $n_0=a_{\max}\cdot b_{\max}$.

(2) Let $a$=0.

(3) If $a$=0, then $b$=0; otherwise $b$=1.

(4) Calculate $c$ as per Eq.(5).

(5) If $c$=0, then $d$=0; otherwise $d$=1.

(6) Determine the values of $e$ and $f$ according to Eqs.(7) and (8). If any of $e$ and $f$ is 0, both are set to be 0.

(7) Compute $g$ and $h$ by Eqs.(9) and (10). If any of them is 0, both are set zero.

(8) Calculate $n=ab+cd+ef+gh$, if $n<n_0$, go to (9); otherwise $n=n_0$, and record the values of $a$, $b$, $c$, $d$, $e$, $f$, $g$, and $h$ and go to (9).

(9) If $d$<int($W/l$), then $d=d+1$, and go back to (6); otherwise go to (10).

(10) If $b$<int($W/w$), then $b=b+1$, and go back to (4); otherwise go to (11).

(11) If $a$<int($L/l$), then $a=a+1$, and go back to (3); otherwise go to (12).

(12) The program terminates and the final values of $n_0$, $a$, $b$, $c$, $d$, $e$, $f$, $g$, and $h$ are the layout design parameters.

## A STATIC LAYER-LAYOUT-BASED HEURISTIC

Fig.2 puts a container in a 3D coordinate system. As mentioned earlier, the first heuristic is a static layer-based algorithm for the CLP with homogeneous boxes. In addition to the six assumptions in the "PROBLEM FORMULATION AND ASSUMPTIONS" section, it is further assumed that:

(vii) When a face of the container is selected to build layers, the 2D layout design for a layer is always referred to as the horizontal plan, and the across-layer structure is always called the vertical pattern.

(viii) When layers are constructed, all individual layers with the same box dimension as the vertical height are placed together and merged into an aggregated layer. The height of an aggregated layer depends on the number of boxes that are vertically stacked one on top of another and which box dimension is fixed as the vertical height.

(ix) Within an aggregated layer, all individual layers follow the same 2D layout design as determined by the aforesaid algorithm.

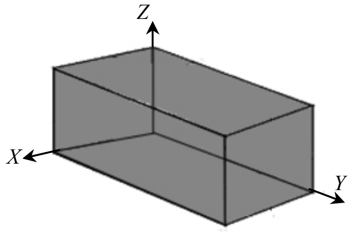


**Fig.2 Container space in a 3D coordinate system**

For a general case that three box dimensions take three different values, Assumption (iii) in the "PROBLEM FORMULATION AND ASSUMPTIONS" section and Assumption (viii) here together indicate that at most three aggregated layers (with $l$, $w$, $h$ as individual layer heights) may be present in the final loading scheme.

The procedure in the previous section handles the horizontal layout design at the layer's level. Two remaining questions are to be addressed here: the container face upon which layers are to be constructed and the vertical structure across different aggregated layers.

Any of the three faces of the container may be selected for building layers. Consistent with Fig.2, when the face $XY$ is chosen, layers are to be stacked from the bottom to the top; if $XZ$ is selected, layers are to be constructed from the right to the left; while layers are to be formed from the back to the front had $YZ$ been chosen.

When the layer face is determined, the vertical structure may consist of up to three aggregated layers with the three box dimensions being individual layer heights. A key concern is to specify the height of each aggregated layer.

When boxes are loaded into the container, nine possible combinations may arise and must be considered, depending on which container face is selected for layers and which box dimension is fixed as the individual layer height. For notational convenience, let $D_j^i$ be the number of individual layers built vertically along the axis $i$ ($i$=$X$, $Y$, $Z$) with $j$ ($j$=$l$, $w$, $h$) being fixed as the individual layer height, and $N_k^i$ be the maximal number of boxes arranged in a layer based on the layout design procedure in the previous section, where the large rectangle is the layer face that is orthogonal to axis $i$ and small rectangles are specified by $k$ ($k$=$lw$, $lh$, $wh$). The nine combinations can then be categorized into three groups based on the container face for building layers as explained below:

(1) The container face $XY$ is selected as the layer face. In this case, layers are built vertically along the $Z$ axis. When the box dimension $h$ is fixed for the individual layer height, the total number of boxes in the aggregated layer is $N_{lw}^Z \cdot D_h^Z$. If the box dimension $w$ is fixed as the individual layer height, the total number of boxes in the aggregated layer is given by $N_{lh}^Z \cdot D_w^Z$. Similarly, if the box dimension $l$ is fixed as the individual layer height, the total number of boxes in the aggregated layer is $N_{wh}^Z \cdot D_l^Z$.

(2) The container face $XZ$ is chosen as the layer face and layers are formed along the $Y$ axis. In this case, if the box dimension $h$ is parallel to $Y$, then the total number of boxes in the aggregated layer is $N_{lw}^Y \cdot D_h^Y$. If the box dimension $w$ is parallel to $Y$, the total number of boxes in the aggregated layer is given by $N_{lh}^Y \cdot D_w^Y$. Similarly, if the box dimension $l$ is parallel to $Y$, the total number of boxes in the aggregated layer is $N_{wh}^Y \cdot D_l^Y$.

(3) When layers are to be stacked along the $X$ axis on the container face $YZ$, the total numbers of boxes in the three aggregated layers are $N_{lw}^X \cdot D_h^X$, $N_{lh}^X \cdot D_w^X$, and $N_{wh}^X \cdot D_l^X$, corresponding to the three box dimensions $h$, $w$, and $l$ being respectively set parallel to the $X$ axis.

The nine combinations are shown in Fig.3.

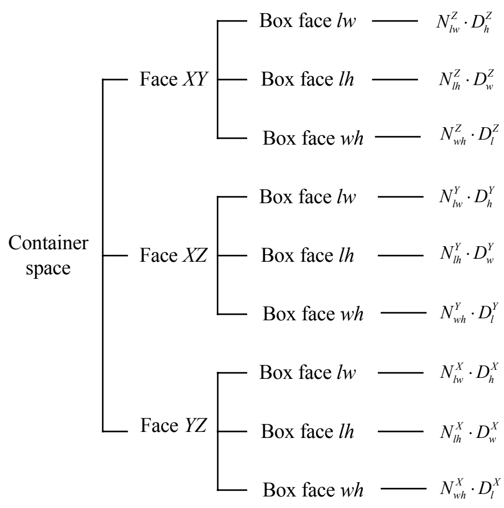


**Fig.3 Illustrations of the nine combinations**

As a matter of fact, the three aforementioned groups correspond to three possible loading schemes. Within each group (or scheme) $i$ ($i$=$X$, $Y$, $Z$), $N_k^i$ is determined by the heuristic in the previous section for each $k$ ($k$=$lw$, $lh$, $wh$). At this stage, the vertical structure across the aggregated layers is concerned with finding an optimal combination of layer heights $D_j^i$ ($j$=$l$, $w$, $h$), such that $N_{\max}^i = \max_{D_j^i} \sum_j \sum_k N_k^i \cdot D_j^i$.

After the three loading schemes are obtained, the optimal solution is determined by $N=\max_i\left\{N^i_{\max}\right\}$.

More specifically, the two-level optimization problems, vertical structure design and scheme selection, are described in detail below.

The vertical structure design optimization is conducted for each axis $X$, $Y$, and $Z$ separately.

(1) Find $N^Z_{\max}$ such that

$$N^Z_{\max}=\max_{D^Z_l,D^Z_w,D^Z_h}\left\{N^Z_{lw}\cdot D^Z_h+N^Z_{lh}\cdot D^Z_w+N^Z_{wh}\cdot D^Z_l\right\}$$

$$\text{s.t.}\begin{cases}0\le D^Z_h\le \text{int}(H/h),\ 0\le D^Z_w\le \text{int}(H/w),\\ 0\le D^Z_l\le \text{int}(H/l),\ D^Z_h\cdot h+D^Z_w\cdot w+D^Z_l\cdot l\le H.\end{cases}$$

(2) Find $N^Y_{\max}$ such that

$$N^Y_{\max}=\max_{D^Y_l,D^Y_w,D^Y_h}\left\{N^Y_{lw}\cdot D^Y_h+N^Y_{lh}\cdot D^Y_w+N^Y_{wh}\cdot D^Y_l\right\}$$

$$\text{s.t.}\begin{cases}0\le D^Y_h\le \text{int}(W/h),\ 0\le D^Y_w\le \text{int}(W/w),\\ 0\le D^Y_l\le \text{int}(W/l),\ D^Y_h\cdot h+D^Y_w\cdot w+D^Y_l\cdot l\le W.\end{cases}$$

(3) Find $N^X_{\max}$ such that

$$N^X_{\max}=\max_{D^X_l,D^X_w,D^X_h}\left\{N^X_{lw}\cdot D^X_h+N^X_{lh}\cdot D^X_w+N^X_{wh}\cdot D^X_l\right\}$$

$$\text{s.t.}\begin{cases}0\le D^X_h\le \text{int}(L/h),\ 0\le D^X_w\le \text{int}(L/w),\\ 0\le D^X_l\le \text{int}(L/l),\ D^X_h\cdot h+D^X_w\cdot w+D^X_l\cdot l\le L.\end{cases}$$

The last step is to choose the maximal scheme from these three values, $N=\max\{N^X_{\max},N^Y_{\max},N^Z_{\max}\}$. The corresponding vertical design $D^*_l$, $D^*_w$, $D^*_h$ and horizontal design parameters, $N^*_{lw}$, $N^*_{lh}$, $N^*_{wh}$ and $a^*$, $b^*$, $c^*$, $d^*$, $e^*$, $f^*$, $g^*$, $h^*$, are obtained accordingly.

## AN ITERATIVE LAYER-LAYOUT-BASED HEURISTIC

In the static heuristic, once a container face is determined for building layers, all layers are to be stacked on the same layer face. In this iterative heuristic, instead of fixing the layer face once for all, it is found out dynamically after layers are loaded into the container iteratively.

Initially, the current space is the full container space, characterized as $LL\times WW\times HH$ along the three axes $X$, $Y$, and $Z$ as shown in Fig.2. After a layer is loaded, either on the bottom face, on the back face, or on the right face, one dimension of the current space will be reduced by the layer height while the other two dimensions remain the same and the exact reduction of the dimension depends on which of the three box dimensions ($l$, $w$, or $h$) is fixed as the layer height. After a layer is packed the residual space is then set as the new current space, denoted by $L\times W\times H$ along $X$, $Y$, and $Z$, for the next step. The iteration continues until the residual space is too small to hold any additional boxes.

To determine a layer to be placed in the current space, nine combinations as shown in Fig.3 have to be examined. For each combination, the 2D heuristic introduced earlier is called for handling the layout design for the layer. Each of the nine layout designs results in a base-area loss, and a so-called minimal base-area loss approach is employed to choose the combination that minimizes the loss. This minimal base-area loss approach can be conveniently depicted in Table 1. Note that $N^i_k$ represents the number of boxes that can be placed in a layer as per the combination of $i$ and $k$, and this value is obtained by applying the 2D layout design heuristic presented earlier. For a current space $L\times W\times H$, the minimal base-area approach finds $i^*$ ($i^*\in\{X, Y, Z\}$, corresponding to the axis that is orthogonal to the layer face) and $k^*$ ($k^*\in\{lw, lh, wh\}$, corresponding to the box face to be placed on the layer face) such that $N^{i^*}_{k^*}$ minimizes the nine values in the last column of Table 1.

**Table 1 Nine combinations and base-area losses**

| Residual space | Layer face | Box face ($k$) | $N^i_k$ | Base-area loss |
|---|---|---|---|---|
| $L\times W\times H$ | $XY$ $i=Z$ | $lw$ | $N^Z_{lw}$ | $L\cdot W-l\cdot w\cdot N^Z_{lw}$ |
| | | $lh$ | $N^Z_{lh}$ | $L\cdot W-l\cdot h\cdot N^Z_{lh}$ |
| | | $wh$ | $N^Z_{wh}$ | $L\cdot W-w\cdot h\cdot N^Z_{wh}$ |
| | $XZ$ $i=Y$ | $lw$ | $N^Y_{lw}$ | $L\cdot H-l\cdot w\cdot N^Y_{lw}$ |
| | | $lh$ | $N^Y_{lh}$ | $L\cdot H-l\cdot h\cdot N^Y_{lh}$ |
| | | $wh$ | $N^Y_{wh}$ | $L\cdot H-w\cdot h\cdot N^Y_{wh}$ |
| | $YZ$ $i=X$ | $lw$ | $N^X_{lw}$ | $W\cdot H-l\cdot w\cdot N^X_{lw}$ |
| | | $lh$ | $N^X_{lh}$ | $W\cdot H-l\cdot h\cdot N^X_{lh}$ |
| | | $wh$ | $N^X_{wh}$ | $W\cdot H-w\cdot h\cdot N^X_{wh}$ |

To facilitate the description of the iterative heuristic, the following notation is introduced. Let $L\times W\times H$ be the three dimensions of the current space along the $X$, $Y$, and $Z$ axes, respectively, and $f(L, W, H)$ be the optimal solution (or the maximum number of boxes that can be loaded in the space $L\times W\times H$) obtained from the static heuristic. Furthermore, denote by $N_{k^*}^{i^*}(L,W,H)$ the number of boxes that can be placed in a layer in the space $L\times W\times H$ as per the minimal base-area loss approach, and let $l_{k^*}^{i^*}, w_{k^*}^{i^*}, h_{k^*}^{i^*}$ be the values that a box takes up along the $X$, $Y$, and $Z$ axes as per the combination of $i^*$ and $k^*$. Now the iterative process can be described as follows:

(1) Input the dimensions of the container and the boxes as *LL*, *WW*, *HH* and *l*, *w*, *h*.
(2) Initialize the iteration.
  (i) *LayBoxNum*=0; // # of boxes loaded as per layers
  (ii) *Rcsx*=*LL*; *Rcsy*=*WW*; *Rcsz*=*HH*;
   // Residual space size
  (iii) *Max_N*=*f*(*LL*, *WW*, *HH*);
   // Iteration starts from the optimal solution of the static heuristic
  (iv) *j*=1; // loop variable
(3) If (*Rcsx*<min(*l*, *w*, *h*)) or (*Rcsy*<min(*l*, *w*, *h*)) or (*Rcsz*<min(*l*, *w*, *h*)) then goto (5).
   // Residual space is too small for any additional boxes
(4) Determine the following values:
  (i) $N_{k^*}^{i^*}(Rcsx, Rcsy, Rcsz)$ and the corresponding $l_{k^*}^{i^*}$, $w_{k^*}^{i^*}$, $h_{k^*}^{i^*}$;
  (ii) *LayBoxNum*=*LayBoxNum*+ $N_{k^*}^{i^*}$ (*Rcsx*, *Rcsy*, *Rcsz*);
  (iii) Reduction of the current space along the three dimensions is calculated as per:

$$L_j^{i_j^*} = \begin{cases} l_{k_j^*}^{i_j^*}, & \text{if } i_j^* = X, \\ 0, & \text{otherwise}, \end{cases}$$

$$W_j^{i_j^*} = \begin{cases} w_{k_j^*}^{i_j^*}, & \text{if } i_j^* = Y, \\ 0, & \text{otherwise}, \end{cases}$$

$$H_j^{i_j^*} = \begin{cases} h_{k_j^*}^{i_j^*}, & \text{if } i_j^* = Z, \\ 0, & \text{otherwise}. \end{cases}$$

  (iv) $Rcsx=Rcsx-L_j^{i_j^*}$; $Rcsy=Rcsy-W_j^{i_j^*}$; $Rcsz=Rcsz-H_j^{i_j^*}$;
   // Update the residual space size
  (v) *Temp_N*=*LayBoxNum*+*f*(*Rcsx*, *Rcsy*, *Rcsz*);
   // # of boxes that can be loaded at step *j*
  (vi) If *Temp_N*>*Max_N* then *Max_N*=*Temp_N*;
  (vii) *j*=*j*+1; goto (3).
(5) Output *Max_N* and the loading scheme.

From these steps, one can understand that the iterative heuristic should perform better than the static approach as it starts with the optimal solution of the static heuristic and settles at the solution that maximizes the number of boxes that can be loaded for all iterations. Of course, this benefit is achieved at a cost: the computational complexity increases for the iterative heuristic compared with the static algorithm. However, the computer implementations of both algorithms indicate that calculations can be carried out expeditiously for the numerical illustrations in the next section.

## NUMERICAL ILLUSTRATIONS

These heuristics have been programmed by using Delphi 6.0 and OpenGL on a Pentium PC within the Windows 2000 operating system. This demo program permits users to conduct numerical experiments and demonstrates the efficiency and effectiveness of the algorithms.

The Greatfox Packer is a commercial container loading software package developed by Hangzhou Greatfox Information Technology Inc. In its demo version, the container size is restricted to a specification of *LL*×*WW*×*ZZ*=1180 cm×230 cm×245 cm and the smallest box dimension is set to be 20 cm. Next, we use our program based on the heuristics in this paper to conduct a comparative study with the demo version of the Greatfox Packer. In this study, the base box specification is set as *l*×*w*×*h*=53 cm×48 cm×37 cm and, at an increment of 5 cm, *l* increases from 53 cm to 63 cm, *w* varies from 48 cm to 58 cm, and *h* changes from 37 cm to 47 cm. In total, 27 combinations are examined and compared with the results obtained with the Greatfox Packer as shown in Table 2.

Table 2 indicates that our heuristics achieve at least the same container volume utilization as the algorithm adopted in the Greatfox Packer in 24 out of 27 cases. As a matter of fact, it is easy to tell that the comparison is on a matched-pair basis, so we can conduct hypothesis tests (Keller, 2005) to verify whether our heuristics indeed outperform the algorithm used in the Greatfox Packer. In this case, our null and alternative hypotheses are

$$H_0{:}\mu_D=0,\ H_1{:}\mu_D>0.$$

**Table 2 Comparative results with the Greatfox Packer**

| Case | Box dimension (cm) | | | Container volume utilization (%) | | |
|---|---|---|---|---|---|---|
| | $l$ | $w$ | $h$ | Greatfox | Static | Iterative |
| 1 | 53 | 48 | 37 | 96.83 | 96.97 | 97.82 |
| 2 | 58 | 48 | 37 | 97.29 | 97.44 | 97.44 |
| 3 | 63 | 48 | 37 | 96.59 | 96.76 | 96.76 |
| 4 | 53 | 53 | 37 | 89.41 | 89.41 | 89.41 |
| 5 | 58 | 53 | 37 | 96.30 | 96.13 | 96.13 |
| 6 | 63 | 53 | 37 | 94.57 | 95.13 | 96.06 |
| 7 | 53 | 58 | 37 | 96.30 | 96.13 | 96.13 |
| 8 | 58 | 58 | 37 | 94.34 | 94.34 | 94.34 |
| 9 | 63 | 58 | 37 | 95.16 | 95.97 | 95.97 |
| 10 | 53 | 48 | 42 | 97.06 | 97.70 | 97.70 |
| 11 | 58 | 48 | 42 | 96.37 | 96.72 | 96.72 |
| 12 | 63 | 48 | 42 | 93.98 | 93.59 | 93.59 |
| 13 | 53 | 53 | 42 | 92.97 | 92.97 | 92.97 |
| 14 | 58 | 53 | 42 | 96.31 | 97.08 | 97.08 |
| 15 | 63 | 53 | 42 | 93.22 | 93.22 | 93.85 |
| 16 | 53 | 58 | 42 | 96.31 | 97.08 | 97.08 |
| 17 | 58 | 58 | 42 | 92.64 | 92.64 | 92.64 |
| 18 | 63 | 58 | 42 | 92.09 | 94.63 | 94.63 |
| 19 | 53 | 48 | 47 | 89.91 | 89.91 | 89.91 |
| 20 | 58 | 48 | 47 | 93.67 | 93.67 | 93.67 |
| 21 | 63 | 48 | 47 | 93.19 | 93.19 | 93.19 |
| 22 | 53 | 53 | 47 | 89.75 | 89.75 | 89.75 |
| 23 | 58 | 53 | 47 | 95.39 | 95.60 | 95.60 |
| 24 | 63 | 53 | 47 | 93.70 | 95.11 | 95.11 |
| 25 | 53 | 58 | 47 | 95.39 | 95.60 | 95.60 |
| 26 | 58 | 58 | 47 | 91.07 | 91.07 | 91.07 |
| 27 | 63 | 58 | 47 | 92.46 | 93.76 | 93.76 |
| Average | | | | 94.16 | 94.50 | 94.59 |

This hypothesis test is conducted three times between the static heuristic and the Greatfox Packer, the iterative heuristic and the Greatfox Packer, and the iterative and the static heuristic, respectively. In these experiments, the sample size is $n$=27. At a significance level of $\alpha$=5%, by looking up a $t$-table, the critical $t$ value to reject the null hypothesis is 1.7056 at a degree of freedom $n$–1=26. The sample mean differences of the volume utilization between the static heuristic and the Greatfox Packer, the iterative heuristic and the Greatfox, the iterative and static heuristic are obtained from the data in Table 2 as $\bar{x}_D = 0.3444,\ 0.4337,\ 0.0892$, respectively. The standard deviations of the differences are computed to be $SD$=0.6196, 0.6613, 0.2609, respectively. According to the $t$-statistic formula (Keller, 2005)

$$t = \frac{\bar{x}_D - \mu_D}{s/\sqrt{n}},$$

one can easily determine the $t$ statistic values of these tests as 2.8884, 3.4076, and 1.7780, respectively. As they are all right-tail tests and the three $t$ statistic values are all greater than the critical $t$ value of 1.7056 at $\alpha$=5%, we infer that there is significant evidence to reject $H_0$ in favour of $H_1$. In the context of this study, there exists sufficient evidence to conclude that: (1) the static heuristic algorithm outperforms the algorithm adopted in the Greatfox Packer; (2) the iterative heuristic outperforms the algorithm adopted in the Greatfox Packer; and (3) the iterative heuristic outperforms the static algorithm.

## CONCLUDING REMARKS

Two layer-layout-based heuristic algorithms are put forward to handle the container-loading problem with homogeneous rectangular boxes. The first heuristic takes a static approach to determine the face for building layers. Layers with the same box dimension as the height are merged into a “thicker” aggregated layer, and the algorithm decides the optimal combination of aggregated layer heights for each layer face (the so-called vertical structure). For each aggregated layer, a 2D layout design procedure is presented to derive an optimal or near-optimal layout pattern, and this pattern (referred to as the horizontal plan) is applied to all individual layers within the same aggregated layer. The algorithm eventually determines an optimal or near-optimal loading scheme that indicates the container face for building layers, the vertical layer structure with three aggregated layer heights, as well as the horizontal plan detailing the box arrangement for each aggregated layer.

The second heuristic selects the layer face in a dynamic fashion: after one layer is placed in the current space of the container, the residual space is treated as a reduced-sized container and the iteration continues until the residual space is too small to hold any more boxes. To determine which face to place a layer and which box dimension is fixed as the layer height, a minimal base-area loss approach is integrated with our 2D layout design heuristic. Numerical studies demonstrate that both proposed algorithms outperform the demo version of a commercial container loading software package in the marketplace in China.

## References


Alvarez-Valdes, R., Parreno, F., Tamarit, J.M., 2005. A tabu search algorithm for the pallet loading problem. *OR Spectrum*, **27**:43-61. [doi:10.1007/s00291-004-0183-5]

Birgin, E.G., Morabito, R., Nishihara, F.H., 2005. A note on an *L*-approach for solving the manufacturer's pallet loading problem. *J. Operat. Res. Soc.*, **56**:1448-1451. [doi:10.1057/palgrave.jors.2601960]

Bischoff, E.E., Ratcliff, M.S., 1995. Issues in the development of approaches to container loading. *OMEGA–Int. J. Manag. Sci.*, **23**(4):377-390. [doi:10.1016/0305-0483(95)00015-G]

Bischoff, E.E., 2006. 3D packing of items with limited load bearing strength. *Eur. J. Operat. Res.*, **168**(3):952-966. [doi:10.1016/j.ejor.2004.04.037]

Bortfeldt, A., Gehring, H., 2001. A hybrid genetic algorithm for the container loading problem. *Eur. J. Operat. Res.*, **131**(1):143-161. [doi:10.1016/S0377-2217(00)00055-2]

Bortfeldt, A., Gehring, H., Mack, D., 2003. A parallel tabu search algorithm for solving the container loading problem. *Parallel Computing*, **29**(5):641-662. [doi:10.1016/S0167-8191(03)00047-4]

Davies, A.P., Bischoff, E.E., 1999. Weight distribution considerations in container loading. *Eur. J. Operat. Res.*, **114**(3):509-527. [doi:10.1016/S0377-2217(98)00139-8]

De Cani, P., 1978. A note on the two-dimensional rectangular cutting-stock problem. *J. Operat. Res. Soc.*, **29**(7):703-706. [doi:10.2307/3009849]

Dyckhoff, H., 1990. A topology of cutting and packing problems. *Eur. J. Operat. Res.*, **44**(2):145-159. [doi:10.1016/0377-2217(90)90350-K]

George, J.A., Robinson, D.F., 1980. A heuristic for packing boxes into a container. *Comput. Operat. Res.*, **7**(3):147-156. [doi:10.1016/0305-0548(80)90001-5]

Jin, Z., Ohno, K., Du, J., 2004. An efficient approach for the three-dimensional container packing problem with practical constraints. *Asia-Pacific J. Operat. Res.*, **21**(3):279-295. [doi:10.1142/S0217595904000254]

Keller, G., 2005. Statistics for Management and Economics (7th Ed). South-Western College, Toronto, p.434-444.

Li, B., Ye, H.Z., 2002. A heuristic layout restriction algorithm for solving two-dimensional rectangular layout loading problems. *J. Southwest Jiao Tong Univ.*, **37**(4):443-447 (in Chinese).

Loh, T.H., Nee, A.Y.C., 1992. A Packing Algorithm for Hexahedral Boxes. Proc. Conf. of Industrial Automation, p.115-126.

Mack, D., Bortfeldt, A., Gehring, H., 2004. A parallel hybrid local search algorithm for the container loading problem. *Int. Trans. Operat. Res.*, **11**(5):511-533. [doi:10.1111/j.1475-3995.2004.00474.x]

Moura, A., Oliveira, J.F., 2005. A GRASP approach to the container-loading problem. *IEEE Intell. Syst.*, **20**(4):50-57. [doi:10.1109/MIS.2005.57]

Ngoi, B.K.A., Tay, M.L., Chua, E.S., 1994. Applying spatial representation techniques to the container packing problem. *Int. J. Prod. Res.*, **32**(1):111-123. [doi:10.1080/00207549408956919]

Pisinger, D., 2002. Heuristics for the container loading problem. *Eur. J. Operat. Res.*, **141**(2):382-392. [doi:10.1016/S0377-2217(02)00132-7]

Pureza, V., Morabito, R., 2005. Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. *Computers and Operations Research*, **33**(3):804-819. [doi:10.1016/j.cor.2004.08.009]

Scheithauer, G., 1992. Algorithm for the Container Loading Problem. Operational Research Proc. 1991, p.445-452.

Scheithauer, G., Terno, J., 1996. The G4-heuristic for the pallet loading problem. *J. Operat. Res. Soc.*, **47**:511-522.

Takahara, S., 2005. Loading Problem in Multiple Containers and Containers and Pallets Using Strategic Search Method. Modeling Decisions for Artificial Intelligence, Proc. Lecture Notes in Artificial Intelligence, **3558**:448-456.

Terno, J., Scheithauer, G., Sommerweiss, U., Riehme, J., 2000. An efficient approach for the multi-pallet loading problem. *Eur. J. Operat. Res.*, **123**(2):372-381. [doi:10.1016/S0377-2217(99)00263-5]

Wang, Z.J., Li, K.W., 2007. A Layer-based Heuristic for the Container Loading Problem with Homogeneous Boxes. Proc. 2nd Int. Conf. on Computer Science and Education, p.233-238.

Wang, Z.J., Li, K.W., Zhang, X.P., 2006. A Heuristic Algorithm for the Container Loading Problem with Heterogeneous Boxes. Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, p.5240-5245.

Wang, Z.J., Li, K.W., Levy, J.K., 2007. A heuristic for the container loading problem: a tertiary-tree-based dynamic space decomposition approach. *Eur. J. Operat. Res.*, in press. [doi:10.1016/j.ejor.2007.08.017]