Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 931092, 13 pages
doi:10.1155/2012/931092

Research Article

Solving Packing Problems by a Distributed Global Optimization Algorithm

Nian-Ze Hu,[1] Han-Lin Li,[2] and Jung-Fa Tsai[3]

[1] *Department of Information Management, National Formosa University, Yunlin 632, Taiwan*
[2] *Institute of Information Management, National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu 300, Taiwan*
[3] *Department of Business Management, National Taipei University of Technology, No. 1, Sec. 3, Chung Hsiao E. Road, Taipei 10608, Taiwan*

Correspondence should be addressed to Jung-Fa Tsai, jftsai@ntut.edu.tw

Received 23 February 2012; Accepted 9 May 2012

Academic Editor: Yi-Chung Hu

Packing optimization problems aim to seek the best way of placing a given set of rectangular boxes within a minimum volume rectangular box. Current packing optimization methods either find it difficult to obtain an optimal solution or require too many extra 0-1 variables in the solution process. This study develops a novel method to convert the nonlinear objective function in a packing program into an increasing function with single variable and two fixed parameters. The original packing program then becomes a linear program promising to obtain a global optimum. Such a linear program is decomposed into several subproblems by specifying various parameter values, which is solvable simultaneously by a distributed computation algorithm. A reference solution obtained by applying a genetic algorithm is used as an upper bound of the optimal solution, used to reduce the entire search region.

1. Introduction

A packing optimization problem is to seek a minimal container, which can hold a given number of smaller rectangular boxes. This problem is also referred to as a container loading problem. Packing cartons into a container is concerning material handling in the manufacturing and distribution industries. For instance, workers in the harbor have to pack more than one type of cartons into a container, and they often deal with this problem by the rule of thumb but a systematic approach. Therefore, the utilization of the container is low , which will cause additional costs.

Similar issues can be found in fields such as knapsack [1, 2], cutting stock [3, 4], assortment problems [5, 6], rectangular packing [7], pallet loading [8, 9], and container loading problems [10, 11]. In addition, researchers have dealt with various related problems. For instance, Dowsland [12] and Egeblad and Pisinger [1], He et al. [13], Wu et al. [14], de Almeida and Figueiredo [15], Miyazawa and Wakabayashi [16], and Crainic et al. [17] proposed different heuristic methods for solving three-dimensional packing problems, Chen et al. [10] formulated a mixed integer program for container loading problems, and Li and Chang [6] developed a method for finding the approximate global optimum of the assortment problem. However, Li and Chang's method [6] requires using numerous 0-1 variables to linearize the polynomial objective function in their model, which would involve heavy computation in solving packing problems. Moreover, Chen et al.'s approach [10] can only find a local optimum of packing problems with nonlinear objective function. Recently, many global optimization methods have been developed, where Floudas' method [18] is one of most promising methods for solving general optimization problems. Although Floudas' method [18] can be applied to solve a packing problem to reach finite ε-convergence to the global minimum, it requires successively decomposing the concave part of the original problem into linear functions. Besides, it adopts lower boundary techniques, which is time consuming. Li et al. [19] designed a distributed optimization method to improve the computational efficiency for solving packing problems. Tsai and Li [20] also presented an enhanced model with fewer binary variables and used piecewise linearization techniques to transform the nonconvex packing problem into a mixed-integer linear problem, which is solvable to find a global solution.

Due to the complexity and hardness of three-dimensional packing problems, most results on this topic are based on heuristics [21]. Furthermore, parallel computing [22] was adopted to improve the efficiency of combinatorial computation. Parallel Genetic Algorithm (GA) [23], parallel with heuristic [24], and parallel Tabu Search Algorithm (TSA) [25] were proposed to solve container-packing problems under some conditions. These methods are capable of obtaining solutions with good performance relative to test examples in the literature. However, the algorithms cannot guarantee to get a global optimum.

This paper proposes another method for finding the optimum of the packing problem. The major advantage of this method is that it can reformulate the nonlinear objective function of original packing problem as an increasing function with single variable and two given parameters. In addition, distributed computation and genetic algorithm are adopted to improve the efficiency and ensure the optimality. The proposed method then solves the reformulated programs by specifying the parameters sequentially to reach the globally optimal solution on a group of network-connected computers.

2. Problem Formulation

Given n rectangular boxes with fixed lengths, widths, and heights, a packing optimization problem is to allocate these n boxes within a rectangular container having minimal volume. Denote x, y, and z as the width, length, and height of the container; the packing optimization problem discussed here is stated as follows:

$$\begin{aligned} &\text{minimize} && xyz \\ &\text{subject to} && (1)\ \text{all of } n \text{ boxes are nonoverlapping.} \end{aligned}$$

$$\begin{aligned}&(2)\ \text{all of } n \text{ boxes are within the range of } x, y, \text{and } z.\\&(3)\ \underline{x} \le x \le \overline{x},\ \underline{y} \le y \le \overline{y},\ \underline{z} \le z \le \overline{z}\\&\left(\underline{x}, \underline{y}, \underline{z},\ \overline{x}, \overline{y}, \overline{z} \text{ are constants}\right).\end{aligned} \tag{2.1}$$

According to Chen et al. [10], the current packing model adopts the terminologies as follows.

(p_i, q_i, r_i): Dimension of box i, p_i is the length, q_i is the width, and r_i is the height, and p_i, q_i, and r_i are integral constants. $i \in J$, $J = \{1, 2, 3, \ldots, n\}$ is the set of the given boxes.

(x, y, z): Variables indicating the length, width, and height of the container.

(x_i, y_i, z_i): Variables indicating the coordinates of the front-left-bottom corner of box i.

(l_{xi}, l_{yi}, l_{zi}): Binary variables indicating whether the length of box i is parallel to the X-axis, Y-axis, or Z-axis. The value of l_{xi} is equal to 1 if the length of box i is parallel to the X-axis; otherwise, it is equal to 0. It is clear that $l_{xi} + l_{yi} + l_{zi} = 1$.

(w_{xi}, w_{yi}, w_{zi}): Binary variables indicating whether the width of box i is parallel to the X-axis, Y-axis, or Z-axis. The value of w_{xi} is equal to 1 if the width of box i is parallel to the X-axis; otherwise, it is equal to 0. It is clear that $w_{xi} + w_{yi} + w_{zi} = 1$.

(h_{xi}, h_{yi}, h_{zi}): Binary variables indicating whether the width of box i is parallel to the X-, Y-, or Z-axis. The value of h_{xi} is equal to 1 if the height of box i is parallel to the X-axis; otherwise, it is equal to 0. It is clear that $h_{xi} + h_{yi} + h_{zi} = 1$.

For a pair of boxes (i, k), where $i < k$, there is a set of 0-1 vector $(a_{ik}, b_{ik}, c_{ik}, d_{ik}, e_{ik}, f_{ik})$ defined as

$a_{ik} = 1$ if box i is on the left of box k, otherwise $a_{ik} = 0$,

$b_{ik} = 1$ if box i is on the right of box k, otherwise $b_{ik} = 0$,

$c_{ik} = 1$ if box i is behind box k, otherwise $c_{ik} = 0$,

$d_{ik} = 1$ if box i is in front of box k, otherwise $d_{ik} = 0$,

$e_{ik} = 1$ if box i is below box k, otherwise $e_{ik} = 0$,

$f_{ik} = 1$ if box$i$ is above box k, otherwise $f_{ik} = 0$.

The front-left-bottom corner of the container is fixed at the origin. The interpretation of these variables is illustrated in Figure 1. Figure 1 contains two boxes i and k, where box i is located with its length along the X-axis and the width parallel to the Z-axis, and box k is located with its length along the Z-axis and the width parallel to the X-axis. We then have l_{xi}, w_{zi}, h_{yi}, l_{zk}, w_{xk}, and h_{yk} equal to 1. In addition, since box i is located on the left-hand side of and in front of box k, it is clear that $a_{ik} = d_{ik} = 1$ and $b_{ik} = c_{ik} = e_{ik} = f_{ik} = 0$.

According to Chen et al. [10] and Tsai and Li [20], the packing problem can be formulated as follows.

Problem 1.

$$\text{Minimize} \quad \text{Obj} = xyz \tag{2.2}$$

$$\text{subject to} \quad x_i + p_i l_{xi} + q_i w_{xi} + r_i h_{xi} \le x_k + (1 - a_{ik})M \quad \forall i, k \in J,\ i < k, \tag{2.3}$$

$$x_k + p_k l_{xk} + q_k w_{xk} + r_k h_{xk} \le x_i + (1 - b_{ik})M \quad \forall i, k \in J,\ i < k, \tag{2.4}$$

$$y_i + p_i l_{yi} + q_i w_{yi} + r_i h_{yi} \le y_k + (1 - c_{ik})M, \quad \forall i, k \in J,\ i < k, \tag{2.5}$$

$$y_k + p_k l_{yk} + q_k w_{yk} + r_k h_{yk} \le y_i + (1 - d_{ik})M, \quad \forall i, k \in J,\ i < k, \tag{2.6}$$

$$z_i + p_i l_{zi} + q_i w_{zi} + r_i h_{zi} \le z_k + (1 - e_{ik})M, \quad \forall i, k \in J,\ i < k, \tag{2.7}$$

$$z_k + p_k l_{zk} + q_k w_{zk} + r_k h_{zk} \le z_i + \left(1 - f_{ik}\right)M, \quad \forall i, k \in J,\ i < k, \tag{2.8}$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} + e_{ik} + f_{ik} \ge 1, \quad \forall i, k \in J,\ i < k, \tag{2.9}$$

$$x_i + p_i l_{xi} + q_i w_{xi} + r_i h_{xi} \le x, \quad \forall i \in J, \tag{2.10}$$

$$y_i + p_i l_{yi} + q_i w_{yi} + r_i h_{yi} \le y, \quad \forall i \in J, \tag{2.11}$$

$$z_i + p_i l_{zi} + q_i w_{zi} + r_i h_{zi} \le z, \quad \forall i \in J, \tag{2.12}$$

$$l_{xi} + l_{yi} + l_{zi} = 1, \quad \forall i \in J, \tag{2.13}$$

$$w_{xi} + w_{yi} + w_{zi} = 1, \quad \forall i \in J, \tag{2.14}$$

$$h_{xi} + h_{yi} + h_{zi} = 1, \quad \forall i \in J, \tag{2.15}$$

$$l_{xi} + w_{xi} + h_{xi} = 1, \quad \forall i \in J, \tag{2.16}$$

$$l_{yi} + w_{yi} + h_{yi} = 1, \quad \forall i \in J, \tag{2.17}$$

$$l_{zi} + w_{zi} + h_{zi} = 1, \quad \forall i \in J, \tag{2.18}$$

where

$$l_{xi}, l_{yi}, l_{zi}, w_{xi}, w_{yi}, w_{zi}, h_{\text{x}i}, h_{yi}, h_{zi}, a_{ik}, b_{ik}, c_{ik}, d_{ik}, e_{ik}, f_{ik} \text{ are 0-1 variables,} \tag{2.19}$$

$$\begin{gathered} M = \max\{\overline{x}, \overline{y}, \overline{z}\}, x_i, y_i, z_i \ge 1, 1 \le \underline{x} \le x \le \overline{x},\ 1 \le \underline{y} \le y \le \overline{y},\ 1 \le \underline{z} \le z \le \overline{z}, \\ \underline{x}, \underline{y}, \underline{z}, \overline{x}, \overline{y}, \overline{z} \text{ are constants,} \end{gathered} \tag{2.20}$$

$$x, y, z \text{ are positive variables.} \tag{2.21}$$

The objective of this model is to minimize the volume of the container. The constraints (2.3)–(2.9) are nonoverlapping conditions used to ensure that none of these n boxes overlaps with each other. Constraints (2.10)–(2.12) ensure that all boxes are within the enveloping container. Constraints (2.13)–(2.18) describe the allocation restrictions among logic variables. For instance, (2.13) implies that the length of box i is parallel to one of the axes. (2.16) implies that only one of the length, the width, and the height of box i is parallel to X-axis.

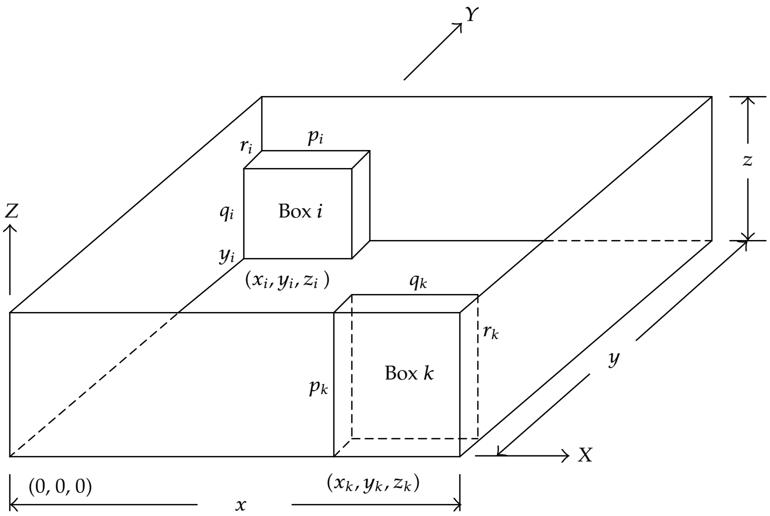

Figure 1: Graphical illustration.

Since the objective function of Problem 1 is a product term, Problem 1 is a nonlinear mixed 0-1 program, which is difficult to be solved by current optimization methods. Chen et al. [10] can only solve linear objective function. Tsai and Li's method [20] can solve Problem 1 at the price of adding many extra 0-1 variables.

3. Proposed Method

Consider the objective function Obj $= xyz$ in (2.2), where $x \geq y \geq z$, $1 \leq \underline{x} \leq x \leq \overline{x}$, $1 \leq \underline{y} \leq y \leq \overline{y}$, $1 \leq \underline{z} \leq z \leq \overline{z}$ and x, y, z are positive variables. Denote r and s as two variables defined as $r = x - y + z$, $s = x - z$. Replace y by $2x - s - r$ and replace z by $x - s$. xyz then becomes Obj' as follows.

$$\text{Obj}' = x\left[2x^2 - (r + 3s)x + rs + s^2\right]. \tag{3.1}$$

We then have the following propositions.

Proposition 3.1. *Suppose r and s in* (3.1) *are fixed values, then Obj' is an increasing function.*

Proof. Since $\partial\text{Obj}'/\partial x = 6x^2 - 2x(r + 3s) + rs + s^2 = xy + yz + 2xz > 0$, it is clear that Obj' is an increasing function. □

Proposition 3.2. *The optimal solution of Problem 1 is integral.*

Proof. Since dimensions of box i, p_i, q_i, r_i, are integral constants for $i = 1, 2, \ldots, n$ and all of n boxes are nonoverlapping, therefore, x^*, y^*, and z^* that indicate the optimal solution of the container must be integral. □

Proposition 3.3. $\sqrt[3]{\sum_{i=1}^{n} p_i q_i r_i} \leq (x+y+z)/3$, *where* p_i *is the length,* q_i *is the width, and* r_i *is the height of the given box* i*. [19, 20]*

Proof. Since $\sum_{i=1}^{n} p_i q_i r_i \leq xyz$ and $\sqrt[3]{xyz} \leq (x+y+z)/3$, then we can have $\sqrt[3]{\sum_{i=1}^{n} p_i q_i r_i} \leq (x+y+z)/3$. □

According to the above propositions and given values of r and s (denoted as $\widehat{r}$ and $\widehat{s}$), consider the following program.

Problem 2.

$$\begin{aligned}
\text{Minimize} \quad & \text{Obj}'(x) = 2x^3 - (\widehat{r} + 3\widehat{s})x^2 + \left(\widehat{r}\widehat{s} + \widehat{s}^2\right)x \\
\text{subject to} \quad & \sqrt[3]{\sum_{i=1}^{n} p_i q_i r_i} \leq \frac{x+y+z}{3}, \\
& \widehat{r} = x - y + z, \quad \widehat{s} = x - z, (2.3) \sim (2.21), \\
& \widehat{r} \text{ and } \widehat{s} \text{ are fixed values.}
\end{aligned} \tag{3.2}$$

Proposition 3.4. *If* $(x^{\Delta}, y^{\Delta}, z^{\Delta})$ *is the solution of Problem 2 found by a genetic algorithm and* (x^*, y^*, z^*) *is the globally optimal solution of Problem 2, then* $x^* \leq x^{\Delta}$.

Proof. Since Obj$'$ is an increasing function with single variable x (following Proposition 3.1) and $2(x^*)^3 - (\widehat{r}+3\widehat{s})(x^*)^2 + (\widehat{r}\widehat{s}+\widehat{s}^2)x^* \leq 2(x^{\Delta})^3 - (\widehat{r}+3\widehat{s})(x^{\Delta})^2 + (\widehat{r}\widehat{s}+\widehat{s}^2)x^{\Delta}$. Hence, $x^* \leq x^{\Delta}$. □

Adding the constraint $x \leq x^{\Delta}$ to Problem 2 for reducing the search region of the optimal solution, we can have the following two programs.

Problem 3.

$$\begin{aligned}
\text{Minimize} \quad & \text{Obj}'(x) = 2x^3 - (\widehat{r} + 3\widehat{s})x^2 + \left(\widehat{r}\widehat{s} + \widehat{s}^2\right)x \\
\text{subject to} \quad & x \leq x^{\Delta}, \\
& \widehat{r} = x - y + z, \quad \widehat{s} = x - z, (2.3) \sim (2.21), (3.2), \\
& \widehat{r} \text{ and } \widehat{s} \text{ are fixed values.}
\end{aligned} \tag{3.3}$$

Problem 4.

$$\begin{aligned}
\text{Minimize} \quad & x \\
\text{subject to} \quad & \text{all the constraints in Problem 3.}
\end{aligned} \tag{3.4}$$

Proposition 3.5. *Let* (x^*, y^*, z^*) *be the global optimum of Problem 3, then* (x^*, y^*, z^*) *is also the global optimum of Problem 4.*

Proof. Since $\text{Obj}'(x)$ is an increasing function with single variable x (following Proposition 3.1), Problems 2 and 3 have the same global optimum (x^*, y^*, z^*). □

According to the above propositions, a packing optimization problem, which is a nonlinear 0-1 programming problem, can be transformed into a linear 0-1 program by introducing two parameters r and s. Then we can guarantee to obtain the global optimum of a packing problem by solving the transformed linear 0-1 programs. The distributed computation scheme is also proposed to enhance the computational efficiency.

4. Distributed Algorithm

The solution procedure for solving Problem 1 to obtain a global optimum is presented in the following with a flow chart shown in Figure 2.

Step 1. (Find an initial solution by GA). From Proposition 3.4, the obtained solution is $(x^{\Delta}, y^{\Delta}, z^{\Delta})$ and the constraint $x \leq x^{\Delta}$ can be utilized to reduce the searching space of the global solution.

Step 2. Denote $s(m)$ and $s(1)$ as the upper and lower bounds of s, respectively. Find the bounds of $s = x - z$ by solving the following linear programs:

$$\begin{aligned} s(m) &= \text{Max}\{x - z \mid \text{subject to } (2.3) \sim (2.21), (3.2)\}, \\ s(1) &= \text{Min}\{x - z \mid \text{subject to } (2.3) \sim (2.21), (3.2)\}. \end{aligned} \tag{4.1}$$

Let $x - z = s(i)$, $i = m$, and go to Step 3.

Step 3. Denote $r(n)$ and $r(1)$ as the upper and lower bounds of r, respectively. Find the bounds of $r = x - y + z$ by solving the following linear programs:

$$\begin{aligned} r(n) &= \text{Max}\{x - y + z \mid \text{subject to } (2.3) \sim (2.21), (3.2),\ x - z = s(i)\}, \\ r(1) &= \text{Min}\{x - y + z \mid \text{subject to } (2.3) \sim (2.21), (3.2),\ x - z = s(i)\}, \end{aligned} \tag{4.2}$$

Let $r = r(j)$, $j = 1$, and go to Step 4.

Step 4. Decompose main problem and perform distributed packing algorithm. According to verity of r and s, the main problem can be decomposed into several subproblems. The transformed subproblem of each iterative process is listed as follows.

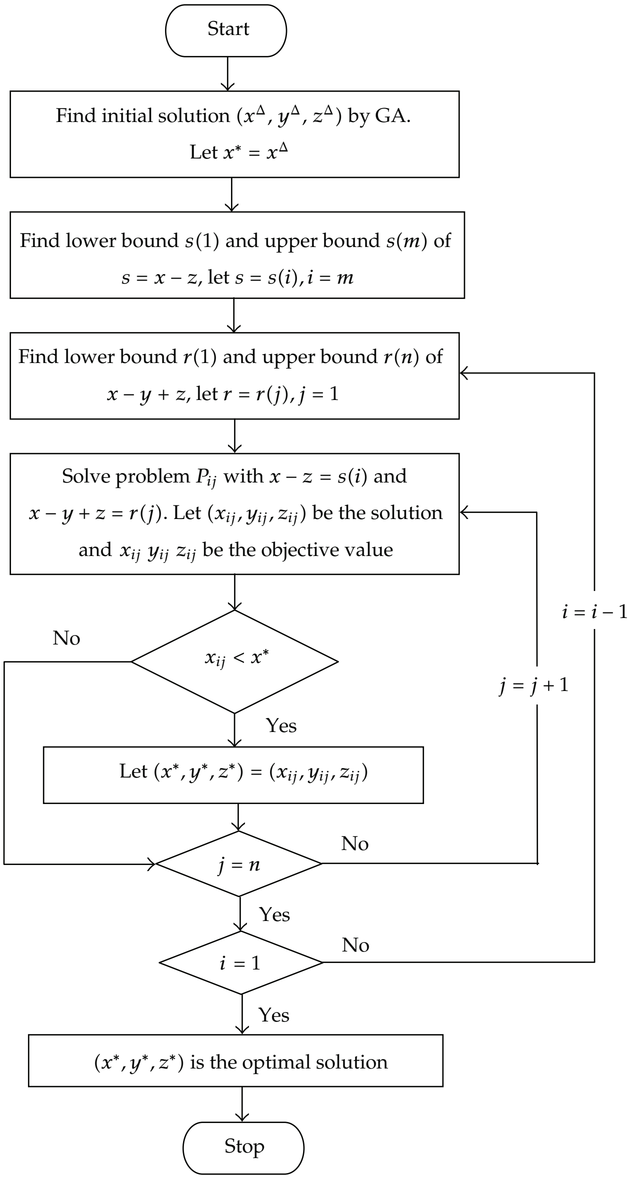
Start
Find initial solution $(x^{\Delta}, y^{\Delta}, z^{\Delta})$ by GA. Let $x^* = x^{\Delta}$
Find lower bound $s(1)$ and upper bound $s(m)$ of $s = x - z$, let $s = s(i), i = m$
Find lower bound $r(1)$ and upper bound $r(n)$ of $x - y + z$, let $r = r(j), j = 1$
Solve problem P_{ij} with $x - z = s(i)$ and $x - y + z = r(j)$. Let (x_{ij}, y_{ij}, z_{ij}) be the solution and $x_{ij}\ y_{ij}\ z_{ij}$ be the objective value
$i = i - 1$
No
$x_{ij} < x^*$
$j = j + 1$
Yes
Let $(x^*, y^*, z^*) = (x_{ij}, y_{ij}, z_{ij})$
No
$j = n$
Yes
No
$i = 1$
Yes
(x^*, y^*, z^*) is the optimal solution
Stop

Figure 2: Solution algorithm.

Problem P_{ij}.

$$\begin{aligned} &\text{Minimize} \quad \text{Obj}_{ij} = x \\ &\text{subject to} \quad (2.3) - (2.21), (3.2), (3.3), x - z = s(i), x - y + z = r\left(j\right), \end{aligned}$$

$$\mathrm{Obj}' = x\left[2x^2 - (r(j) + 3s(i))x + r(j)s(i) + s(i)^2\right],$$
$$1 \le i \le m,\ 1 \le j \le n. \tag{4.3}$$

Every sub-problem can be submitted to client computer and solved independently. Server computer controls the whole process and compares the solutions (x_{ij}, y_{ij}, z_{ij}) of the problem P_{ij} with the initial solution $(x^{\Delta}, y^{\Delta}, z^{\Delta})$. If x_{ij} is smaller than x^{Δ}, then x^{Δ} is replaced by x_{ij}.

The structure of distributed packing algorithm is developed based on star schema. Owing to reduce the network loading and improve the computational performance of each client computer, all results found on all clients are directly sent to host computer.

Step 5. Let $j = j + 1$. If $j > n$, then go to Step 6. Otherwise, go to Step 4.

Step 6. Let $i = i - 1$. If $i < 1$, then go to Step 3. Otherwise, go to Step 7.

Step 7. The whole process is finished and the host computer obtains the optimal solution (x^*, y^*, z^*) with the objective value $x^*y^*z^*$.

5. Numerical Examples

To validate the proposed method, several examples with different number of boxes are solved by LINGO 11.0 [26] with the distributed algorithm. Two of the test problems denoted as Problems 1 and 2 are taken from Chen et al. [10]. The other examples are arbitrarily generated. Solving these problems by the proposed method, the obtained globally optimal solutions are listed in Tables 1 and 3. Comparison results between GA and the proposed method are shown in Table 2, and the associated graphs are presented in Figures 3 and 4.

Packing problems often arise in logistic application. The following example (Problem 5) demonstrates how to apply the proposed algorithm in transportation problem and compare the result with traditional genetic algorithm.

Problem 5. Several kinds of goods are packed into a container so as to deliver to 6 different stores on a trip. The dimensions of width and height of the container are 5 and 4. All goods are packed in cubic boxes, which have three different sizes. In order to take less time during unloading, boxes sent to the same store must be packed together. Different groups of boxes cannot overlap each other. Moreover, the packing order to each group must be ordered of the arriving time to each store. The boxes required to be sent to each store are listed in Table 4. The objective is to determine the smallest dimensions of the container.

Solution 1. The arrangement of boxes can be treated as level assortment. The boxes packed in the same level will be delivered to the same store. After performing the proposed method, list of the optimal solutions are shown in Table 5, and illustrated graph is presented in Figure 5.

Table 1: Computational results.

Problem number	Box number	p_i	q_i	r_i	x_i	y_i	z_i	(x, y, z)	Objective value
1	1	25	8	6	0	0	0	(28, 26, 6)	4368
	2	20	10	5	8	0	0		
	3	16	7	3	8	10	2		
	4	15	12	6	16	11	0		
2	1	25	8	6	10	20	0	(35, 28, 6)	5880
	2	20	10	5	25	0	0		
	3	16	7	3	0	0	3		
	4	15	12	6	10	8	0		
	5	22	8	3	5	0	0		
	6	20	10	4	0	8	0		
3	1	25	8	6	0	10	0	(31, 16, 12)	5952
	2	20	10	5	0	0	0		
	3	16	7	3	9	0	9		
	4	15	12	6	25	0	0		
	5	22	8	3	3	7	9		
	6	20	10	4	0	0	5		
	7	10	8	4	31	0	0		

Table 2: Solution comparison of the proposed algorithm and genetic algorithm (GA).

Problem number	GA		Proposed method	
	(x, y, z)	Objective value	(x, y, z)	Objective value
1 (4 boxes)	(30, 30, 6)	5400	(28, 26, 6)	4368
2 (6 boxes)	(33, 26, 7)	6006	(35, 28, 6)	5880
3 (7 boxes)	(25, 25, 10)	6250	(31, 16, 12)	5952

Table 3: Computational results for all boxes are cubic.

Problem number	Number of cubes	Side $p_i \times q_i \times r_i$	(x, y, z)	CPU time (h:min:s)	Objective value
4	3	$1 \times 1 \times 1$	(8, 5, 5)	00:00:08	200 Global optimum
	3	$2 \times 2 \times 2$			
	1	$3 \times 3 \times 3$			
	1	$5 \times 5 \times 5$			
5	4	$1 \times 1 \times 1$	(8, 6, 5)	00:01:26	240 Global optimum
	2	$2 \times 2 \times 2$			
	3	$3 \times 3 \times 3$			
	1	$5 \times 5 \times 5$			
6	4	$1 \times 1 \times 1$	(8, 8, 5)	00:10:12	320 Global optimum
	3	$2 \times 2 \times 2$			
	4	$3 \times 3 \times 3$			
7	4	$1 \times 1 \times 1$	(12, 7, 5)	01:37:44	420 Global optimum
	3	$2 \times 2 \times 2$			
	3	$3 \times 3 \times 3$			
	1	$4 \times 4 \times 4$			
	1	$5 \times 5 \times 5$			

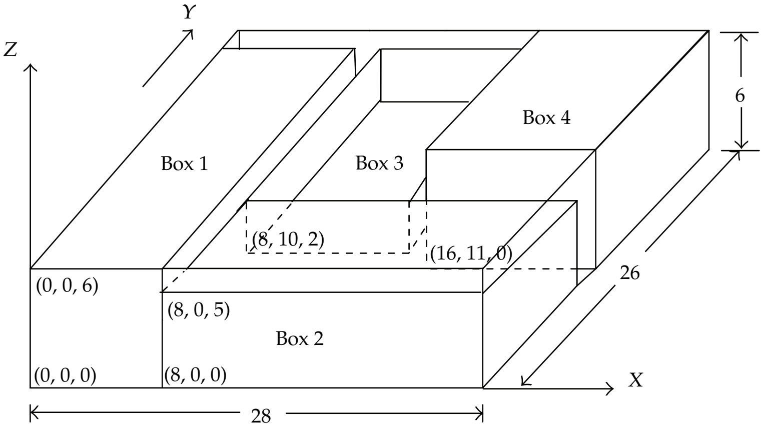

Figure 3: The graphical representation of 4 boxes.

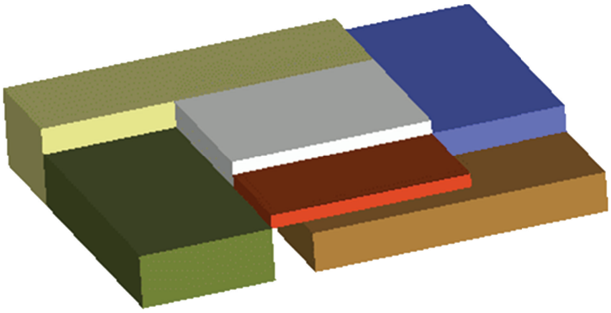

Figure 4: The solid graphical representation of 6 boxes.

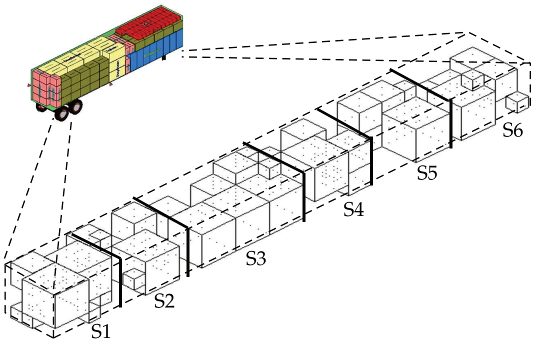

Figure 5: The graphical presentation of the 48 boxes for the 6 stores.

6. Conclusions

This paper proposes a new method to solve a packing optimization problem. The proposed method reformulates the nonlinear objective function of the original packing problem into a linear function with two given parameters. The proposed method then solves the reformulated linear 0-1 programs by specifying the parameters sequentially to reach the globally optimal solution. Furthermore, this study adopts a distributed genetic algorithm and distributed

Table 4: List of stores and boxes (48 boxes).

Store	Goods
S1	A, A, A, B, B, B, B, C
S2	A, A, B, B, B, B, C, C
S3	A, A, A, B, B, B, B, C
S4	A, A, B, B, B, C, C, C
S5	A, A, A, A, B, B, B, C
S6	A, A, A, A, A, B, C, C

A: 1-inch cubic box; B: 2-inch cubic box; C: 3-inch cubic box.

Table 5: List of optimal arrangement of the boxes.

Store	S1	S2	S3	S4	S5	S6
(x_1, y_1, z_1)	A (1, 2, 0)	A (0, 1, 3)	A (4, 0, 2)	A (0, 0, 3)	A (4, 0, 0)	A (0, 0, 0)
(x_2, y_2, z_2)	A (3, 3, 3)	A (5, 0, 0)	A (0, 1, 0)	A (0, 2, 3)	A (4, 1, 3)	A (5, 2, 0)
(x_3, y_3, z_3)	A (0, 4, 0)	B (3, 0, 0)	A (1, 2, 0)	B (3, 0, 1)	A (0, 2, 0)	A (0, 3, 0)
(x_4, y_4, z_4)	B (4, 0, 0)	B (1, 0, 2)	B (3, 0, 0)	B (7, 0, 1)	A (4, 4, 1)	A (2, 4, 0)
(x_5, y_5, z_5)	B (0, 0, 1)	B (3, 0, 2)	B (0, 3, 0)	B (1, 0, 2)	B (2, 0, 0)	A (5, 1, 0)
(x_6, y_6, z_6)	B (2, 0, 0)	B (1, 0, 0)	B (1, 0, 2)	C (0, 2, 0)	B (2, 0, 2)	B (3, 0, 2)
(x_7, y_7, z_7)	C (0, 2, 1)	C (0, 2, 0)	B (0, 3, 2)	C (6, 2, 0)	B (0, 0, 2)	C (0, 1, 1)
(x_8, y_8, z_8)	C (3, 2, 0)	C (3, 2, 0)	C (2, 2, 1)	C (3, 2, 0)	C (1, 2, 0)	C (3, 2, 1)
Dimension of s_i	$6 \times 5 \times 4$	$6 \times 5 \times 4$	$5 \times 5 \times 4$	$9 \times 5 \times 4$	$5 \times 5 \times 4$	$6 \times 5 \times 4$
Volume of s_i	120	120	100	180	100	120
The global solution is (37, 5, 4), and the minimal volume of the container is 740.						

packing algorithm to enhance the computational efficiency. Numerical examples demonstrate that the proposed method can be applied to practical problems and solve the problems to obtain the global optimum.

Acknowledgments

The authors would like to thank the anonymous referees for contributing their valuable comments regarding this paper and thus significantly improving its quality. The paper is partly supported by Taiwan NSC Grant NSC 99-2410-H-027-008-MY3.

References

[1] J. Egeblad and D. Pisinger, “Heuristic approaches for the two- and three-dimensional knapsack packing problem,” *Computers and Operations Research*, vol. 36, no. 4, pp. 1026–1049, 2009.

[2] D. Fayard and V. Zissimopoulos, “An approximation algorithm for solving unconstrained two-dimensional knapsack problems,” *European Journal of Operational Research*, vol. 84, no. 3, pp. 618–632, 1995.

[3] M. Hifi and R. Ouafi, “Best-first search and dynamic programming methods for cutting problems: the cases of one or more stock plates,” *Computers and Industrial Engineering*, vol. 32, no. 1, pp. 187–205, 1997.

[4] J. F. Tsai, P. L. Hsieh, and Y. H. Huang, “An optimization algorithm for cutting stock problems in the TFT-LCD industry,” *Computers and Industrial Engineering*, vol. 57, no. 3, pp. 913–919, 2009.

[5] J. E. Beasley, “An algorithm for the two-dimensional assortment problem,” *European Journal of Operational Research*, vol. 19, no. 2, pp. 253–261, 1985.

[6] H. L. Li and C. T. Chang, "An approximately global optimization method for assortment problems," *European Journal of Operational Research*, vol. 105, no. 3, pp. 604–612, 1998.

[7] J. F. Tsai, P. C. Wang, and M. H. Lin, "An efficient deterministic optimization approach for rectangular packing problems," *Optimization*. In press.

[8] F. H. F. Liu and C. J. Hsiao, "A three-dimensional pallet loading method for single-size boxes," *Journal of the Operational Research Society*, vol. 48, no. 7, pp. 726–735, 1997.

[9] J. Terno, G. Scheithauer, U. Sommerweiß, and J. Riehme, "An efficient approach for the multi-pallet loading problem," *European Journal of Operational Research*, vol. 123, no. 2, pp. 372–381, 2000.

[10] C. S. Chen, S. M. Lee, and Q. S. Shen, "An analytical model for the container loading problem," *European Journal of Operational Research*, vol. 80, no. 1, pp. 68–76, 1995.

[11] G. Scheithauer, "LP-based bounds for the container and multi-container loading problem," *International Transactions in Operational Research*, vol. 6, pp. 199–213, 1999.

[12] W. B. Dowsland, "Three-dimensional packing-solution approaches and heuristic development," *International Journal of Production Research*, vol. 29, no. 8, pp. 1673–1685, 1991.

[13] Y. He, Y. Wu, and R. de Souza, "A global search framework for practical three-dimensional packing with variable carton orientations," *Computers and Operations Research*, vol. 39, no. 10, pp. 2395–2414, 2012.

[14] Y. Wu, W. Li, M. Goh, and R. de Souza, "Three-dimensional bin packing problem with variable bin height," *European Journal of Operational Research*, vol. 202, no. 2, pp. 347–355, 2010.

[15] A. de Almeida and M. B. Figueiredo, "A particular approach for the Three-dimensional Packing Problem with additional constraints," *Computers and Operations Research*, vol. 37, no. 11, pp. 1968–1976, 2010.

[16] F. K. Miyazawa and Y. Wakabayashi, "Three-dimensional packings with rotations," *Computers and Operations Research*, vol. 36, no. 10, pp. 2801–2815, 2009.

[17] T. G. Crainic, G. Perboli, and R. Tadei, "*TS^2PACK*: a two-level tabu search for the three-dimensional bin packing problem," *European Journal of Operational Research*, vol. 195, no. 3, pp. 744–760, 2009.

[18] C. A. Floudas, *Deterministic Global Optimization: Theory, Computational Methods and Applications*, Kluwer Academic Publishers, Dodrecht, The Netherlands, 2000.

[19] H. L. Li, J. F. Tsai, and N. Z. Hu, "A distributed global optimization method for packing problems," *Journal of the Operational Research Society*, vol. 54, no. 4, pp. 419–425, 2003.

[20] J. F. Tsai and H. L. Li, "A global optimization method for packing problems," *Engineering Optimization*, vol. 38, no. 6, pp. 687–700, 2006.

[21] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.

[22] B. Bourbeau, T. Gabriel Crainic, and B. Gendron, "Branch-and-bound parallelization strategies applied to a depot location and container fleet management problem," *Parallel Computing*, vol. 26, no. 1, pp. 27–46, 2000.

[23] H. Gehring and A. Bortfeldt, "A parallel genetic algorithm for solving the container loading problem," *International Transactions in Operational Research*, vol. 9, pp. 497–511, 2002.

[24] J. Btazewicz and R. Walkowiak, "A new parallel approach for multi-dimensional packing problems," *Lecture Notes in Computer Science*, vol. 2328, pp. 585–591, 2006.

[25] A. Bortfeldt, H. Gehring, and D. Mack, "A parallel tabu search algorithm for solving the container loading problem," *Parallel Computing*, vol. 29, no. 5, pp. 641–662, 2003.

[26] LINDO System, *LINGO Release 11.0*, LINDO System, Chicago, Ill, USA, 2008.

Advances in
Operations Research
Advances in
Decision Sciences
Mathematical Problems
in Engineering
Algebra
Journal of
Probability and Statistics
The Scientific
World Journal
International Journal of
Differential Equations
International Journal of
Combinatorics
Advances in
Mathematical Physics
Journal of
Complex Analysis
Journal of
Mathematics
International Journal of
Stochastic Analysis
Abstract and
Applied Analysis
Discrete Dynamics in
Nature and Society
International
Journal of
Mathematics and
Mathematical
Sciences
Journal of
Discrete Mathematics
Journal of
Function Spaces
Journal of
Applied Mathematics
Journal of
Optimization