Applied Mathematical Modelling 38 (2014) 1548–1562

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Efficient approaches for furnace loading of cylindrical parts

Mojahid Saeed Osman [a,*], Bala Ram [b], Funda Samanlioglu [c]

[a] Department of Systems Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[b] Department of Industrial and Systems Engineering, North Carolina A&T State University, Greensboro, NC 27411, USA
[c] Department of Industrial Engineering, Kadir Has University, 34083 Cibali, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 25 February 2011
Received in revised form 4 July 2013
Accepted 16 August 2013
Available online 17 September 2013

Keywords:
Packing problem
Furnace-loading
Mixed Integer Programming

ABSTRACT

This paper addresses the heat treatment operation in a manufacturing plant that produces different types of cylindrical parts. The immediate prior process to heat treatment is furnace-loading, where parts are loaded into baskets. The furnace-loading process is complex and involves issues relating to geometry, and heterogeneity in the parts and in their processing requirements. Currently, furnace-loading is accomplished by operator ingenuity; consequently, the parts loaded in heat treatment often do not use furnace capacity adequately. Efficiency in furnace operation can be achieved by improving basket utilization, which is determined by the furnace-loading process. This paper describes the development of integer and mixed integer LP models for 3D loading of cylindrical parts into furnace baskets. The models consider the exact location of parts to be loaded on the basket and incorporate three models with different objectives; the first addresses the nesting of parts within one another, the second addresses the number of basket layers used, and the third addresses the number of baskets used.

1. Introduction

This research addresses a loading problem of cylindrical parts (bearings) in a manufacturing company. In the heat treatment operation, a batch of part blanks is placed into a rectangular basket and then the basket is processed in the furnace, a batch machine used for the heat treatment operation, in order to impart specified case thickness. Prior to heat treatment operation there are two operations for bearings, ring rolling and turning, and the subsequent operation to heat treatment is the finishing operation which includes grinding, turning and assembly. Often the basket-loading process determines the throughput of the furnace system.

In the process of loading bearings into baskets, a mix of different sizes of bearings is placed in baskets to form layers which are separated by a perforated screens placed on top of the parts. This process is complex and involves issues relating to geometry, and heterogeneity in the parts and their processing requirements. Considering the part external and internal diameters, height, and processing required in the furnace (termed "recipe"), several rules must be followed in loading a basket: (1) Material type: All parts in a basket must be made from the same material; (2) Part height: Parts in a layer must have the same height; (3) Recipe: There are a total of 30 recipes, 15 for each material type. These recipes are characterized by a cycle time, a temperature profile, and a profile for the rate of carbon injection. In a single basket, parts with recipe numbers within a two-number range are permissible; for example, parts with recipe numbers 3 and 4 or 4 and 5 are permitted to be loaded into the same basket; and (4) Nesting: Parts can be nested in a layer; "nesting" implies putting parts within the

* Corresponding author. Tel.: +966 13 860 3162.
E-mail addresses: mfsaeedo@gmail.com, mfsaeedo@kfupm.edu.sa (M. Saeed Osman).

0307-904X/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.apm.2013.08.032

internal diameter of another part. If the internal diameter of the outer part is x millimeters then the external diameter of the inner part must be a maximum of $(x - 50)$ millimeters, to keep 25 mm clearance between the parts.

Currently, the planning department develops a production plan under a "push" strategy approach. The plan assigns priorities to different orders and sequences the orders for processing. Based on this plan the shop-floor picks up orders for processing. Such a plan does not take into account the rules to be adhered to in loading parts for the heat treatment operation. Consequently, the heat treatment department could become a bottleneck in the production process. Furthermore, the loading process is implemented ingenuity by operators such that the batches loaded for heat treatment do not adequately utilize basket and furnace capacities. Thus, an efficient tool is needed to produce three-dimensional loading models that optimally utilize the basket capacity which is critical in achieving furnace throughput maximization/improvement.

The primary objective of this paper is to optimize furnace operations by minimizing the unutilized capacity of the baskets used in the heat treatment operation of the cylindrical parts.

2. Review of the packing problem

In the context of this research, it is relevant to review published research on the packing problem. The problem of packing items onto pallets or into containers in an efficient manner can be computationally complex [1]. Existing approaches to pallet/container loading problems usually apply to a specific class of problem encountered in practice, but there are many scenarios for which no adequate methodologies currently exist. The problem of positioning/packing/loading small items inside bigger spaces which is usually referred to as the "pallet packing/loading problem" arises in several industrial activities. The primary issue in pallet packing has been that of maximizing the area of the pallet used [2]. The general pallet-packing problem can be viewed as a two-dimensional cutting stock problem. The objective is to minimize the waste [3]. Ram [2] provided a survey of research relating to considerations in pallet packing, the models and solution procedures used in pallet packing, and implementation of these approaches in palletization stations. It is generally known that the effectiveness of these approaches can be quite an important economical factor for the success of these industries. Chen et al. [3] developed a mathematical model for palletization problem for boxes that are of varying dimensions with the objective of placing a given set of cartons on minimal number of uniform pallets. The model was guaranteed to lead to an optimal solution.

The three-dimensional packing problem most frequently consists in finding efficient positioning patterns of identical (rectangular or cylindrical) objects on a rectangular base (pallet), where the vertical orientation of the objects is determined by practical constraints. These patterns are repeated for each layer stacked on the pallet. Therefore, the standardization of the object's vertical dimension supports the reduction of the dimensionality of the problem; Dowsland [4] applied the knowledge of the two-dimensional pallet loading problem to the three-dimensional loading problem to maximize the number of equal-sized cartons inside a container without overlapping. When boxes have different base dimensions; some interesting heuristics for pallet loading were considered by Han et al. [5], Abdou and Yang [6], and Abdou and Elmasry [7].

The "cylinder packing problem" is basically concerned with the densest packing of identical circles (cylinders base) inside a rectangle (the pallet). Several papers, have investigated the problem of packing pipes [1], reels and cylinders into containers. Although not able to solve the three-dimensional case the concepts incorporated in these heuristics could be used in solving various aspects of the cylinder packing problem. Several studies addressed the problem of packing circles of the same size into a rectangle including Dowsland [8], and Fraser and George [9]. They depended on fast heuristic algorithms for more complex cases to generate approximate solutions. Isermann [10] proposed some heuristic techniques for packing identical circles in homogeneous patterns developed geometrically, restricting its attention to problems where the product surface area is smaller than the rectangle.

A few authors have studied the problem of packing circles of different sizes into a rectangle [11]. Fraser and George [9] discussed loading reels inside a container in the context of a paper industry, where the relative position circle is chosen among a set of easily stowed pre-defined patterns. Most recently published research also considers the problem of packing circles of different sizes inside a rectangle. Hifi et al. [12] developed a simulated annealing based approach to solve the generic "circular cutting problem". Stoyan and Yaskov [13] proposed a solution method based on a branch-and-bound algorithm and a reduced gradient method to solve a mathematical model of the problem of placement of rectangles and circles in a larger rectangle. George et al. [11] discussed several heuristic approaches and included stability considerations in the context of loading pipes of different diameters inside a container. They also proposed a mixed integer non-linear formulation for the problem of fitting different circles inside a rectangle, which is difficult to solve, even for a small number of pieces, by the current general purpose mixed integer non-linear optimization software packages. Zhang et al. [14] studied the problem of packing different-sized circles into a rectangular container. They formulated this problem as a nonlinear optimization problem and developed a heuristic simulated annealing algorithm for solving this problem. Huang et al. [15] proposed two new heuristics to pack unequal circles into a two-dimensional circular container. The first one, denoted by A1.0, is a basic heuristic which selects the next circle to place according to the maximal hole degree rule. The second one, denoted by A1.5, uses a self look-ahead strategy to improve A1.0.

Babu and Babu [16] proposed a hybrid approach, employing both genetic and heuristic algorithms, for nesting of different rectangular parts in multiple rectangular sheets with the objective of utilizing the sheet material effectively. The proposed genetic approach gave the best sequence of sheets and parts to generate an effective nested pattern with a heuristic algorithm. The heuristic approach arranges each of the parts in the bottom-left-most position of the sheet(s) by considering the sequence of sheets and parts given by the genetic algorithm. Wu et al. [17] proposed hybrid heuristic algorithms for the nesting of two-dimensional rectangular parts in multiple plates. The nesting algorithm of Babu and Babu [16] was first modified and a new heuristic nesting algorithm, IBH, was proposed to utilize the material plate further. IBH is then combined in a meta-heuristic approach, simulated annealing. The proposed hybrid algorithms can then be extended to solve the nesting problem involving irregular parts by embedding irregular parts into rectangles. One problem arises in this 'irregular-to-rectangular' process, which is conversion of demands of the original irregular parts into demands of the embedding rectangles. A greedy heuristic rule is therefore presented to determine the number of embedding rectangles of different types to be used in order to maximize the utilization of the material plate given that the demand of each irregular part must be satisfied. Promising computational results were obtained.

As key findings from reviewing published research on the packing problem, there are a few published research efforts on this subject and, due to its complexity, the approaches to solve the problem are mainly heuristics [1]; Several studies addressed the problem of packing circles into a rectangle, but of the identical circles/same size, they depended on fast heuristic algorithms to generate approximate solutions; A few authors have studied the problem of packing circles of different sizes into a rectangle; All work devoted to three-dimensional packing problems does not present mathematical models for optimizing nesting problem and offer only heuristic solution procedures to handle them. Furthermore, all work devoted to 3D packing problems does not present mathematical models for optimizing nesting problem and offers only heuristic solution procedures to handle them.

The research work presented in this paper makes the following contributions: it proposes promising approach for three-dimensional packing problem of cylindrical parts, the approach calls for reformulation of nonlinear integer programming model into two linear mathematical models to provide an optimal solution to 3D packing problem that considers the exact location of parts into the basket. Additionally, in this paper we will propose parts nesting optimization model along with layer loading, and basket loading models for optimizing a three-dimensional packing problem of cylindrical parts.

3. Furnace loading problem

The furnace loading problem (FLP) can be formulated as a mixed integer non-linear model. The model considers the exact location (coordinate) of parts to be positioned on the basket and the objective function is designed to minimize the number of baskets used. It is assumed that the rectangular basket's base/layer defines a Cartesian coordinate system. We introduce the following notations:

- e_i External diameter of part i
- F Number of layers available for use
- h_i Height of part i
- H Height of basket
- I Number of part types
- L Length of basket/layer
- M An arbitrarily large number
- n_i Internal diameter of part i
- N Number of parts
- P_{ik} Profit of nesting inner part i in outer part k

$$P_{ik} = \begin{cases} \frac{e_i}{n_k} & \text{if} \quad n_k - e_i \geqslant 50 \\ 0 & \text{Otherwise} \end{cases}$$

- Q_i Number of inner parts of type i
- S_{ik} Number of inner parts k to be nested in outer parts i
- U Number of baskets available for use
- W Width of basket/layer

Decision variables

- a_{ikj} The 'left' relative position indicator for parts i and k on layer j; binary, and is 1 if part i is to the left of part k on layer j, and 0 otherwise

$$a_{ikj} = \begin{cases} 1 & \text{if} \quad x_{kj} - e_k \geqslant x_{ij} \\ 0 & \text{Otherwise} \end{cases}$$

- b_{ikj} The 'right' relative position indicator for parts i and k on layer j; binary, and is 1 if part i is to the right of part k on layer j, and 0 otherwise

$$b_{ikj} = \begin{cases} 1 & \text{if} \quad x_{ij} - e_i \geqslant x_{kj} \\ 0 & \text{Otherwise} \end{cases}$$

- c_{ikj} The 'below' relative position indicator for parts i and k on layer j; binary, and is 1 if part i is below part k on layer j, and 0 otherwise

$$c_{ikj} = \begin{cases} 1 & \text{if} \quad y_{kj} - e_k \geqslant y_{ij} \\ 0 & \text{Otherwise} \end{cases}$$

d_{ikj} The 'above' relative position indicator for parts i and k on layer j; binary, and is 1 if part i is above part k on layer j, and 0 otherwise

$$d_{ikj} = \begin{cases} 1 & \text{if} \quad y_{ij} - e_i \geqslant y_{kj} \\ 0 & \text{Otherwise} \end{cases}$$

g_j Height of layer j
r_j Binary, and is 1 if layer j is used, and 0 otherwise
q_{ij} Binary, and is 1 if part i is placed on layer j, and 0 otherwise
u_{jt} Binary, and is 1 if layer j is placed into basket t, and 0 otherwise
x_{ij} The x coordinate of the top right corner of the circumscribing square of part i on layer j
y_{ij} The y coordinate of the top right corner of the circumscribing square of part i on layer j
z_t Binary, and is 1 if basket t is used, and 0 otherwise

As shown in Fig. 1, for any pair of parts i, k ($i \neq k$), the indicator variables are defined to indicate the relative placement of the two parts with respect to each other. The graphical interpretation for the relative position of part i if it is placed to the left, to the right, below, or above part k on layer j is illustrated in Fig. 1(a)–(d) respectively.

When positioning part i to the left of part k on layer j as shown in Fig. 1(a), the 'left' relative position indicator a_{ikj} becomes 1 as the x coordinate of the top right corner of the circumscribing square of part k on layer j (x_{kj}) minus the external diameter of part k (e_k) is equal to the x coordinate of the top right corner of the circumscribing square of part i on layer j (x_{ij}) and the other relative indicators equal to zero. In Fig. 1(b), part i is positioned to the right of part k on layer j, and the x coordinate of the top right corner of the circumscribing square of part i on layer j (x_{ij}) less the external diameter of part i (e_i) is equal to the x coordinate of the top right corner of the circumscribing square of part k on layer j (x_{kj}) and that makes the 'right' relative position indicator b_{ikj} equal 1 and the other relative indicators equal zero. If part i is positioned below part k on layer j as given in Fig. 1(c), the 'below' relative position indicator c_{ikj} would become 1 as the ***y*** coordinate of the top right corner of the circumscribing square of part k on layer j (y_{kj}) minus the external diameter of part k (e_k) is equal to the y coordinate of the top right corner of the circumscribing square of part i on layer j (y_{ij}) and the other relative indicators would equal to zero. Similarly, Fig. 1(d) shows the position of part i above part k on layer j where the ***y*** coordinate of the top right corner

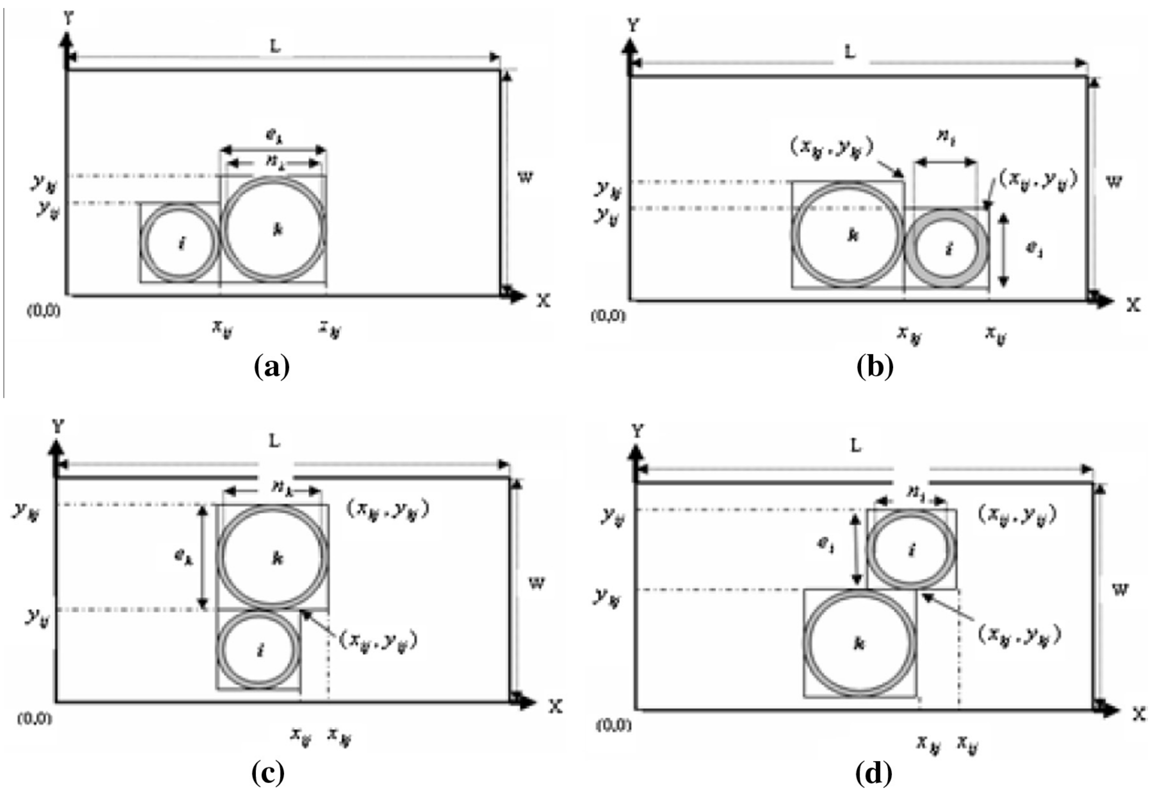

Fig. 1. Graphical interpretation for the relative position of any pair of parts i, k.

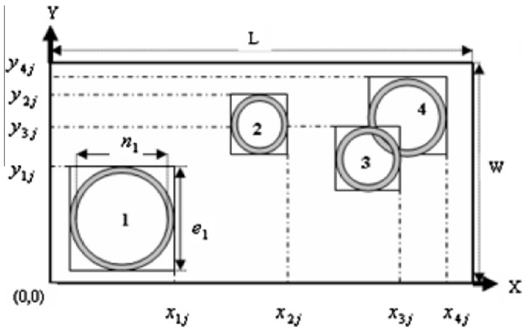

Fig. 2. Graphical illustration for four parts on layer *j*.

Table 1
The values of position indicators for the illustrative example.

Pair of parts		Position indicators				Total
i	k	a_{ikj}	b_{ikj}	c_{ikj}	d_{ikj}	$a_{ikj} + b_{ikj} + c_{ikj} + d_{ikj}$
1	2	1	0	1	0	2
1	3	1	0	0	0	1
1	4	1	0	1	0	2
2	1	0	1	0	1	2
2	3	1	0	0	0	1
2	4	1	0	0	0	1
3	1	0	1	0	0	1
3	2	0	1	0	0	1
3	4	0	0	0	0	[a]0
4	1	0	1	0	1	2
4	2	0	1	0	0	1
4	3	0	0	0	0	[a]0

[a] For any pair of parts placed on the same layer; obtaining a value of zero as a result of summing the position indicators indicates overlapping situation.

of the circumscribing square of part i on layer j (y_{ij}) less the external diameter of part k (e_i) is equal to the y coordinate of the top right corner of the circumscribing square of part k on layer j (y_{kj}) and, as a result, the 'above' relative position indicator d_{ikj} becomes 1 and the other relative indicators become zero.

Considering an example with four parts 1, 2, 3, 4 shown in Fig. 2, the values of the position indicator for each pair of parts (i, k) are shown in Table 1.

The formulation of a non-linear mixed-integer model for the furnace-loading problem (3D-FLP) is given below:

3.1. 3D-FLP Model

$$\text{Minimize}(z) = \sum_{t=1}^{U} z_t LWH - \frac{1}{4}\sum_{i=1}^{N} \pi(e_i^2 - n_i^2)h_i. \tag{1}$$

The objective function minimizes the unutilized volume of the baskets.
Subject to:

- Basket/layer length constraints:

$$x_{ij} \le q_{ij}L \quad \forall i = 1,\ldots,\mathrm{N},\ j = 1,\ldots,\mathrm{F}, \tag{2}$$

$$x_{ij} \ge q_{ij}e_i \quad \forall i = 1,\ldots,\mathrm{N},\ j = 1,\ldots,\mathrm{F}. \tag{3}$$

- Basket/layer width constraints:

$$y_{ij} \le q_{ij}W \quad \forall i = 1,\ldots,N,\ j = 1,\ldots,\mathrm{F}, \tag{4}$$

$$y_{ij} \ge q_{ij}e_i \quad \forall i = 1,\ldots,N, j = 1,\ldots,F. \tag{5}$$

Constraints sets (2)–(5) ensure that all parts are within the physical dimensions of the rectangular layer.

- Part/layer height constraints:

$$g_j = q_{ij} h_i \quad \forall i = 1, \ldots, N,\ j = 1, \ldots, F. \tag{6}$$

Constraint (6) guarantees that all parts on same layer must have same height

- Basket height constraints:

$$\sum_{j=1}^{F} u_{jt} g_j \leq z_t H \quad \forall t = 1, \ldots, U. \tag{7}$$

- Exactly one layer for each part constraint:

$$\sum_{j=1}^{F} q_{ij} = 1 \quad \forall i = 1, \ldots, N. \tag{8}$$

Constraint (8) guarantees that each part will be placed on only one layer.

- Exactly one basket for each layer constraint:

$$\sum_{t=1}^{U} u_{jt} = 1 \quad \forall i = 1, \ldots, F. \tag{9}$$

- Non-overlapping parts constraints:

$$\forall i, k = 1, \ldots, N, j = 1, \ldots, F \text{ and } i \neq k,$$

$$x_{kj} - q_{kj} e_k + (1 - a_{ikj}) M \geqslant x_{ij}, \tag{10}$$

$$x_{ij} - q_{ij} e_i + (1 - b_{ikj}) M \geqslant x_{kj}, \tag{11}$$

$$y_{kj} - q_{kj} e_k + (1 - c_{ikj}) M \geqslant y_{ij}, \tag{12}$$

$$y_{ij} - q_{ij} e_i + (1 - d_{ikj}) M \geqslant y_{kj}, \tag{13}$$

$$a_{ikj} + b_{ikj} + c_{ikj} + d_{ikj} \geqslant q_{ij} + q_{kj} - 1, \tag{14}$$

$$a_{ikj}, b_{ikj}, c_{ikj}, d_{ikj}, q_{ij}, u_{jt} = 0 \text{ or } 1. \tag{15}$$

Constraints sets (10)–(14) ensure that the parts will not be overlapping. It is possible for part i to be either to the 'left of' or to the 'right of' part k. It is also possible for part i to be 'above' or 'below' part k. Constraint (14) ensures that if any pair of parts $((i,k),\ i \neq k)$, is placed in the same layer j, at least one position indicator must be 1 (refer to note below Table 1 for explanation). The above mathematical model contains a non-linear term, (see constraint (7)); therefore, the furnace-loading problem has the form of a non-linear mixed-integer programming problem. The number of constraints is $[5(N * F * (N-1)) + (5 * N * F) + N + F + U]$ and the number of variables is $F(2N-3) + F(N+U+1)]$. For $N = 9$, $F = 3$, and $U = 3$, the number of constraints, and variables are 1230 and 957 respectively.

Nonlinear programming problems generally are more complex and difficult to solve than linear programming problems, and often the solution found is only a local optimum. For reducing the complexity difficulty of solving the three-dimensional furnace-loading problem entirely as a nonlinear mixed integer problem it would be beneficial to solve the problem as two mixed integer LP problems. We propose to break down the entire problem into two subproblems, layer-loading, and basket-loading subproblem, with different objectives. The objective of the first subproblem is to minimize the number of layers used, and the objective of the second subproblem is to minimize the number of baskets used. Additionally, the nonlinear 3D-FLP model does not consider parts nesting, in solving the two subproblems we propose optimization model that considers parts nesting. The two models, layer-loading and basket-loading, together along with the parts nesting model will achieve the same ultimate goal of minimizing the unutilized capacity of the baskets. It must be mentioned here, that the overall approach described in next Section is inspired by the 3D-FLP model but is not equivalent to solving the 3D-FLP model.

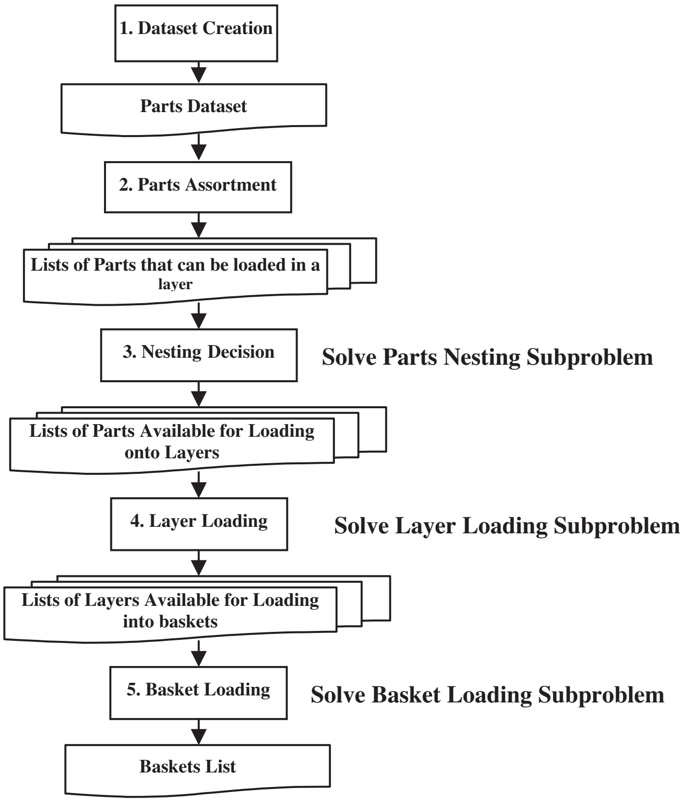

Fig. 3. Steps in solving the furnace loading problem.

4. Proposed approach for solving 3D-FLP

Three subproblems were identified in solving the 3D furnace-loading problem. The general steps for solving the furnace loading problem are presented in Fig. 3, and detailed below:

1. *Create dataset of parts available for loading*: A set of parts will be created from a production plan database associated with turning. The following attributes of the individual parts will be retrieved from the database: part number, material type, external diameter, internal diameter, height, recipe number, and order quantity.
2. *Sort parts dataset:* This step considers the several rules that need to be followed in loading a basket. Lists of parts will be created for heat treatment based on material type, recipe number, and height.
   2.1. Sort the list of parts available for loading by material type. This step is necessary as material types cannot be mixed in a basket.
   2.2. Sort the resulting list from step (2.1) by height. This step will help ensure that only parts with the same height are assigned to a single layer.
   2.3. Sort the lists generated from step (2.2) by recipe number. This step will help ensure that only a two number range of recipe is assigned to a single basket.
3. *Solve the parts nesting subproblem*: This step will help check each list to see if any parts satisfy the nesting criterion, i.e., the external diameter (e_i) of the nested part i is less than the internal diameter (n_k) of any other part k by 50 mm or more (see rules in "Introduction" section) and decide on the best nesting.
4. *Solve the layer-loading subproblem*: The information created in step 3 is used to generate layers. The dimensional constraints of parts and layer are considered to find the candidate parts for the layers to create a layers list.
5. *Solve the basket-loading subproblem*: Each layer list is considered to build baskets. In basket-loading, the idea is to put all of the layers in the minimum number of baskets in such a way that the total height of layers in each basket does not exceed the basket height.

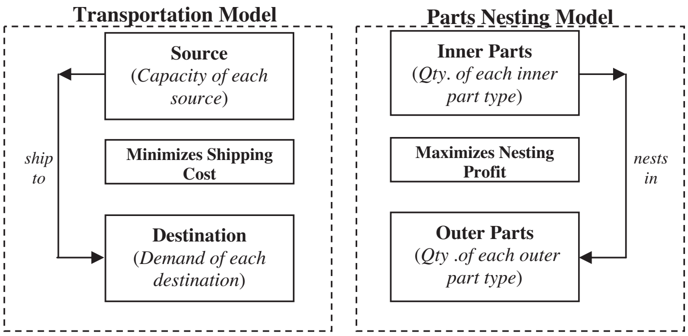

Fig. 4. Analogy between part nesting model and classical transportation model.

5. Parts nesting subproblem

After sorting the dataset (input data) and obtaining information on which parts can be nested, the next step is to decide on the best nesting that maximizes the number of parts nested by maximizing the total nesting profit. To accomplish this task, an optimization model, which resembles the transportation problem as shown in Fig. 4, is formulated.The objective function of this optimization model is to maximize the total nesting profit. The constraints of this model are the quantity of outer and inner parts; note that in the model we can define all part types as both inner and outer parts with the profit parameter P_{ik} defined appropriately. Suppose that we have I inner part types, with Q_i defined to be the quantity for inner part type i; there are K outer part types ($K = I$), also, with Q_k defined to be the quantity for outer part type k. Each inner part type i nested in outer part type k incurs a nesting profit P_{ik}. The formulation of the part nesting subproblem is stated below:

5.1. PN-FLP model:

$$\text{Maximize}(z) = \sum_{i=1}^{I}\sum_{K=1}^{I} S_{ik}P_{ik}. \tag{16}$$

Subject to:

- Inner part quantity constraints:

$$\sum_{k=1}^{I} S_{ik} \leq Q_i \quad \forall i = 1, \ldots, I. \tag{17}$$

- Outer part quantity constraints:

$$\sum_{i=1}^{I} S_{ik} \leq Q_k \quad \forall k = 1, \ldots, I. \tag{18}$$

- Non-negativity constraints:

$$S_{ik} \geq 0 \quad \forall i, k = 1, \ldots, I. \tag{19}$$

The mathematical model of the part-nesting subproblem has an integer programming form. The total number of all variables is equal to $N * (N - 1)$, and the number of constraints is equal to 2 N. The above optimization model is solved using GAMS 22.6 using the CPLEX solver to generate the output that shows the optimum quantity of parts to be nested (the best nesting), based on objective function and underlying constraints.

6. Layer loading subproblem

The layer-loading subproblem involves placing all cylindrical parts (bearings) on a minimum number of layers. At this point, the lists (obtained from the nesting step) of the parts that can be accommodated in a layer are modified. All the inner parts, which are listed as a nested part, are removed from the list because they are now represented as a nested part and not as a primary part. Only the primary parts, the parts with zero nesting level, are considered in creating layers. Once the lists are modified, the next step is to create layers. The dimensional constraints of all parts and each layer are considered to find the candidate parts for the layers, and to create a list of layers that can accommodate all parts. Each part is considered as a square with each side equal to the external diameter of the part. Each layer is a rectangular area with constant width and length.

Fig. 8 shows an example of a layer loading pattern. Fig. 8(a) shows a loading pattern that arises when there is no part that satisfies nesting conditions. In this case, layers consist of unnested parts only. Fig. 8(b) shows a loading pattern with nested parts.

Positioning the candidate parts onto a minimum number of layers can be achieved by modeling the layerloading subproblem as a two-dimensional loading problem. The layer loading subproblem (LLFLP) can be formulated as a mixed linear programming model as described below

6.1. LL-FLP model

$$\text{Minimize}(z) = \sum_{j=1}^{F} r_j. \tag{20}$$

Subject to:

Constraints (2)–(6) and Eq. (8) (from 3D-FLP model)

Layer usage constraint:

$$\sum_{i=1}^{N} q_{ij} \leq r_j M \quad \forall j = 1, \ldots, \mathrm{F}. \tag{21}$$

Constraint (21) ensures that if a layer is not used then no part will be placed on it; here we can use N in place of the large number *M*.

Constraint (10)–(14) (from 3D-FLP model)

$$\sum_{j=1}^{F} r_j \geq \frac{\sum_{i=1}^{N} (e_i)^2}{LW}, \tag{22}$$

$$r_j = 0 \text{ or } 1 \quad \forall j = 1, \ldots, \mathrm{F}. \tag{23}$$

The mathematical model of the layer-loading subproblem has the form of a linear mixed-integer programming problem. The total number of all variables is equal to $4(N * F * (N - 1)) + (3 * N * F) + 2F$, and the number of constraints is equal to $5(N * F * (N - 1)) + (5 * N * F) + N + F$. For this mathematical model, the number of constraints cannot be determined directly since it requires the values of the number of layers used, F, which is unknown. It would be necessary to determine appropriate value for the number of layers used, F, the approximate number of layers that are available for use, F, is arbitrarily assumed to be determined as follows:

$$F = \frac{\sum \text{ surface area of parts}}{\text{surface area of a layer}} \times 3.$$

This choice of *F* implies that we can achieve a layer surface utilization no less than one-third; while this may be true in general, it may not be true when most parts are large relative to surface area of a basket.

7. Basket-loading subproblem

The loading of the layers into a minimum number of baskets without exceeding the basket height can be formulated as an integer programming model using the solution from LL-FLP as input. The mathematical formulation of the basket-loading subproblem is stated below:

7.1. BL-FLP Model:

$$\text{Minimize}(z) = \sum_{t=1}^{U} z_t. \tag{24}$$

Subject to:

- Exactly one basket for each layer constraint:

$$\sum_{t=1}^{U} u_{jt} = 1 \quad \forall j = 1, \ldots, F. \tag{25}$$

Constraint (24) guarantees that each layer will be placed on only one basket.

- Basket usage constraint:

$$\sum_{j=1}^{F} u_{jt} \le z_t M \quad \forall t = 1, \ldots, U. \tag{26}$$

Constraint (25) ensures that if a basket is not used then no layer is placed into it; here we can use F in place of the large number M.

- Basket height constraints:

$$\sum_{j=1}^{F} u_{jt} g_j \le H \quad \forall t = 1, \ldots, U, \tag{27}$$

$$z_t, u_{jt} = 0 \text{ or } 1 \quad \forall j = 1 \ldots F, \text{ and } t = 1, \ldots, \mathrm{U}. \tag{28}$$

The mathematical model of the basket-loading subproblem has the form of an integer programming model. The total number of all variables is equal to $U * (1 + F)$, and the number of constraints is equal to $F + 2U$.

8. Computational experiments

8.1. PN-FLP Model

Suppose that after sorting the dataset (input data) based on material type, recipe number and height and satisfying the several rules that need to be followed in loading a basket, a set of four part types (see Fig. 5) is generated. The material type, external diameter, internal diameter, height, recipe number, and order quantity of the four parts are given in Table 2. To illustrate and validate the formulation of the part-nesting model presented in Section 5, the corresponding parts nesting problem is solved using the PN-FLP model.

The parts information in Table 2 is coded in GAMS 22.6 using the CPLEX solver. The nesting profit matrix and the optimum nested quantities for this experiment are shown in Fig. 6.

The optimal solution in Fig. 6 indicates that 82 units of part type #101 will be nested in part type #102, 80 units of part type #101 will be nested in part type #108, 20 units of part type #102 will be nested in part type #110, and 80 units of part type #108 will be nested in part type #110. Recognizing that multiple levels of nesting are implied in this solution, the solution in Fig. 6 is equivalent to having 62 units of part type #101 nested in part type #102, 20 units of part type #101 nested in part type #102 which in turn are nested in 20 units of part type #110, 80 units of part type #101 nested in part type #108 which in turn are nested in 80 units of part type #110, and 78 units of part type #101 are not nested. These results are presented in Table 3, and in graphical form in Fig. 7. For this experiment, the PN-FLP model contains 12 variables and 20 constraints. The PN-FLP model solved 502 parts problem in approximately 10 CPU milliseconds. Even if for too large number of parts, PN-FLP can yield optimal solutions in reasonable computer CPU time and memory.

8.2. LL-FLP model

To illustrate and validate the formulation of the layer-loading model presented in Section 6, an experiment of four part types was considered and solved using the LL-FLP model. The information for the four-part types is given in Table 4.

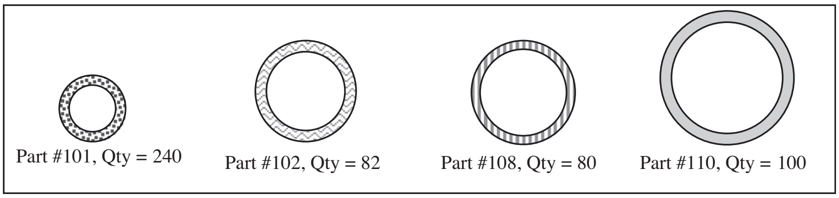

Fig. 5. Parts available for nesting.

Table 2
Input data for nesting.

Part #	Material	Order qty	Recipe #	e_i (mm)	n_i (mm)	Height (mm)
101	Std	240	16	270.6	235.4	107.2
102	Std	82	16	382.5	341.2	107.2
108	Std	80	17	385.9	350.0	107.2
110	Std	100	16	470.3	440.8	107.2

			Inner Parts (i)				
		Part Type #	101	102	108	110	
	Part Type #	n_i \ e_k	270.6	382.5	385.9	470.3	
Outer Parts (k)	101	235.4	*0* / **0**	*0* / **0**	*0* / **0**	*0* / **0**	**240**
	102	341.2	*0.79* / **82**	*0* / **0**	*0* / **0**	*0* / **0**	**82**
	108	350.0	*0.77* / **80**	*0* / **0**	*0* / **0**	*0* / **0**	**80**
	110	440.8	*0.61* / **0**	*0.87* / **20**	*0.88* / **80**	*0* / **0**	**100**
			240	**82**	**80**	**100**	

Fig. 6. Nesting profit matrix and optimum quantities.

Table 3
Output for the illustrative experiment.

Set no.	Part type no.	Material	Order qty	Recipe no.	e_k (mm)	n_i (mm)	Height (mm)	Nest level
1	110	Std	20	16	475.3	440.8	107.2	0
	102	Std	20	16	382.5	341.2	107.2	1
	101	Std	20	16	270.6	235.4	107.2	2
2	110	Std	80	16	475.3	440.8	107.2	0
	108	Std	80	17	385.9	350.0	107.2	1
	101	Std	80	16	270.6	235.4	107.2	2
3	102	Std	62	16	382.5	341.2	107.2	0
	101	Std	62	16	270.6	235.4	107.2	1
4[a]	101	Std	78	16	270.6	235.4	107.2	0

[a] Not nested.

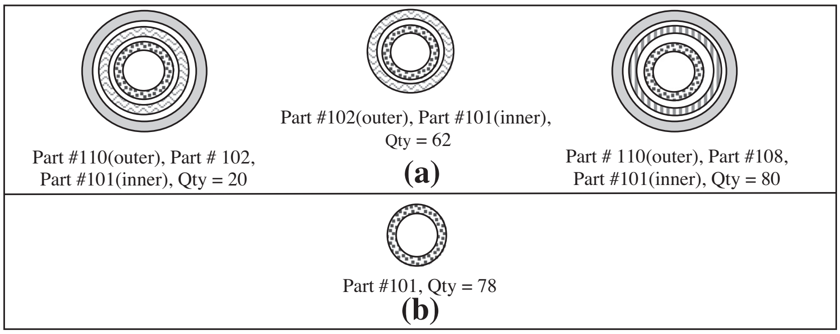

Fig. 7. (a) Three patterns of nested parts and (b) Parts left unnested, for illustrative example.

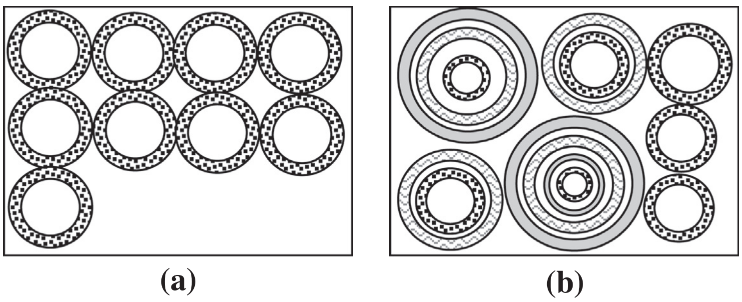

Fig. 8. (a) Layer with no nesting, (b) Layer with multi-level nesting.

Table 4
Information for parts used in the illustrative experiment.

Part type	Material type	Order qty	Recipe #	e_i (mm)	n_i (mm)	Height (mm)
110	Std	3	16	325.2	293.7	107.2
108	Std	5	16	216.8	184.1	107.2
102	Std	4	17	162.6	140.0	107.2
101	Std	15	17	108.4	86.6	107.2

Table 5
The optimal nesting for parts in Table 4.

Set #	Part #	Material	Order qty	Recipe #	e_i (mm)	n_i (mm)	Height (mm)	Nest level
1	110	Std	3	16	325.2	293.7	107.2	0
	108	Std	3	16	216.8	184.1	107.2	1
	101	Std	3	17	108.4	86.6	107.2	2
2	108	Std	2	16	216.8	184.1	107.2	0
	101	Std	2	16	108.4	86.6	107.2	1
3	102	Std	4	16	162.6	140.0	107.2	0
4	101	Std	10	17	108.4	86.6	107.2	0

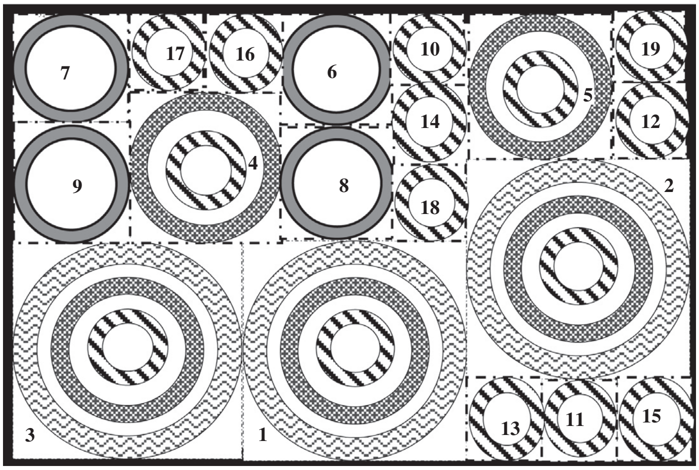

Fig. 9. Graphical representation for the layer-loading optimal solution.

Table 6
The optimal solution for the layer-loading illustrative experiment.

Part type	Layer ID no.	Part ID no.	Coordinates of part i	
No.	j	i	x_{ij}	y_{ij}
110	1	1	650.4	325.2
110	1	2	975.6	433.6
110	1	3	325.2	325.2
108	1	4	379.4	542.0
108	1	5	867.2	650.4
102	1	6	542.0	650.4
102	1	7	162.6	650.4
102	1	8	542.0	487.8
102	1	9	162.6	487.8
101	1	10	650.4	650.4
101	1	11	867.2	108.4
101	1	12	975.6	542.0
101	1	13	758.8	108.4
101	1	14	650.4	542.0
101	1	15	975.6	108.4
101	1	16	379.4	650.4
101	1	17	271.0	650.4
101	1	18	650.4	433.6
101	1	19	975.6	650.4

Table 7
CPU time and optimality gap using LL-FLP model.

Experiment no.	No. of parts	Parts fit exactly in	MIP solution obtained	Absolute gap	CPU time (sec)
1	19	1 Layer	1 Layer	0	2.537
2	38	2 Layers	2 Layers	0	14.846
3	57	3 Layers	3 Layers	0	100.088
4	76	4 Layers	4 Layers	0	711.872

Table 8
Input data for the basket-loading subproblem.

Layer ID no.	Material type	Max. recipe no.	Height (mm)
1	STD	17	110.7
2	STD	17	107.2
3	STD	17	107.2
4	STD	17	107.2
5	STD	17	119.3
6	STD	17	91.7
7	STD	17	147.2
8	STD	17	84.9
9	STD	17	87.5
10	STD	17	98.5
11	STD	17	100.0
12	STD	17	75.8
13	STD	17	137.1
14	STD	17	125.7

Table 9
Output for the basket-loading illustrative experiment.

Basket ID no.	Layer ID No.	Material type	Max. recipe no.	Height (mm)
1	4	STD	17	107.2
	5	STD	17	119.3
	8	STD	17	84.9
	11	STD	17	100.0
	12	STD	17	75.8
	13	STD	17	137.1
	14	STD	17	125.7
2	1	STD	17	110.7
	2	STD	17	107.2
	3	STD	17	107.2
	6	STD	17	91.7
	7	STD	17	147.2
	9	STD	17	87.5
	10	STD	17	98.5

In this experiment, the objective is to load all parts such that the required number of layers is minimized. The layer width and length are arbitrarily fixed at 650.4 and 975.6 mm, respectively. Table 5 presents the optimal nesting for the candidate parts obtained as output of PN-FLP model. Only the outer parts, the parts with zero nesting level, are considered as input data for the LL-FLP model. The input data for this experiment consists of the nesting level zero rows in Table 5.

A careful examination of each row in Table 5 reveals that there are 19 parts available as input to the layer-loading subproblem. Therefore, this layer-loading problem involves loading 19 primary parts (outer parts that have a zero nesting level). The approximate number of layers that are available for use in this problem would be three layers, computed using the expression for *F*. Therefore, the layer-loading model contains 5380 constraints and 4279 (117 continuous and 4164 binary) variables. The LL-FLP for this example problem has been developed and solved using GAMS 22.6 using the CPLEX solver. A graphical representation of the optimal loading pattern for this problem, turns out to be just one layer, is shown in Fig. 9. Table 6 shows the same solution using the top right corner coordinates of each part on the layer, with the origin located at its bottom left corner.

LL-FLP model provided an optimal solution of exactly one layer required about 2.5 CPU seconds with an absolute optimality gap of zero. Using LL-FLP model, the CPU times obtained for solving loading problems with number of parts that exactly fit into two, three, and four layers are given in Table 7.

The exact numbers of layers obtained as final MIP solution reflect the accuracy of the LL-FLP model formulations. For even number of parts than reported above, LL-FLP model is anticipated to yield optimal solutions in reasonable computer CPU

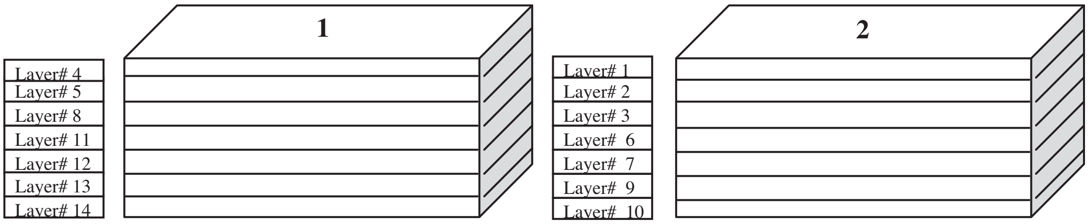

Fig. 10. Graphical representation of the optimal solution to the basket-loading illustrative example.

time. However, the computational work was limited by the memory size on the computer used. Solving the layer-loading subproblem with too large number of parts, a higher processor speed computer and larger memory size are required.

8.3. BL-FLP model

To illustrate and validate the formulation of the BL-FLP model proposed in Section 7, a set of 14 layers to be allocated in a minimum number of baskets (we carefully developed 14 layers that can fit into exactly two baskets). Each basket has a height of 750 mm. It is assumed that all layers and baskets have the same length and width dimensions. The layer information shown in Table 8 is assumed to be obtained as output of solving the LL-FLP.

To accomplish the task of allocating all layers in a minimum number of baskets, the BL-FLP model was used; the number of baskets available for loading was assumed to be equal to the number of layers (in this case 14 baskets). The BL-FLP model contains 210 variables and 238 constraints. The output of solving BL-FLP using GAMS 22.6 with CPLEX solver is shown in Table 9.

The numbers of baskets obtained as final MIP solution, exactly 2, reflect the accuracy of the BL-FLP model formulations. BL-FLP model solved the 14 layers problem in approximately 25 CPU milliseconds. It is anticipated that even if for too large number of layers, BL-FLP can still yield optimal solutions in reasonable computer CPU time and memory. The graphical illustration of the optimum allocation of all layers using two baskets as the final optimal solution is shown in Fig. 10.

9. Conclusions

This paper described the development of a promising approach that calls for reformulation of nonlinear integer programming model into three linear mathematical models to provide a solution to three-dimensional packing problem of cylindrical parts. For solving the furnace-loading problem, three subproblems were identified: parts nesting, layer-loading, and basket-loading subproblems. For the first subproblem; a model that resembles a transportation problem was developed with the objective of maximizing the nesting of parts within one another. A mixed integer linear programming model was developed for the second subproblem to minimize the number of layers used; this model considered the exact location (coordinates) of each part to be positioned on a basket layer. For the third subproblem, a 0–1 model was developed with an objective to minimize the number of baskets used. These models were tested for illustrative experiments. The number of layers and baskets obtained as a final solution reflect the accuracy of the model formulations. The results show that the proposed PN-FLP and BL-FLP models can be applied large problems and we can still yield optimal solutions in reasonable computer CPU time and memory. However, the computational work using LL-FLP model was limited by the memory size on the computer used. Solving large problems may require a higher processor speed computer and larger memory size.

As far as the authors are aware the three-dimensional packing problem of cylindrical part of different sizes that can be nested within one another into a rectangle has not been addressed in the literature, and no similar approach has been reported in the published research. Therefore, the models developed for solving three-dimensional packing problem of cylindrical parts have no "bench-mark" with which to compare or test their performance other than current manual methods.

References

[1] H.T. Dean, J.N. Baggaley, R.J.W. James, Three dimensional container packing of drums and pallets [online], University of Canterbury, New Zealand, Available from: http://www.orsnz.org.nz/conf34/PDFs/James.pdf. [Accessed 28 July 2006], 1999.
[2] B. Ram, The pallet loading problem: a survey, Int. J. Prod. Econ. 28 (1992) 217–225.
[3] C.S. Chen, S. Sarin, B. Ram, The pallet packing problem for non-uniform box sizes, Int. J. Prod. Res. 29 (10) (1991) 1963–1968.
[4] W.B. Dowsland, Three-dimensional packing-solution approaches and heuristic development, Int. J. Prod. Res. 29 (8) (1991) 1673–1685.
[5] C.P. Han, P.J. Egbelu, K. Knott, A heuristic approach to the three-dimensional cargo loading problem, Int. J. Prod. Res. 27 (5) (1989) 757–774.
[6] G. Abdou, M. Yang, A systematic approach for the three-dimensional palletization problem, Int. J. Prod. Res. 32 (10) (1994) 2381–2394.
[7] G. Abdou, M. Elmasry, 3D random stacking of weakly heterogeneous palletization problems, Int. J. Prod. Res. 37 (1999) 1505–1524.
[8] K.A. Dowsland, Palletisation of cylinders in cases, OR Spektrum 13 (1991) 171–172.
[9] H. Fraser, J. George, Integrated container loading software for pulp and paper industry, Eur. J. Oper. Res. 77 (3) (1994) 466–474.
[10] H. Isermann, Heuristiken Zur Lösung Des Zweidimensionalen Pack problems Für Rundgefäβe, OR Spektrum 13 (1991) 213–223.
[11] J. George, J. George, B. Lamar, Packing different sized circles into a rectangular container, Eur. J. Oper. Res. 84 (1995) 693–712.

[12] M. Hifi, V.T. Paschos, V. Zissimopoulos, Circular cutting problem: a simulated annealing approach. Paris I – Economie Mathematique et Applications, technical report 94.15, Université de Paris I, Pantheon, Sorbon, 1994.
[13] Y.G. Stoyan, G.N. Yaskov, Mathematical model and solution method of optimization problem of placement of rectangles and circles taking into account special constraints, Int. Trans. Oper. Res. 5 (1998) 45–57.
[14] D. Zhang, Y. Liu, S. Chen, Packing different-sized circles into a rectangular container using simulated annealing algorithm, in: International Conference on, Computational Intelligence, 2004, pp. 388–391.
[15] W.Q. Huang, Y. Li, C.M. Li, R.C. Xu, New heuristics for packing unequal circles into a circular container, Comput. Oper. Res. 33 (8) (2006) 2125–2142.
[16] A.R. Babu, N.R. Babu, Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms, Int. J. Prod. Res. 37 (1999) 1625–1643.
[17] T.H. Wu, J.F. Chen, C. Low, P.T. Tang, Nesting of two-dimensional parts in multiple plates using hybrid algorithm, Int. J. Prod. Res. 41 (16) (2003) 3883–3900.